



ARTICLE

Integrating Texture Attention and Task Guidance for Waterline Keypoint Detection

Jinlin Chen^{1,2} and Yiquan Wu^{1,*}

¹College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

²College of Finance and Mathematics, Huainan Normal University, Huainan, China

*Corresponding Author: Yiquan Wu. Email: nuaatracking@163.com

Received: 19 May 2026; Accepted: 16 July 2026; Published: 13 August 2026

ABSTRACT: Accurate waterline detection is critical for automated ship draft monitoring but remains challenging due to weak textures, low contrast, and dynamic maritime interferences. This paper presents TGNet, a task-guided framework that jointly optimizes character recognition and waterline keypoint localization. TGNet introduces a triple attention network (TANet) with channel, spatial, and texture attention modules to enhance discriminative feature extraction. Crucially, a task-to-task guidance mechanism leverages detected draft characters to spatially constrain and crop feature maps, focusing the keypoint detection head on the most relevant waterline region. Extensive experiments on three large-scale aerial datasets show that TANet consistently improves baseline detectors by an average of 2.8% recall and 3.6% mean average precision (mAP). On our self-built UAV ship draft dataset, TGNet achieves 92.3% recall and 93.8% mAP₇₅ at 56 FPS, outperforming state-of-the-art keypoint and segmentation methods while maintaining real-time efficiency. The proposed approach demonstrates a robust and efficient solution for practical autonomous draft reading.

KEYWORDS: Ship draft measurement; waterline keypoint detection; task-guided deep learning; triple attention mechanism; UAV-based maritime vision; real-time ship monitoring

1 Introduction

With the rapid advancement of digital image processing technology, computer vision-based ship waterline detection methods have emerged as a crucial research direction in the field of ship automated inspection. From theoretical groundwork to technological breakthroughs, this domain has undergone a comprehensive evolutionary process.

In the theoretical foundation phase, Schmidt [1] conducted pioneering research on the linearized wake of two-dimensional shallow-draft ships. The explicit expressions for wake amplitude and phase derived in their work provided a critical theoretical basis for subsequent waterline analysis and calculations. However, significant gaps in accuracy, robustness, and computational efficiency between early theoretical research and practical applications have driven continuous exploration of new technical approaches.

Current research status indicates that existing methods still confront three core challenges: insufficient robustness under complex wave interference and variable lighting conditions, difficulty balancing high accuracy with real-time performance, and limited precision in handling deformed or occluded scale characters. These bottlenecks have spurred the development of new solutions.

From the perspective of task characteristics, waterline detection and character recognition are closely interrelated sub-tasks that possess a natural foundation for collaboration. Waterline localization relies on character markers as spatial reference benchmarks, while the distinct features of character regions provide structural information support for waterline detection.

From the perspective of feature representation, the relatively simple structure of waterlines allows the application of methods such as edge detection, instance segmentation, and keypoint detection. Existing segmentation-based methods often fail under challenging maritime conditions. For instance, under low-contrast lighting or strong wave reflections, segmentation networks tend to misclassify water surface ripples as waterline boundaries, producing fragmented or biased predictions. Edge detection methods, while computationally efficient, are sensitive to noise and hull texture variations, frequently generating spurious edges that obscure the true waterline location. In contrast, keypoint detection offers a more robust alternative by leveraging geometric priors and sparse spatial constraints [2].

Although instance segmentation is one of the commonly used methods for waterline detection in water level images, it still faces significant challenges when applied under complex environmental conditions such as glare, contamination, or occlusion. In contrast, the geometric prior information provided by keypoints can effectively overcome the limitations of pixel-level methods in scenarios with blurred boundaries or occlusions, and this characteristic has been fully validated in fields such as medical imaging and autonomous driving. However, simply combining keypoint detection with segmentation methods, while improving accuracy, inevitably increases model complexity, hindering practical deployment. Meanwhile, the dual attention mechanism integrating channel attention and spatial attention [3] has demonstrated excellent performance in enhancing relevant features and suppressing noise interference. Nevertheless, its application in UAV-based waterline detection scenarios requires lightweight structural redesign to satisfy onboard real-time inference requirements.

Based on the above in-depth analysis, this study proposes an innovative waterline recognition scheme that combines keypoint detection with a lightweight multi-attention mechanism. The scheme aims to establish a guided task interaction mechanism, leveraging the discriminative features of character recognition to optimize waterline localization. This study makes three key innovations in ship waterline detection:

- First, to meet the core needs of accurate draft mark localization and waterline positioning in intelligent draft detection, we propose an innovative task-guidance approach. A unified model is constructed to simultaneously handle draft character recognition and waterline detection. Using a shared backbone for feature extraction, we crop feature maps of identified draft character regions and further detect waterline keypoints via a lightweight feature fusion module, significantly enhancing local waterline localization accuracy.
- Second, to extract waterline keypoint features more precisely, we upgrade the dual attention network to a triple attention network by adding a texture attention module. Meanwhile, we compress the module structure from five to three levels, reducing parameters while maintaining efficiency. Experiments show this lightweight design preserves keypoint detection accuracy and greatly improves inference speed.
- Third, we propose a comprehensive ship draft detection solution: waterline keypoints are used for curve fitting, combined with a curve model built from the center points of draft characters' lower boundaries. The intersection of the two curves is the actual waterline position, enabling intelligent calculation of ship draft. Tests on a self-built UAV draft dataset verify that this method achieves real-time, high-precision draft measurement.

The remainder of this paper is organized as follows. [Section 2](#) reviews related work on waterline detection. [Section 3](#) presents the proposed TGNNet in detail. [Section 4](#) describes the experimental setup and reports comprehensive results. [Section 5](#) concludes the paper.

2 Related Work

To provide a comprehensive overview of prior research, this section reviews existing waterline detection methods. The technological evolution can be divided into three main stages.

The first stage was characterized by traditional image processing methods. Tsujii et al. [4] developed an automatic draft reading system based on morphological operations and Canny edge detection, achieving centimeter-level accuracy in controlled environments. Simultaneously, Liu et al. [5] enhanced detection accuracy to ± 3 cm by fusing visible and infrared image information, significantly improving system reliability under low-visibility conditions. Ding et al. [6] demonstrated the potential of synthetic aperture radar data in waterline extraction, yet the stringent requirements of single-polarization methods for high land-sea contrast and the limitations of dual-polarization methods in intertidal zone water detection highlighted the application bottlenecks of remote sensing technology.

The second stage was driven by the rise of deep learning. Zheng et al. [7] proposed a difference-focusing fusion decision method, an ensemble learning framework, to improve deep learning segmentation for waterline extraction in synthetic aperture radar imagery. Zhang et al. [8] proposed a dual-branch convolutional neural network architecture incorporating an attention mechanism, achieving 96.7% accuracy through joint load line recognition and draft calculation. Chen et al. [9] developed a multi-scale convolutional neural network(CNN) that achieved 98.2% accuracy while substantially improving detection efficiency. Liu et al. [10] innovatively reformulated waterline detection as a classification task, each contributing to the field from distinct perspectives.

The third stage exhibits a trend of multi-technology integration and collaborative innovation. Zhou et al. [11] proposed BGTracker, which introduces a cross-task bidirectional guidance strategy to align detection and re-identification features for multi-object tracking. Zhu et al. [12] presented FTMNet, which employs a fusion-driven task mutual-guidance framework to jointly enhance image quality and classification accuracy for few-shot hyperspectral image classification. However, the direct transplantation of the Mask region-based CNN(R-CNN) method applied by von Braun et al. [13] in canoe sprint waterline detection to ship draft reading still faces substantial challenges. Wang et al. [14] proposed SDRNet, a multi-task network combining keypoint detection, semantic segmentation, and text recognition for draft reading. Zhang et al. [15] introduced MFFKD, a keypoint detection network that directly localizes waterline keypoints via multi-scale feature fusion, achieving faster inference than segmentation-based approaches.

Beyond single-task methods, the multi-task collaboration mechanism has formed a comprehensive system in the field of computer vision. From early works such as joint detection-embedding [16], joint detection and tracking [17], and multi-object tracking [18], which achieved detection-tracking collaboration through shared feature extraction networks, to multi-task learning with attention [19] and Cross-Stitch [20].

Notably, recent research on task collaboration has shown a trend of deepening from architectural sharing to semantic fusion. In detection-segmentation collaboration, the work of [21] exemplifies this paradigm shift by deeply integrating open-vocabulary detection and referring segmentation within a unified architecture, thereby facilitating cross-scene feature adaptation. Xu et al. [22] proposed H2FA R-CNN, which addresses cross-domain weakly supervised object detection through holistic and hierarchical feature alignment. Additionally, the “decoupling-collaboration” paradigm [23], proposed to address multi-task conflicts by explicitly modeling inter-task logical dependencies, provides a new approach for task collaboration in

complex scenarios. Xu et al. [24] proposed a multi-task learning framework tailored for inland waterways, enabling simultaneous ship detection and waterline segmentation. In 2024, Fang et al. [25] introduced the detection-segmentation network based on an improved YOLOv4 and DeepLab architecture, achieving sub-centimeter (<1 cm) detection accuracy. Zhang et al. [26] fused near-infrared and RGB image features using a dual-stream network, attaining recognition accuracy of 99.3% and a measurement accuracy of ± 0.01 m. A joint semantic segmentation and edge detection approach was employed by Steccanella et al. [27] to effectively mitigate water surface reflection interference. These technological breakthroughs provide critical support for intelligent shipping systems and automated port management.

Unlike conventional multi-task learning approaches that share features passively, our task guidance mechanism actively transfers character-derived spatial priors to constrain the waterline search region. Compared with ROI-based detection, our feature-level cropping preserves contextual information essential for keypoint regression. These design choices collectively distinguish our method from existing paradigms.

Based on the above review, existing methods still have several limitations. First, most approaches treat character recognition and waterline detection as independent or loosely coupled tasks, lacking explicit task-to-task guidance. Second, existing attention mechanisms do not specifically address the weak-texture nature of waterlines in complex maritime environments. Third, many high-accuracy methods sacrifice real-time performance, limiting their deployability on UAV platforms. It is worth distinguishing our task-guidance mechanism from seemingly similar paradigms. First, compared with standard ROI-based detection (e.g., Faster R-CNN), our approach operates at the feature-map level with adaptive boundary adjustments (Algorithm 2) rather than resizing cropped image patches. This preserves spatial continuity and resolution, which is critical for accurate keypoint regression. Second, unlike conventional multi-task learning (MTL) that passively shares features via soft parameter constraints (e.g., MTAN, Cross-Stitch), our framework establishes a causal dependency where character detection actively determines the waterline's search region via explicit feature cropping. This goes beyond "cooperative" learning and implements a "guiding" mechanism tailored for weak-texture targets. Third, while two-stage keypoint detectors rely on generic region proposals, our guidance is driven by semantic understanding (recognizing the lowest draft character), incorporating high-level knowledge about the spatial relationship between characters and the waterline. These distinctions highlight the novelty of our approach as a new causal task-routing paradigm. These observations motivate the proposed approach presented in the next section.

3 Proposed Approach

3.1 Draft Survey Methodology

The water gauge calculation scheme comprises three phases. In the detection phase, the system extracts all gauge character sets $S_c = \{(C_i, x_i, y_i)\}$ and waterline key point sets $S_p = \{(x_j, y_j)\}$. In the modeling phase, to convert these discrete point sets into continuous geometric representations, we employ least-squares polynomial fitting. Specifically, the waterline curve is modeled as $x = f_1(y)$, where f_1 is a k -th order polynomial (typically $k = 2$ for slight hull curvature or $k = 1$ for linear approximation) that minimizes the squared error $\sum_j [x_j - f_1(y_j)]^2$ over the detected keypoints S_p . For the gauge modeling, the vertical alignment of the character centers is represented by $y = f_2(x)$, defined analogously as a polynomial fitted to $\{(x_i, y_i)\}$ from S_c . The relationship between the horizontal coordinate and the actual draft value is established by $C = f_3(x)$, which is implemented via linear interpolation between consecutive detected characters: for $x \in [x_i, x_{i+1}]$, $f_3(x) = C_i + (C_{i+1} - C_i) \frac{x - x_i}{x_{i+1} - x_i}$. In the evaluation phase, the system solves for the intersection point (x^*, y^*) satisfying $x^* = f_1(y^*)$ and $y^* = f_2(x^*)$, then computes the final water level reading as $C^* = f_3(x^*)$. For detailed pseudocode, refer to Algorithm 1.

Algorithm 1: Water gauge calculation**Require:** Input image I , pre-trained detection model $\mathcal{M}$ (character + keypoint)**Ensure:** Final water level reading $c^* \in \mathbb{R}$

```

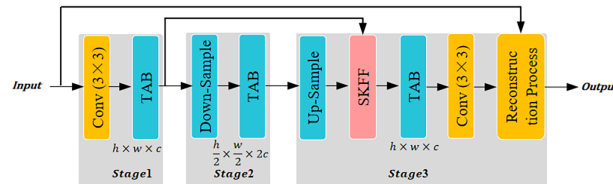
1: function WATERGAUGE_CALCULATION( $I$ )
2:   Phase 1: Detection
3:    $\mathcal{S}_c \leftarrow \emptyset$ 
4:    $\mathcal{S}_p \leftarrow \emptyset$ 
5:   Run  $\mathcal{M}$  on  $I$  to obtain  $\mathcal{S}_c$  and  $\mathcal{S}_p$ 
6:   Phase 2: Geometric Modeling
7:    $f_1(y) = \arg \min_{f \in \mathcal{P}_k} \sum_{(x_j, y_j) \in \mathcal{S}_p} (x_j - f(y_j))^2$ 
8:    $f_2(x) = \arg \min_{f \in \mathcal{P}_l} \sum_{(x_i, y_i) \in \mathcal{S}_p} (y_i - f(x_i))^2$ 
9:    $f_3(x) = \text{linear interpolation of } \{(x_i, C_i)\}$ 
10:  Phase 3: Reading Evaluation
11:   $(x^*, y^*)$  such that  $x^* = f_1(y^*)$  and  $y^* = f_2(x^*)$ 
12:   $c^* = f_3(x^*)$ 
13:  return  $c^*$ 
14: end function

```

On the methodological level, this research systematically innovates across three key areas: a triple attention mechanism, an inter-task guidance architecture, and a joint training strategy. First, by constructing a triple attention network integrating channel, spatial, and texture information, the model significantly enhances its perception of waterline texture features. Second, a task-guided mechanism based on feature cropping is designed, where character detection results direct waterline keypoint localization, enabling cross-task knowledge transfer. Finally, a two-stage joint training strategy is adopted, which maintains the feature extraction capability of the backbone network while separately optimizing the character recognition and waterline detection branches. These three core components mutually reinforce each other, collectively forming an efficient and accurate water level detection solution.

3.2 Triple Attention Network

In this section, we provide a detailed description of the architecture of triple attention network (TANet), designed for enhancing texture in water gauge images. As shown in Fig. 1, TANet consists of three stages, each incorporating a triple attention block (TAB). Compared to the previous dual attention network (DANet) [3], a texture attention module (TAM) has been introduced, enabling the three attention modules to effectively integrate channel, spatial, and texture information simultaneously. A reconstruction process is employed to extract more water surface texture features from the observed water gauge images. Additionally, the number of stages has been reduced from five to three, thereby decreasing the model size.

**Figure 1:** Overview of TANet.

The detailed channel configuration of TANet is as follows: Stage 1 transforms the input from C to 64 channels via a 3×3 convolution, followed by two TABs with 64 channels each. Stage 2 reduces spatial resolution by half via 3×3 convolution and applies two TABs with 128 channels. Stage 3 repeats this pattern with 256 channels and a single TAB. All TABs maintain a bottleneck ratio of 4 in the channel attention branch, and the TAM's dilated convolution uses a kernel size of 3×3 with dilation rate $r = 2$. This three-stage design reduces parameters by 40% compared to the five-stage DANet baseline.

The TAB is a lightweight feature extraction module and consists of three main components: the position attention module(PAM), the channel attention module(CAM), and the texture attention module, as illustrated in Fig. 2. The PAM calculates the similarity between any two spatial positions on the feature map to generate a position attention map. This map is then used to weight and aggregate spatial features, thereby emphasizing semantically important regions. The CAM computes inter-channel correlations to produce a channel attention map, which enhances discriminative channel-wise features via weighted aggregation. The TAM adopts a dual-branch architecture to effectively extract and integrate texture-related features.

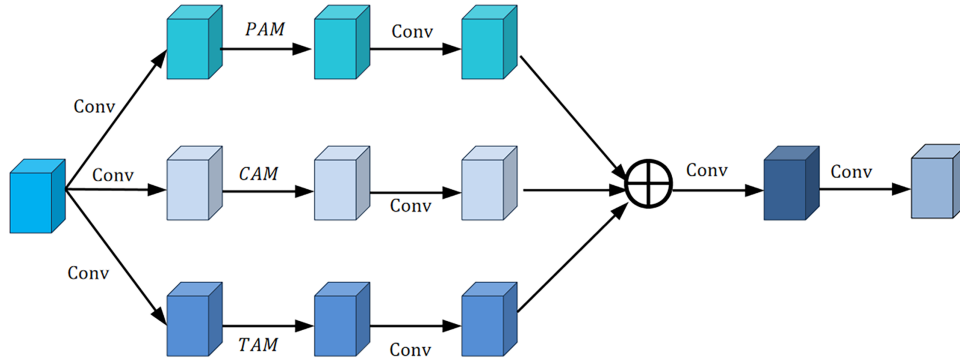


Figure 2: Overview of TAB.

In the TAM, the main branch conducts basic feature extraction through a sequence of 3×3 convolution, batch normalization, and rectified linear unit activation(ReLU), maintaining an output dimension of (B, C_{out}, H, W) . Simultaneously, the texture branch utilizes a 3×3 dilated convolution with a dilation rate of 2, followed by batch normalization and ReLU, to capture broader contextual texture information, producing features with dimension (B, C_{out}, H, W) . During the feature fusion stage, the outputs of the two branches are first concatenated along the channel dimension, expanding the channel count to $2 \times C_{out}$. This is subsequently processed by a 1×1 convolution, batch normalization, and ReLU to compress the channel dimension back to (B, C_{out}, H, W) . When the input and output channel dimensions are inconsistent, a conditional 1×1 convolutional shortcut connection is employed to ensure compatibility, and the result is added element-wise to the output of the main branch. The final output feature map retains the dimension (B, C_{out}, H, W) . This design significantly enhances the texture feature extraction capability through a dedicated texture branch while preserving the advantages of the original residual structure. The architecture of our TAM is illustrated in Fig. 3.

The feature representations enriched by the TANet, particularly the texture details, establish a robust foundation for the subsequent inter-task guidance. The following subsection elaborates on how these enhanced features are leveraged to guide the waterline detection task using information from character recognition.

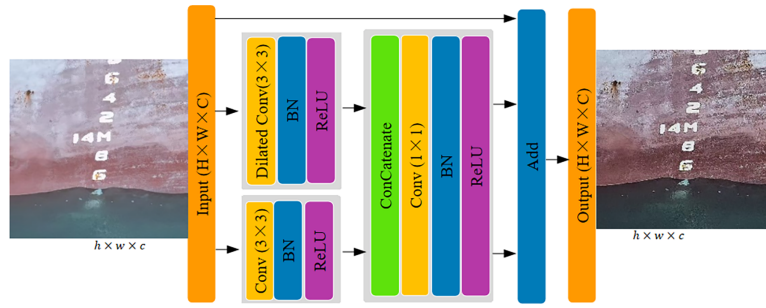


Figure 3: Overview of TAM.

Fig. 4 shows the overall workflow. The shared backbone with TAnet extracts features for both character detection and keypoint detection. Character detection results (bounding boxes) are used as spatial priors to guide feature cropping, and the cropped features are then fed into the keypoint head for accurate waterline localization.

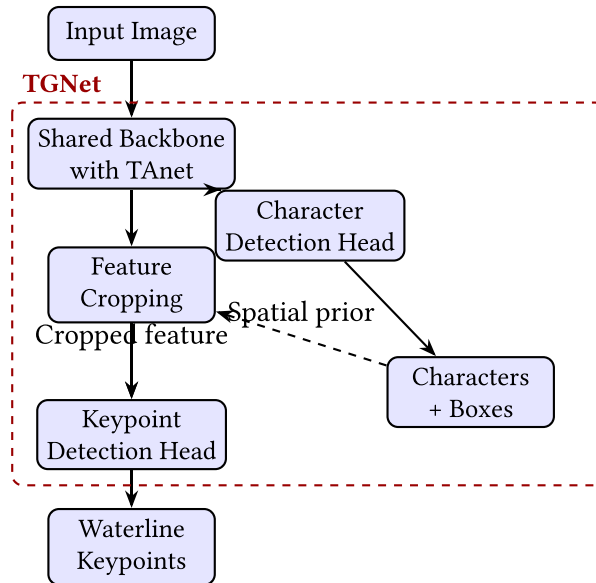


Figure 4: High-level architecture of TGNNet. The character detection head provides spatial priors to guide feature cropping, which supplies focused features to the keypoint head for waterline localization.

3.3 Task-to-Task Guidance

To elucidate the inter-task guidance mechanism, we have developed a novel model based on the empirically validated and reliable YOLOv8 architecture. This design builds upon the broad academic consensus regarding YOLOv8's community support, computational efficiency, and technical reliability, thereby ensuring the robustness of our research baseline.

As illustrated in Fig. 5, the overall architecture comprises three core components: a shared backbone network, a multi-scale feature fusion module, and dual detection heads dedicated to character recognition and waterline keypoint detection, respectively. The right branch of the architecture corresponds to the character detection module, which consists of a three-layer feature fusion structure and its specialized detection

The diagram illustrates the architecture of the proposed YOLOv5s-TP model. It starts with an input image of size 640x640x3. The input is processed by a Backbone consisting of a sequence of CBS, C2f, CBS, C2f, CBS, C2f, CBS, C2f, and SPP blocks. The output of the Backbone is fed into a Neck and a Detect-Head. The Neck consists of Concat, UpSample, and C2f blocks. The Detect-Head consists of Decoupled Head blocks for 80x80, 40x40, and 20x20 feature maps. A Pose-Head is also shown, consisting of Conv2d and CBS blocks. The Neck and Detect-Head are connected to the Backbone via dashed lines.

We designate this watermark detection method based on the YOLO task guidance mechanism as the task guidance network (TGNet). In TGNet, the guidance from the right branch to the left branch requires feature cropping, while simultaneously recording the cropping position to facilitate subsequent coordinate mapping of keypoints to the original image. The specific algorithm workflow is detailed in Algorithm 2. The algorithm accepts two inputs: first, a set of k detected character bounding boxes ($\mathcal{B}$), where each bounding box is represented by a quadruple (x_i, y_i, w_i, h_i) with (x_i, y_i) denoting the top-left coordinates and w_i, h_i representing width and height, respectively; second, the feature map $\mathbf{F}$ to be cropped, with dimensions (W, H, C) corresponding to width, height, and number of channels. The algorithm outputs include a cropped feature map $\tilde{\mathbf{F}}$ with spatial dimensions reduced to $(\lfloor W/2 \rfloor, \lfloor H/2 \rfloor)$ while maintaining the original channel count, along with the top-left coordinates (x_0, y_0) of the cropped region.

The core algorithm workflow proceeds as follows: First, the top-left vertical coordinates of all characters are extracted from the character bounding box set, and the target character A corresponding to the maximum y-value is selected (this strategy is based on the prior knowledge that the bottom-most character has the closest spatial proximity to the waterline, thus maximizing coverage of the critical waterline region). Next, the center coordinates (c_x, c_y) of character A are calculated and rounded to the nearest integer to ensure pixel-level alignment with the feature map. The target cropping size is then set to half of the original feature map dimensions (using floor division to guarantee integer dimensions), and the initial cropping

region's top-left and bottom-right coordinates are computed based on character A's center. To address potential boundary overflow of the initial cropping region, a bidirectional adjustment strategy is implemented: horizontal left-boundary overflow triggers rightward shifting, right-boundary overflow triggers leftward shifting, with analogous adjustments in the vertical direction, ensuring the cropped region remains entirely within the valid feature map area. Finally, the adjusted cropped feature map and its positional information are output. By concentrating on feature preservation within critical waterline regions, this algorithm not only enhances feature discriminability but also maintains compatibility with subsequent detection processes, all while simultaneously meeting the system's real-time performance requirements.

Algorithm 2: Task-guided feature cropping

Require:

- 1: $\mathcal{B} = \{(x_i, y_i, w_i, h_i)\}_{i=1}^k$.
- 2: $\mathbf{F} \in \mathbb{R}^{W \times H \times C}$.

Ensure:

- 3: $\tilde{\mathbf{F}} \in \mathbb{R}^{W_c \times H_c \times C}$, $W_c = \lfloor W/2 \rfloor$, $H_c = \lfloor H/2 \rfloor$, (x_0, y_0)
 - 4: **function** TASKGUIDEDCROP($\mathcal{B}, \mathbf{F}$)
 - 5: $A \leftarrow \arg \max_{(x,y,w,h) \in \mathcal{B}} (y + h)$
 - 6: $A: c_x \leftarrow \lfloor x_A + w_A/2 \rfloor$, $c_y \leftarrow \lfloor y_A + h_A/2 \rfloor$
 - 7: $W_c \leftarrow \lfloor W/2 \rfloor$, $H_c \leftarrow \lfloor H/2 \rfloor$
 - 8: $x_s \leftarrow c_x - \lfloor W_c/2 \rfloor$, $x_e \leftarrow c_x + \lfloor W_c/2 \rfloor$
 - 9: $y_s \leftarrow c_y - \lfloor H_c/2 \rfloor$, $y_e \leftarrow c_y + \lfloor H_c/2 \rfloor$
 - 10: **Adjust boundaries to stay within** $[0, W] \times [0, H]$:
 - 11: **if** $x_s < 0$ **then**
 - 12: $x_e \leftarrow x_e - x_s$, $x_s \leftarrow 0$
 - 13: **else if** $x_e > W$ **then**
 - 14: $x_s \leftarrow x_s - (x_e - W)$, $x_e \leftarrow W$
 - 15: **end if**
 - 16: **if** $y_s < 0$ **then**
 - 17: $y_e \leftarrow y_e - y_s$, $y_s \leftarrow 0$
 - 18: **else if** $y_e > H$ **then**
 - 19: $y_s \leftarrow y_s - (y_e - H)$, $y_e \leftarrow H$
 - 20: **end if**
 - 21: $(x_0, y_0) \leftarrow (x_s, y_s)$
 - 22: $\tilde{\mathbf{F}} \leftarrow \mathbf{F}[x_0 : x_0 + W_c, y_0 : y_0 + H_c, :]$
 - 23: **return** $(\tilde{\mathbf{F}}, x_0, y_0)$
 - 24: **end function**
-

The waterline extraction process proceeds as follows: first, the character detection head locates the lowest visible draft character and its bounding box; then, Algorithm 2 crops the feature map around this character's spatial position to obtain a focused region; after which, the keypoint detection head regresses waterline keypoints from this cropped feature map; subsequently, curve fitting reconstructs the waterline boundary; finally, the computed reading is derived by intersecting the fitted waterline with the draft-scale model following Algorithm 1.

3.4 Model Training and Detection

The training and parameter update strategy for the proposed model consists of two core stages. The first stage focuses on training the character recognition Backbone, treating the waterline as a special character

(Class '12'). Specifically, the YOLOv8 architecture is used as the base model, with its Backbone serving as a shared feature extractor. The Neck and detection Head are modified so that the final classification output dimension is set to 12, corresponding to the 11 gauge characters (0...9 and 'm') plus an additional special class representing the waterline. Draft figure markings shall be concentrated and appear only within the local area adjacent to the waterline, rather than being fully annotated along the entire length of the draft scale from bottom to top. This method ensures clarity and aligns with standard ship draft marking conventions, as illustrated in Fig. 6 for marking keypoints. In our experiment, to facilitate subsequent polynomial curve fitting, we annotated 16 keypoints along each waterline.

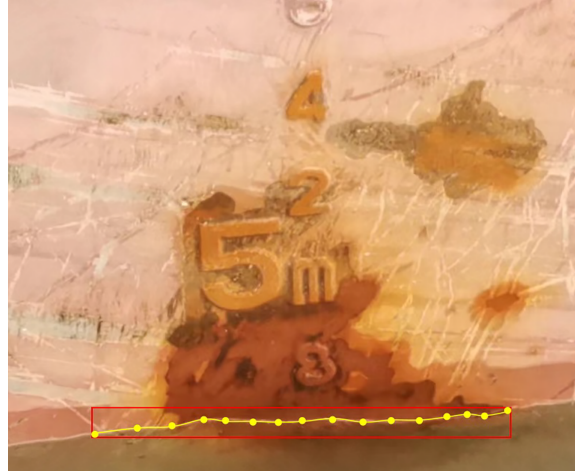


Figure 6: Examples of characters and waterline markings.

In terms of data preparation, the annotation of the training dataset is crucial: each waterline region in the image is no longer labeled as a line, but rather as a special bounding box with a unique waterline character label, while all digits and the 'm' character are annotated normally. Specifically, paired characters refer to the digits (0–9) that are spatially adjacent on the draft scale and jointly encode continuous depth values; they are annotated individually but logically paired during final reading computation. The special symbol includes both the unit marker 'M' and the waterline class '12'—the latter being a novel design that treats the waterline as a character class, thereby unifying detection without a separate segmentation branch. The objective of this stage is to enable the model to accurately localize and recognize all gauge characters as well as the waterline region. By explicitly defining the waterline as a character, the loss function assigns equal importance to waterline features and digit characters. This forces the backbone to actively extract and retain key visual characteristics of the waterline—such as its horizontal edge features and spatial relationship with the digits—effectively preventing the waterline features from being ignored or overshadowed.

During this stage, we update all model weights—including the Backbone, Neck, and Head—via back-propagation. The loss function used is the standard YOLO multi-task loss, comprising classification loss, bounding box regression loss, and confidence loss.

The second stage focuses on training the waterline keypoint detection branch, during which the weights of the backbone trained in the first stage are frozen. The network structure is modified as follows: the pre-trained Backbone weights, which have learned to extract rich features of both characters and the waterline, are retained but frozen. The original Neck and Head for object detection are disabled. A new, parallel task-specific Neck and Head are added for waterline keypoint detection. This branch can be connected to one

or multiple feature maps from the Neck and may consist of a small convolutional network that ultimately outputs either a heatmap of the waterline keypoints or directly regresses their coordinates.

The sole objective of this stage is to train the newly added waterline keypoint detection branch. The model leverages the well-learned shared features from the first stage to accurately predict the location of the waterline. Feature maps from the backbone are sliced and fed into the dedicated branch Neck. The parameter update strategy is clear: all frozen Backbone parameters do not undergo gradient computation or weight updates, ensuring that the valuable features learned in the first stage remain intact. Only the weights of the new waterline keypoint detection branch are trained and updated via backpropagation based on the keypoint detection loss.

During testing, the model performs water level recognition efficiently in a forward inference manner. The input image first passes through the shared backbone for feature extraction and character detection. The character detection branch outputs all recognized gauge characters along with their locations. The system then intelligently selects the center point of the lowest character as a key reference. A small local feature map-containing crucial contextual information where the waterline is likely present-is precisely cropped from the backbone's output feature maps. This cropped feature patch is directly fed into the dedicated waterline detection branch for refined processing. The core task of this branch is to regress the precise coordinates of the waterline keypoints based on these focused features.

Our key innovation lies in integrating gauge character detection and waterline keypoint detection within a unified model framework, achieving task synergy through a dual-branch architecture. While performing its own task, the character detection branch effectively acts as an attention guidance mechanism for the other branch, enabling it to quickly focus on the image region where the waterline is most likely to appear. This design cleverly leverages prior knowledge of character spatial distribution, not only significantly improving the localization accuracy of waterline detection and avoiding computational waste from full-image search, but also greatly reducing the overall inference time and computational complexity of the dual-task detection system-achieving a dual optimization of both accuracy and efficiency.

3.5 Unified Pipeline Formulation

The overall TGNet pipeline, illustrated in Fig. 4, comprises a sequence of five learnable mappings that transform the input image I into the final water-level reading c^* . Formally, this process is expressed as:

$$\begin{aligned}
 F &= \mathcal{F}(I; \theta_b) \\
 \{(C_i, b_i)\} &= \mathcal{D}_{\text{char}}(F; \theta_c) \\
 \tilde{F} &= \mathcal{G}(F, \{b_i\}) \\
 \{p_j\} &= \mathcal{D}_{\text{kpt}}(\tilde{F}; \theta_k) \\
 c^* &= \mathcal{H}(\{(C_i, b_i)\}, \{p_j\})
 \end{aligned} \tag{1}$$

Here, $\mathcal{F}$ denotes the shared backbone enhanced with TAnet (parameterized by θ_b), which extracts multi-scale feature maps F from the input. The character detection head $\mathcal{D}_{\text{char}}$ (with parameters θ_c) then predicts all visible draft characters C_i and their bounding boxes b_i . Using the spatial priors from these detections, the task-guided cropping operation $\mathcal{G}$ (detailed in Algorithm 2) selectively extracts a region $\tilde{F}$ of the feature map that is most likely to contain the waterline, effectively focusing subsequent computations. The keypoint detection head $\mathcal{D}_{\text{kpt}}$ (parameterized by θ_k) regresses the waterline keypoints p_j from this cropped feature. Finally, the post-processing function $\mathcal{H}$ (Algorithm 1) computes the intersection between the fitted waterline and the draft-scale model to produce the final reading c^* .

The entire model is trained either end-to-end or in two stages, with the total loss $\mathcal{L}_{\text{total}}$ defined in Eq. (2) in the Experimental Setup section. This unified formulation clearly separates the roles of feature extraction, task-specific detection, spatial guidance, and geometric reasoning, providing a concise mathematical basis for the proposed TGNNet framework.

4 Dataset

This study employed three datasets for validation: two public datasets were used to examine the role of TAnet in object detection, and one self-constructed dataset was utilized to verify the feasibility of the proposed task-guided learning method.

4.1 Public Datasets

Three publicly available datasets are used: DOTA [28], VisDrone [29], and AU-AIR [30]. DOTA contains 409,471 instances across 15 categories, with diverse object scales and orientations. VisDrone provides 10,209 images and 2.6 million bounding boxes for 10 object classes (e.g., pedestrians, vehicles), captured from UAVs in urban scenes. AU-AIR is a multi-modal dataset with over 32,000 annotated boxes for 8 categories, suitable for dynamic outdoor object detection.

4.2 Self-Constructed USWGD Dataset

We also build the Unmanned Aerial Vehicle Ship Water Gauge Dataset (USWGD) for draft measurement. Images were captured using a DJI Mavic Air2 drone at a distance of 20–40 m from ships and an altitude of 5–20 m above water, with a resolution of 1920×1080 pixels. A total of 1435 video clips yield approximately 15,000 extracted frames.

Annotation combines manual and automated procedures to ensure accuracy. The dataset covers 11 categories (digits 0–9, 'M', and a special waterline class '12'), with 225,485 odd characters, 91,528 even characters, and additional waterline annotations. All labels are cross-validated. To avoid overfitting and enable rigorous evaluation, we randomly split the data into training, validation, and test sets in a 5:2:3 ratio, and all reported results are from the test set.

5 Experiments and Results

This section presents comprehensive experiments conducted on the aforementioned datasets from a UAV perspective to validate the effectiveness of the proposed method. We first describe the experimental setup, including implementation details and evaluation metrics. Ablation studies are then carried out to verify the feasibility of individual modules. Finally, by comparing the proposed task-guided model with existing state-of-the-art approaches, the advantages and innovations of our method are further demonstrated.

5.1 Experimental Setup

We conducted all experiments on a Windows 10 system using PyTorch, with an Intel(i7) 3.60 GHz CPU and an NVIDIA RTX4070 GPU (12 GB VRAM). During training, Mosaic and Mixup data augmentation were applied, with special care to preserve waterline edge integrity. The model was trained for 200 epochs with a batch size of 16 using an SGD optimizer (initial learning rate 0.001, momentum 0.9, weight decay 0.0005) and a cosine annealing scheduler.

The training loss follows the standard YOLO multi-task formulation, comprising three components: classification loss ($\mathcal{L}_{cls}$), bounding box regression loss ($\mathcal{L}_{box}$), and distribution focal loss ($\mathcal{L}_{dfl}$). In the standard YOLOv8 implementation, $\mathcal{L}_{box}$ is specifically realized as an IoU-based loss (Complete IoU, CIoU),

which accounts for overlap area, center distance, and aspect ratio. The total loss is a weighted sum:

$$\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{box}\mathcal{L}_{box} + \lambda_{dfl}\mathcal{L}_{dfl}, \quad (2)$$

where the coefficients follow the default YOLOv8 configuration.

To complement the existing detection metrics, this work additionally employs a keypoint-specific evaluation approach that quantifies localization accuracy through pixel-level distance errors between predicted and annotated ground truth points. The object keypoint similarity (OKS) is computed using the formula [31]:

$$OKS = \frac{\sum_i [\exp(-d_i^2/2s^2\kappa_i^2) \cdot \delta(v_i > 0)]}{\sum_i [\delta(v_i > 0)]} \quad (3)$$

where d_i denotes the Euclidean distance between the predicted and ground truth coordinates of the i -th keypoint, s represents the object scale (calculated as the square root of the bounding box area of the waterline detection), κ_i is a decay constant for the i -th keypoint derived from dataset statistics to reflect annotation difficulty, v_i indicates the visibility label of the keypoint, and δ is a judgment function. The OKS ranges from 0 to 1, with higher values indicating more accurate predictions. In keypoint detection tasks, we use OKS as the criterion for determining whether a detection result is considered positive: when $OKS \geq T$ (a set threshold), it is regarded as a positive sample, based on which the average precision (AP) under the corresponding threshold can be calculated. By integrating AP values across multiple thresholds, the mAP is obtained. The mAP_{75} reported in our experiments corresponds to the AP value when the OKS threshold is set to $T = 0.75$.

The main hyperparameters are summarized in Table 1.

Table 1: Key hyperparameters used in training.

Parameter	Value
Optimizer	SGD
Initial learning rate	0.001
Learning rate scheduler	Cosine annealing
Momentum	0.9
Weight decay	0.0005
Batch size	16
Epochs	200
Data augmentation	Mosaic, Mixup

Based on the above experimental setup, we conduct a task-guided task experiment to systematically validate the feasibility and effectiveness of the joint learning framework, in which a YOLO-based character detection task guides keypoint detection, using the Ship-D dataset. Through comparison with the stepwise keypoint detection approach of YOLOv8n-Pose, the performance of the models is evaluated from multiple dimensions, including precision (P), recall (R), mAP, parameter count (Params), and inference speed, etc. Furthermore, ablation studies on the TAnet module are conducted to further analyze its impact on feature extraction and detection performance across different architectures. Experimental results demonstrate that the proposed guiding mechanism significantly enhances the accuracy and efficiency of ship draft mark keypoint detection while maintaining a low parameter count.

5.2 Comparative Experiments

Based on comprehensive experimental results across five mainstream detection models (YOLOv8n, YOLOv12n [32], Transformer [33], FCOS [34], and CenterNet++ [35]) on the DOTA, VisDrone, and AU-AIR datasets, our proposed TAnet module demonstrates significant performance improvements when integrated into various backbone networks.

Table 2 shows the experimental results. Our proposed TAnet module consistently enhances detection performance across all five models and three datasets. Compared to baseline models, TAnet achieves average improvements of 2.8% in recall and 3.6% in mean average precision (mAP), while maintaining comparable computational efficiency with only marginal variations in frames per second (FPS). Notably, TAnet outperforms both the original DAnet and baseline models in most evaluation scenarios, delivering an average mAP gain of 0.3% across all tested configurations. Particularly impressive is the performance on the VisDrone dataset, where TAnet integrated with YOLOv8n achieves a 4.4% mAP improvement over the baseline (from 42.1% to 46.5%) while simultaneously increasing FPS from 34.0 to 39.1. Similarly, on the DOTA dataset, YOLOv8n with TAnet shows a 3.4% mAP gain. For the Transformer architecture, TAnet delivers substantial recall improvements on DOTA.

Table 2: Performance comparison of various detection models with and without the TAnet module on DOTA, VisDrone and AU-AIR.

Method	Backbone	DOTA			VisDrone			AU-AIR		
		R%	mAP%	FPS	R%	mAP%	FPS	R%	mAP%	FPS
Yolov8n	C2f	73.2	75.1	37.5	44.3	42.1	34.0	44.9	45.1	46.2
	C2f+DAnet	73.5	78.2	36.4	45.6	46.2	36.2	45.1	45.7	44.4
	C2f+TAnet	74.1	78.5	37.3	45.6	46.5	39.1	45.6	46.4	44.3
Transformer	CNN	74.2	72.1	11.2	52.1	48.3	27.3	45.5	45.7	23.2
	CNN+DAnet	75.6	75.1	13.8	53.8	49.6	28.1	45.8	45.9	23.0
	CNN+TAnet	77.8	75.6	13.5	54.2	49.6	23.2	46.0	46.2	23.3
FCOS	RESnet101	63.2	62.3	18.3	36.4	35.4	33.5	38.2	38.0	30.2
	RESnet101+DAnet	65.2	63.8	19.3	37.5	36.7	34.5	38.0	38.7	31.0
	RESnet101+TAnet	65.8	64.1	20.4	38.8	36.7	35.2	38.3	39.0	29.9
CenterNet++	HG-104	64.3	68.2	19.3	31.2	33.4	32.1	30.5	30.7	28.6
	HG-104+DAnet	66.0	69.1	20.3	32.4	36.4	34.2	30.9	31.1	28.4
	HG-104+TAnet	66.1	68.9	20.1	33.3	36.8	35.3	31.6	32.1	28.2
Yolov12n	C3k2	73.3	75.2	37.4	44.4	42.2	33.9	45.1	45.2	46.1
	C3k2+DAnet	73.6	78.3	36.3	45.7	46.3	36.1	45.2	45.8	44.3
	C3k2+TAnet	74.2	78.8	37.4	45.7	46.7	39.1	45.7	46.6	44.4

Notably, all improvements in this paper are developed based on the YOLOv8 architecture. Due to the high structural similarity among YOLO series models, our proposed method can be conveniently transferred to other YOLO variants. Experimental results show that although YOLOv12n integrated with TAnet achieves marginally better performance on some metrics, these improvements are modest and inconsistent across different datasets. Considering YOLOv8's competitiveness as a baseline architecture, its widespread adoption

in the academic community, and its strong reproducibility, we ultimately select YOLOv8n as the foundation for subsequent experiments and the development of our Task Guidance Network (TGNet). This choice helps ensure that our contributions are built upon a reliable and widely recognized benchmark, facilitating fair comparisons with existing methods. The comprehensive experimental results above fully validate the effectiveness and generalization capability of the TAnet module in enhancing object detection performance across different architectures and application scenarios.

This ablation study validates the effectiveness of the proposed TAnet module on the Ship-D test set. As shown in the Table 3, the incorporation of TAnet improves precision across most character categories in all four detection models-YOLOv8n, Transformer, FCOS, and CenterNet++-demonstrating the module's strong generalization and effectiveness. Particularly notable is its performance on the critical waterline target "12": TAnet increases the precision of "12" in YOLOv8n from 92.8% to 94.0%, significantly outperforming both the baseline and DANet variants, while achieving top or competitive results across the other models. These results fully indicate that the TAnet module specifically enhances the perception of key features in ship waterlines, substantially improving the accuracy and reliability of draft mark recognition.

Table 3: The P comparison of 13 characters (0–10, M, 12) with/without TAnet on Ship-D test set.

Method	Backbone	0	1	2	3	4	5	6	7	8	9	M	12
Yolov8n	C2f	88.0	89.1	91.9	88.1	92.0	83.9	90.8	83.9	92.1	83.6	89.8	92.8
	C2f+DAnet	89.1	91.5	92.0	88.2	93.0	85.9	91.6	84.9	93.9	85.8	91.0	93.7
	C2f+TAnet	85.4	92.0	92.4	88.3	93.1	85.4	91.7	85.4	93.7	85.7	91.2	94.0
Transformer	CNN	88.2	89.1	90.7	88.1	91.6	85.2	88.8	83.0	91.1	82.8	90.7	91.2
	CNN+DAnet	88.8	89.8	91.0	88.6	92.1	86.1	89.0	83.0	92.0	83.1	91.0	91.3
	CNN+TAnet	89.1	89.8	91.3	88.8	91.8	86.0	89.4	83.4	91.9	83.1	91.3	91.5
FCOS	RESnet101	85.3	83.0	85.5	82.0	85.4	81.9	83.2	79.1	85.6	80.1	88.2	84.1
	RESnet101+DAnet	85.4	82.7	86.0	83.0	85.9	82.4	83.9	79.7	86.0	81.4	88.0	85.8
	RESnet101+TAnet	85.7	83.8	86.4	83.1	85.7	82.8	83.8	79.9	86.2	81.6	87.9	86.0
CenterNet++	HG-104	84.4	81.3	81.1	79.8	83.9	79.0	82.6	79.8	81.2	81.3	82.7	83.3
	HG-104+DAnet	84.4	81.7	81.4	80.0	83.6	79.1	82.8	80.2	81.8	81.2	82.9	84.4
	HG-104+TAnet	85.0	81.9	81.6	80.1	83.4	79.2	83.0	80.3	82.4	82.0	82.9	84.2

To verify the role of the proposed TAnet in texture feature extraction, we incorporated the DANet module and TAnet module, respectively, into the feature extraction components of four target detection models (YOLOv8, Transformer, FCOS, and CenterNet++). We then conducted visual analysis on the feature maps output at this stage and generated corresponding heatmaps (as shown in Fig. 7). The results indicate that the second column presents the original heatmaps without any additional modules, while the third and fourth columns display the results after introducing DANet and TAnet, respectively. It can be clearly observed that after integrating TAnet, the contour details of character and waterline regions in the heatmaps are more abundant, with significantly enhanced texture responses. This demonstrates that TAnet can effectively improve the texture feature responses of character and waterline regions, highlight key texture structures, and simultaneously suppress unnecessary background information. This mechanism enhances the discriminative ability of features, enabling fine structures such as watermarks to be more clearly preserved in the feature maps. Consequently, it provides more distinguishable feature representations for subsequent detection tasks.

The waterline, as a typical weak-texture and low-contrast target, often suffers from its meaningful signals being overwhelmed by complex background interference such as waves, watermarks, and hull rust in global feature representations. To address this issue, we introduce two complementary strategies. On one hand, the

TAnet branch serves as a feature detail enhancer by explicitly learning an attention map that significantly enhances the texture characteristics in the waterline region while suppressing local background distractions. On the other hand, the task-guided mechanism leverages character detection results to crop a focused feature patch around the waterline area, further reducing interference from irrelevant regions.

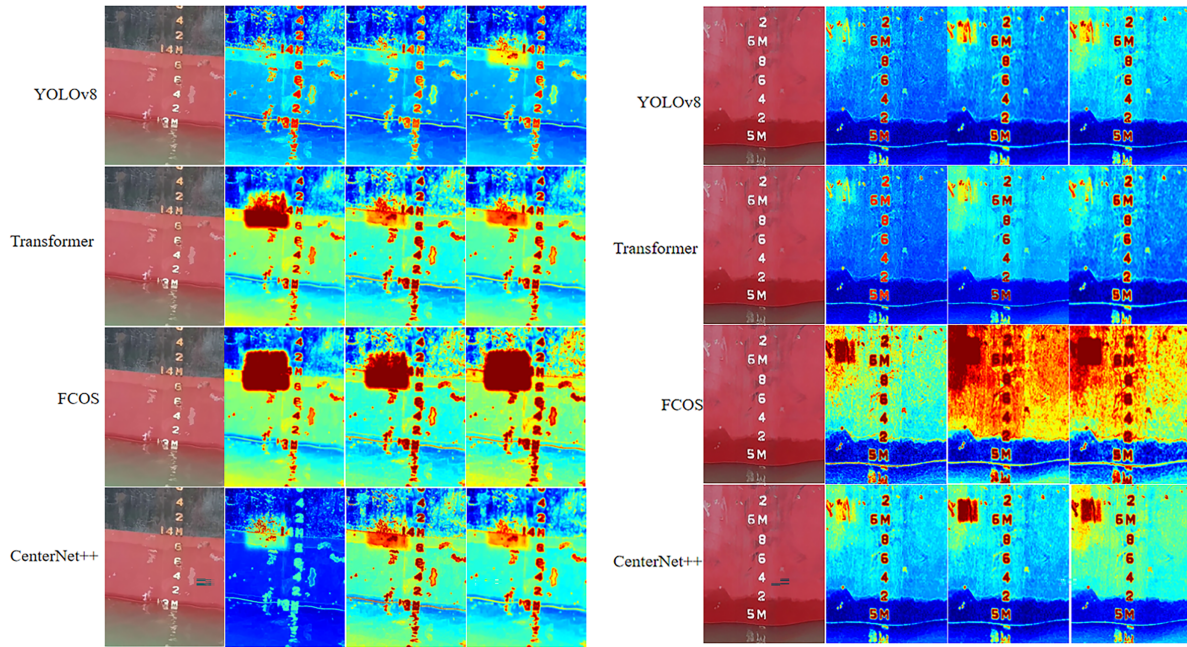


Figure 7: There are two sub-figures, Left and Right, illustrating the visualization results of two distinct sample images. In each sub-figure, each row corresponds to a network (YOLOv8, Transformer, FCOS, and CenterNet++), while each column from left to right sequentially corresponds to the original input, the feature map of the baseline model, the feature map of baseline+DAnet, and the feature map of baseline+TAnet, respectively.

Fig. 8 presents visualization examples of feature maps under three complex scenarios (from top to bottom: dark watermark scenes, heavy fog backgrounds, and night scenes). The model shown in Fig. 5 was applied for testing. In each row of the comparison, the second column displays feature maps without the TAnet, while the third column shows the corresponding results with the TAnet incorporated. With the addition of the TAnet, waterline feature details become more prominent, clearly demonstrating the significant optimization effect of TAnet on feature extraction in complex scenarios.

To further validate the superiority of our proposed TAnet, we compare it against several representative lightweight attention mechanisms: SE(Squeeze-and-Excitation) [36], CBAM (Convolutional Block Attention Module) [37], ECA(Efficient Channel Attention) [38], and CoordAttention (CA) [39]. All modules are integrated into the same YOLOv8n backbone and trained under identical settings on the Ship-D test set. As shown in Table 4, TAnet achieves the highest mAP_{75} of 93.8%, outperforming the second-best method(ECA) by 1.4%. Notably, SE introduces only 0.5% additional parameters [36], while ECA adds merely 80 parameters and 4.7×10^{-4} GFLOPs against ResNet50. Although TAnet introduces slightly more overhead (4.6M parameters) compared to these lightweight modules, it delivers the best performance by effectively integrating channel, spatial, and texture attention, which is crucial for weak-texture waterline detection.

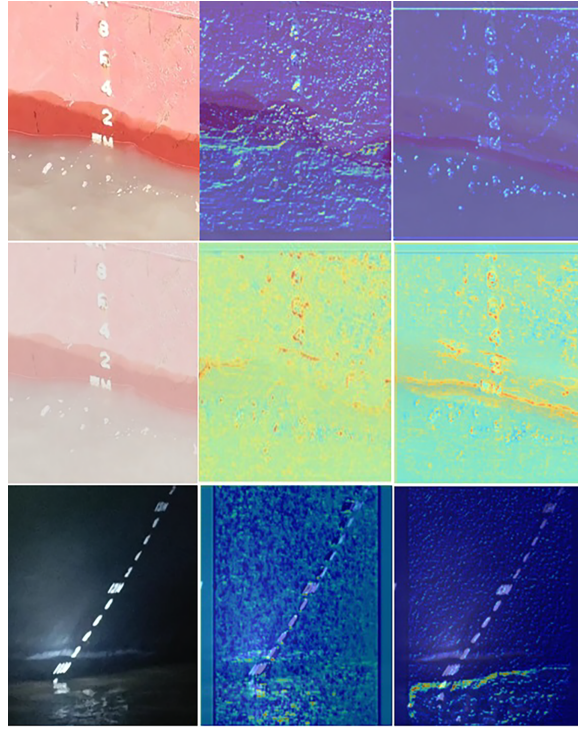


Figure 8: Visualization examples of feature maps under three complex scenarios (top to bottom: scene with dark watermarks, heavy fog background, and night scene). In each row, the second column presents feature maps without the TANet layer incorporated, while the third column shows those with the corresponding TANet layer. This comparison intuitively illustrates the optimization effect of TANet on feature extraction in complex scenarios.

Table 4: Comparison of different attention modules on the Ship-D test set.

Attention Module	P/%	R/%	mAP ₇₅ /%	Params (M)
Baseline	88.2	86.7	87.9	3.3
+SE	90.5	88.6	90.1	3.4
+CBAM	91.2	89.4	91.0	3.5
+ECA	92.0	90.1	92.4	3.4
+CA	91.5	89.8	91.7	3.4
+DAnet	91.8	90.2	91.4	4.3
+TAnet (Ours)	94.0	92.3	93.8	4.6

5.3 Task-Guided Task Experiment

To isolate the contribution of each component and ensure fair evaluation, we construct three controlled baselines: (1) Full Image, where the keypoint detection branch processes the entire feature map without cropping; (2) ROI, which crops the feature map using the bounding box of the lowest detected character (instead of our adaptive region); and (3) Guide (w/o TANet), which retains the task guidance mechanism but removes the TANet module. All baselines share the same backbone, training epochs, and data augmentation as TGNet.

As shown in Table 5, Full Image achieves the lowest performance (87.9% mAP₇₅), confirming that direct keypoint regression on the full feature map is suboptimal for waterline localization. The ROI baseline

improves mAP_{75} to 89.5%, indicating that character-based spatial cropping provides useful prior information. TGNet surpasses ROI by 4.3%, demonstrating that our adaptive feature-level cropping preserves richer contextual information than simple box-based cropping. Guide (w/o TAnet) further improves mAP_{75} to 91.4%, validating the effectiveness of the task guidance mechanism itself. The addition of TAnet boosts performance to 93.8%, confirming that texture attention and task guidance work synergistically to enhance waterline feature extraction.

Table 5: Performance comparison on the Ship-D test set (mean $\pm$ std over 5 runs).

Method	Backbone	P/%	R/%	$mAP_{75}/\%$	Params (M)	FLOPs (G)	FPS
YOLOv8n-pose (Full Image)	C2f	88.2 ± 0.6	86.7 ± 0.7	87.9 ± 0.5	3.3	8.7	76
YOLOv8n-pose+ROI	C2f	90.1 ± 0.5	88.3 ± 0.6	89.5 ± 0.4	3.4	8.9	71
YOLOv8n-pose+Guide (w/o TAnet)	C2f	91.8 ± 0.4	90.2 ± 0.5	91.4 ± 0.4	4.3	10.2	62
TGNet	C2f+TAnet	94.0 ± 0.3	92.3 ± 0.4	93.8 ± 0.3	4.6	11.5	56

To comprehensively evaluate model efficiency, we report FLOPs and inference latency alongside parameter count and FPS. FLOPs are measured using the thop library at 640×640 input resolution, and latency is averaged over 1000 forward passes on the RTX4070 GPU. Compared to the baseline YOLOv8n-pose, TGNet introduces a 39% parameter increase (3.3M to 4.6M) and a 26% FPS reduction (76 to 56), yet maintains real-time performance for UAV deployment. These results demonstrate that TGNet's superiority stems from the collaborative effect of task-guided spatial constraint and texture-enhanced feature learning, rather than from feature cropping alone.

We further compare our method with four mainstream models (YOLOv8, Transformer, FCOS, and CenterNet++). Fig. 9 presents two representative cases, each with five subfigures corresponding to our method and the four baselines—showing waterline contours obtained from keypoint detection and curve fitting. Our method consistently outperforms all baselines, which exhibit various defects such as shadow misclassification, positional deviation, or missing keypoints, while ours achieves superior accuracy and robustness.

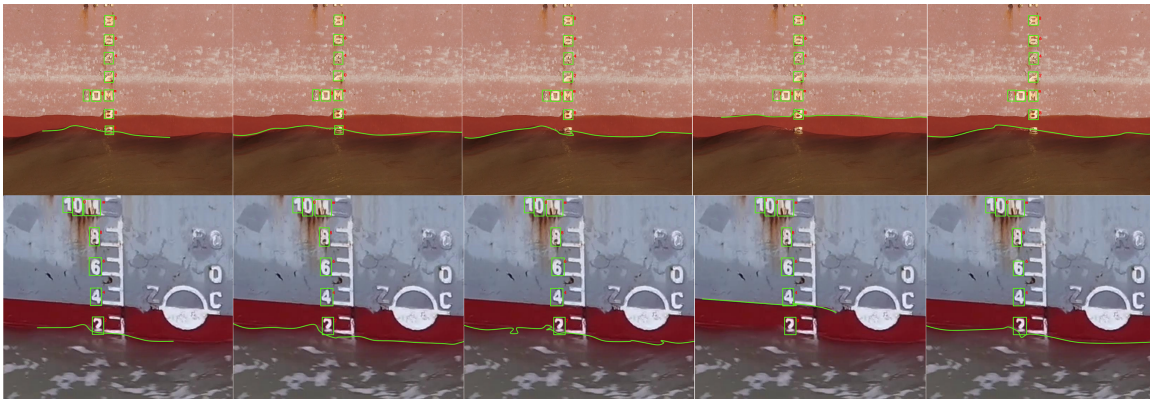


Figure 9: Comparison of waterline detection results for different models. The leftmost image of the subgraph displays the keypoint detection method proposed in this study, while the last four images show the results of the applied detection models YOLOv8n, Transformer, FCOS, and CenterNet++.

To systematically validate each component, we conduct ablation studies on the Ship-D test set. All experiments share identical training settings for fair comparison.

As shown in Table 6, the Full Image baseline achieves 87.9% mAP₇₅. ROI cropping improves it to 89.5%, while our adaptive cropping further boosts to 91.4%, confirming the importance of preserving contextual information. Using ground-truth character boxes instead of predicted ones yields only 0.6% higher mAP₇₅, demonstrating robustness to detection errors. The two-stage training outperforms end-to-end training by 1.4%, validating our training strategy. Regarding crop size, the half-size feature map achieves the best performance, while further reduction causes a slight drop. For attention components, removing TAM causes the largest degradation (from 93.8% to 92.1%), confirming texture attention as the most critical module for waterline feature enhancement.

Table 6: Ablation studies on the Ship-D test set (mean $\pm$ std over 5 runs).

Experiment	Setting	P/%	R/%	mAP ₇₅ /%
Full Image	No cropping	88.2 $\pm$ 0.6	86.7 $\pm$ 0.7	87.9 $\pm$ 0.5
ROI	Character box cropping	90.1 $\pm$ 0.5	88.3 $\pm$ 0.6	89.5 $\pm$ 0.4
Guide (w/o TAnet)	Adaptive cropping	91.8 $\pm$ 0.4	90.2 $\pm$ 0.5	91.4 $\pm$ 0.4
Guide+GT	Ground-truth boxes	92.2 $\pm$ 0.4	90.6 $\pm$ 0.5	92.0 $\pm$ 0.4
Guide+Pred	Predicted boxes	91.8 $\pm$ 0.4	90.2 $\pm$ 0.5	91.4 $\pm$ 0.4
E2E	End-to-end training	92.7 $\pm$ 0.5	90.8 $\pm$ 0.6	92.4 $\pm$ 0.4
Two-stage	Two-stage training	94.0 $\pm$ 0.3	92.3 $\pm$ 0.4	93.8 $\pm$ 0.3
1 $\times$	Full feature map	88.2 $\pm$ 0.6	86.7 $\pm$ 0.7	87.9 $\pm$ 0.5
1/2 $\times$	Half feature map	94.0 $\pm$ 0.3	92.3 $\pm$ 0.4	93.8 $\pm$ 0.3
1/4 $\times$	Quarter feature map	93.1 $\pm$ 0.4	91.0 $\pm$ 0.5	92.5 $\pm$ 0.4
w/o PAM	Remove position attention	93.5 $\pm$ 0.4	91.6 $\pm$ 0.5	93.0 $\pm$ 0.4
w/o CAM	Remove channel attention	93.3 $\pm$ 0.4	91.2 $\pm$ 0.5	92.8 $\pm$ 0.4
w/o TAM	Remove texture attention	92.6 $\pm$ 0.5	90.5 $\pm$ 0.6	92.1 $\pm$ 0.4
TGNet (Full)	PAM+CAM+TAM	94.0 $\pm$ 0.3	92.3 $\pm$ 0.4	93.8 $\pm$ 0.3

5.4 Application Testing

To validate the practical feasibility of the proposed method, we integrated it into an intelligent ship draft mark calculation system. The system automatically computes water level values by detecting draft mark characters and waterline key information, following the computational workflow outlined in Algorithm 1, ultimately determining the water level through the intersection point of the waterline and the fitted curve of the draft mark.

To thoroughly demonstrate the superiority of our method, we selected 120 drone-captured ship draft mark videos as the test dataset, covering diverse weather and illumination conditions including clear days, foggy scenes, and nighttime. The system processes the videos at a sampling rate of approximately 2 FPS, and the average of the computational results from all sampled frames is taken as the final system reading for each video segment. By comparing the system readings with professional manual measurements, we evaluate the performance using both categorical accuracy and continuous error metrics including MAE, RMSE, and MedAE.

The experimental results in Table 7 clearly demonstrate the advantages of our unified detection approach. While FCOS+pose, Transformer+pose, and YOLOv8n+pose represent conventional methods that perform character detection and waterline detection as separate tasks, TGNet achieves superior performance with the lowest parameter count of 4.6M and shortest processing time of 32.4 ms. Specifically, TGNet achieves the highest accuracy of 88.3% with MAE of 1.02 cm and RMSE of 1.68 cm, significantly

outperforming all comparison methods. The median absolute error of 0.71 cm indicates that most predictions are within sub-centimeter accuracy. This represents a significant improvement over the best separate approach, which requires $3.7\times$ more parameters and achieves 8.3% lower accuracy. The performance gap can be attributed to the synergistic effect of joint feature learning in our unified framework, which enables more consistent and reliable detection while reducing computational overhead from redundant feature extraction in separate models.

Table 7: Comparison of water gauge reading accuracy across different methods and scenarios.

Model-Based	Params (M)	MAE (cm)	RMSE (cm)	MedAE (cm)	Accuracy	Time (ms)
FCOS+pose	8.1	3.42	5.81	2.15	50.8%	47.4
Transformer+pose	17.2	1.85	2.96	1.23	80.0%	60.2
YOLOv8n+pose	6.2	2.31	3.87	1.56	75.8%	55.1
TGNet (Ours)	4.6	1.02	1.68	0.71	88.3%	32.4

6 Conclusion

This study proposes a novel TGNet for high-precision, real-time ship waterline detection in complex maritime environments. By deeply integrating a TAnet with a base detection model, TGNet innovatively establishes an inter-task guidance mechanism that utilizes spatial information from character recognition to guide and optimize waterline keypoint localization. The TAnet module significantly enhances the discriminative capability of waterline features through the fusion of channel, spatial, and texture attention mechanisms.

Experiments on multiple large-scale public datasets and a self-built ship draft dataset demonstrate the method's outstanding effectiveness and generalization capability. The TAnet module improves the average recall of various detection models by 2.8% and increases average precision by 3.6%. The complete TGNet system achieves a recall rate of 92.3% and an mAP_{75} of 93.8% on the self-built dataset, while maintaining real-time processing performance at 56 FPS.

In practical applications, TGNet demonstrates exceptional accuracy and robustness in intelligent draft reading systems, achieving a water level calculation accuracy of 88.3% (error < 0.02 m) and an approximate accuracy rate of 95% (error < 0.05 m). Its lightweight design and low-latency characteristics fully meet the edge deployment requirements of UAV platforms.

In summary, this research provides an efficient and reliable AI solution for automated ship draft monitoring, offering valuable insights for multi-task learning in maritime computer vision. However, the method still has several limitations. First, missed detection of draft characters directly affects the task guidance effectiveness, representing a major bottleneck in the current architecture; second, the two-stage training pipeline limits feature sharing and computational efficiency; third, its generalization to different ship types and extreme weather conditions (e.g., fog, rain, strong glare) has not been systematically verified under complex field maritime environments. Future work will focus on end-to-end joint training with adaptive loss balancing, lightweight model compression for edge deployment, and robust data augmentation to handle diverse maritime scenarios. Addressing these issues will further enhance the system's practicality and performance.

Acknowledgement: The authors acknowledge the valuable comments and suggestions that helped improve this paper.

Funding Statement: This work was supported by the National Natural Science Foundation of China under Grant No. 61573183, and the Scientific Research Project of Colleges and Universities in Anhui Province under Grant No. 2022AH051584.

Author Contributions: Jinlin Chen: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft, visualization. Yiquan Wu: conceptualization, methodology, resources, supervision, project administration, funding acquisition, writing—review & editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Due to privacy and security concerns regarding the captured ship imagery, the UAV-based USWGD dataset is not publicly available. Access may be granted upon reasonable request to the corresponding author with appropriate approvals.

Ethics Approval: This study did not involve human participants or animal subjects. Ethical approval is not applicable.

Conflicts of Interest: Given his role as Editorial Board Member of this journal, Fa Zhu had no involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. The authors declare no other conflicts of interest.

References

- Schmidt GH. Linearized stern flow of a two-dimensional shallow-draft ship. *J Ship Res.* 1981;25(4):236–42. doi:10.5957/jsr.1981.25.4.236.
- Dutta HPJ, Bhuyan MK, Karsh RK, Alfarhood S, Safran M. Multiscale attention-based hand keypoint detection. *IEEE Trans Instrum Meas.* 2024;73:1–11. doi:10.1109/TIM.2024.3413196.
- Fu J, Liu J, Tian H, Li Y. Dual attention network for scene segmentation. In: *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun 15–20; Long Beach, CA, USA.* p. 3141–9.
- Tsujii T, Yoshida H, Iiguni Y. Automatic draft reading based on image processing. *Opt Eng.* 2016;55(10):104104. doi:10.1117/1.oe.55.10.104104.
- Liu Y, Zhang W, Wang H, Chen X, Li M. Automatic waterline detection for unmanned surface vehicles using visual and infrared images. *Ocean Eng.* 2018;158:18–25. doi:10.1016/j.oceaneng.2018.03.085.
- Ding X, Nunziata F, Li X, Migliaccio M. Performance analysis and validation of waterline extraction approaches using single- and dual-polarimetric SAR data. *IEEE J Sel Top Appl Earth Obs Remote Sens.* 2015;8(3):1019–27. doi:10.1109/JSTARS.2014.2362511.
- Zheng G, Zhou Y, Liu B, Zhou L, Jiang H, Wan X, et al. Difference-focusing fusion decision method: an ensemble learning framework and its application in improving deep learning sea-land segmentation for waterline extraction in synthetic aperture radar imagery. *IEEE Trans Geosci Remote Sens.* 2024;62:1–14. doi:10.1109/TGRS.2024.3446317.
- Zhang L, Wang Y, Chen X, Liu Z, Li M. A novel deep learning-based method for detecting ship load lines and calculating ship drafts. *Sensors.* 2021;21(4):1325. doi:10.3390/s21041325.
- Chen Z, Wang L, Liu Y, Zhang H, Li W. A CNN-based automatic ship draft reading system. *Ocean Eng.* 2021;235(2):109376. doi:10.1016/j.oceaneng.2021.109376.
- Liu X, Wu J, Liao Y, Tu X. Column-based waterline detection for lightweight ship draft reading. *IEEE Trans Instrum Meas.* 2022;71:1–10. doi:10.1109/tim.2022.3154840.
- Zhou C, Jiang M, Kong J. BGTracker: cross-task bidirectional guidance strategy for multiple object tracking. *IEEE Trans Multimed.* 2023;25:8132–44. doi:10.1109/TMM.2023.3256761.
- Zhu Y, Qu J, Dong W, Li Y. Fusion-driven task mutual-guidance network for few-shot hyperspectral image classification. *IEEE Trans Multimed.* 2026;28:1211–25. doi:10.1109/TMM.2025.3632632.

13. von Braun M-S, Frenzel P, Käding C, Fuchs M. Utilizing mask R-CNN for waterline detection in canoe sprint video analysis. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2020 Jun 14–19; Seattle, WA, USA. p. 3826–35. doi:10.1109/CVPRW50498.2020.00446.
14. Wang B, Liu Z, Shen Y, Wang S. Sdrnet: a hybrid approach with deep convolutional networks for ship draft reading. *Measurement*. 2025;247(3):116798. doi:10.1016/j.measurement.2025.116798.
15. Zhang B, Yin Y, Ma K, Wang H. Multi-scale feature fusion keypoint detection network for ship draft line localization. *Sci Rep*. 2025;15(1):26397. doi:10.1038/s41598-025-10594-x.
16. Wang Z, Zheng L, Liu Y, Li Y, Wang S. Towards real-time multi-object tracking. arXiv:1909.12605. 2019.
17. Brasó G, Leal-Taixé L. Learning a neural solver for multiple object tracking. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 6246–56. doi:10.1109/CVPR42600.2020.00628.
18. Zhang Y, Wang C, Wang X, Zeng W, Liu W. Fairmot: on the fairness of detection and re-identification in multiple object tracking. *Int J Comput Vis*. 2020;129:3069–87.
19. Liu S, Johns E, Davison AJ. End-to-end multi-task learning with attention. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA. p. 1871–80.
20. Misra I, Shrivastava A, Gupta A, Hebert M. Cross-stitch networks for multi-task learning. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27–30; Las Vegas, NV, USA. p. 3994–4003. doi:10.1109/CVPR.2016.433.
21. Wang Z, Li Y, Chen X, Lim S-N, Torralba A, Zhao H, et al. Detecting everything in the open world: towards universal object detection. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–23; Vancouver, BC, Canada. p. 11433–43. doi:10.1109/CVPR52729.2023.01100.
22. Xu Y, Sun Y, Yang Z, Miao J, Yang Y. H2FA R-CNN: holistic and hierarchical feature alignment for cross-domain weakly supervised object detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 14309–19. doi:10.1109/CVPR52688.2022.01393.
23. Chen X, Zhang L, Wang B, Li Q. Mitigating task interference in multi-task learning via explicit task routing with non-learnable primitives. In: Proceedings of the European Conference on Computer Vision (ECCV); 2025 Oct 19–23; Honolulu, HI, USA. p. 890–908.
24. Xu X, Li Y, Wang H, Zhang L, Chen W. Multi-task learning for simultaneous ship detection and waterline segmentation in inland waterways. *Remote Sens*. 2022;14(3):767. doi:10.3390/rs14030767.
25. Fang A, Wang Y, Yin X, Wang P, Li Z, Liu Z. DetSegNet: a high-precision water gauge level detection network based on detection and segmentation. *J Hohai Univ*. 2024;52(2):28–34. doi:10.3876/j.issn.1000-1980.2024.02.004.
26. Zhang B, Li J, Tang H, Liu X. Smart ship draft reading by dual-flow deep learning architecture and multispectral information. *Sensors*. 2024;24(17):5580. (In Chinese). doi:10.3390/s24175580.
27. Steccanella L, Bloisi J, Blum D, Farinelli A. Deep learning waterline detection for low-cost autonomous boats. In: *Intelligent autonomous systems 15*. Berlin/Heidelberg, Germany: Springer; 2018. p. 613–25.
28. Xia G-S, Bai X, Ding J, Zhu Z, Belongie S, Luo J, et al. DOTA: a large-scale dataset for object detection in aerial images. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 3974–83. doi:10.1109/CVPR.2018.00418.
29. Wen L, Zhu W, Du D, Bian X, Ling H, Hu Q, et al. VisDrone-MOT2019: the vision meets drone multiple object tracking challenge results. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW); 2019 Oct 27–28; Seoul, Republic of Korea. p. 189–98. doi:10.1109/ICCVW.2019.00028.
30. Bozcan I, Kaya E. AU-AIR: a multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA); 2020 May 31–Aug 31; Paris, France.
31. Lin T-Y, Maire M, Belongie SJ, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. arXiv:2204.06806. 2014.
32. Al Rabbani Alif M, Hussain M. YOLOv12: a breakdown of the key architectural features. arXiv:2502.12524. 2025.

33. Xu J, Shi L, Li S, Zhang Y, Zhao G, Shi Y, et al. PointFormer: keypoint-guided transformer for simultaneous nuclei segmentation and classification in multi-tissue histology images. *IEEE Trans Image Process.* 2025;34:2883–95. doi:10.1109/TIP.2025.3565184.
34. Tian Z, Shen C, Chen H, He T. FCOS: a simple and strong anchor-free object detector. *IEEE Trans Pattern Anal Mach Intell.* 2022;44(4):1922–33. doi:10.1109/TPAMI.2020.3032166.
35. Guo H, Yang X, Wang N, Gao X. A CenterNet++ model for ship detection in SAR images. *Pattern Recognit.* 2021;112(7):107787. doi:10.1016/j.patcog.2020.107787.
36. Hu J, Shen L, Albanie S, Sun G, Wu E. Squeeze-and-excitation networks. *IEEE Trans Pattern Anal Mach Intell.* 2020;42(8):2011–23. doi:10.1109/TPAMI.2019.2913372.
37. Lyu Z, Wang C, Sun X, Zhou Y, Ni X, Yu P. Real-time ship detection system for wave glider based on YOLOv5s-lite-CBAM model. *Appl Ocean Res.* 2024;144(13):103833. doi:10.1016/j.apor.2023.103833.
38. Kakade MN. RosMS-ECDBTM: efficient channel attention enabled distributed deep learning model for mental state detection using EEG signal. *Neurocomputing.* 2025;657:131559. doi:10.1016/j.neucom.2025.131559.
39. Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design. In: *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; 2021 Jun 20–25. Nashville, TN, USA. p. 13708–17. doi:10.1109/CVPR46437.2021.01350.