JOURNAL OF MARITIME RESEARCH

Vol XXIII. No. II (2026) pp 290–298

ISSN: 1697-4840, www.jmr.unican.es

# Real-Time YOLOv8 Detection and Particle Swarm Optimization Path Planning for COLREGs-Aware Collision Avoidance in Smart-Port Unmanned Surface Vehicle Operations

Nguyen Dinh Thach[1,*]

ARTICLE INFO

*Article history:*
Received 03 Oct 2025;
in revised form 23 Oct 2025;
accepted 11 Nov 2025.

*Keywords:*
Unmanned Surface Vehicles (USVs), Collision Avoidance, YOLOv8 Object Detection, COLREGs Compliance, Particle Swarm Optimization (PSO).



ABSTRACT

Automation and digital transformation are being applied and widely implemented and advancing rapidly at seaports worldwide. With rapid growth, the number of ships entering and leaving the port is increasing, causing significant pollution in the port ecosystem. Therefore, regular monitoring and safety inspections at seaports need to be optimized and modern Artificial Intelligence (AI) technologies applied. Unmanned Surface Vehicles (USVs) are used for monitoring environmental quality with surveying water pollution at ports. However, current methods are often decoupled from object recognition and planning, and do not incorporate safety constraints. In this study, an object recognition method for seaports is proposed, utilizing the YOLOv8 deep learning model for multi-layer detection. Safe-distance calculations and classifying situations are based on COLREGs. The Particle Swarm Optimization (PSO) selects the optimal route and generates waypoints (W) in USV navigation. The experimental model is implemented in Unity 3D simulation software, and the controller is written in MATLAB. The research results show that mAP50 = 0.90 and 39 FPS demonstrate exemplary performance in the simulated smart-port seaport environment.

## 1. Introduction.

The increasing number of ships operating at seaports has led to a large and crowded fleet of vessels. On the other hand, this large number of vessels emit a significant amount of exhaust gases and pollutants into the marine environment. This has a significant impact on the marine ecosystem in these areas and on the people living there. Meanwhile, monitoring of large areas requires a large number of monitoring stations and substantial capital investment. Currently, automation and digital transformation in the mart-port models are being researched and developed. However, some risks are faced by seaports in developing countries, such as cybersecurity, deployment models and interoperability and communications with existing systems (Boadu et al., 2025).

Equipment and systems that incorporate AI and modern technology have been deployed and operated, demonstrating superior results compared to traditional methods. USVs operated in ports help enhance work quality and operational efficiency (Shen et al., 2024). Research and development of USVs for water surfaces are underway to support topographic surveying and mapping, monitoring air and water quality, and detecting water pollution risks, with promising results (Santos, 2025). Indeed, given differing topographic conditions across seaports, collision accidents often occur within narrow port channels (Lyu et al., 2023; Q. Zhu et al., 2024). USVs must operate in compliance with maritime laws and safety regulations. Port operations with high traffic density, poor visibility, and diverse target sizes and speeds require USVs to operate efficiently and accurately. The primary research direction is to utilize machine learning models for target classification. Research has developed obtacles

[1]Assoc. Prof., Lecturer, Faculty of Navigation, Vietnam Maritime University, Haiphong, Vietnam.
[*]Corresponding author: Nguyen Dinh Thach. E-mail Address: thachnd.ddt@vimaru.edu.vn.

detection algorithms based on advanced neural networks to detect objects such as ships, speedboats, and buoys in real time at approximately 30 frames per second (Lee et al., 2018). The proposed YOLO-MRS model is based on the YOLOv8 architecture to improve object detection in complex and data-deficient conditions. The dataset contains 9,000 labeled images across 13 classes with noise combinations. The research results show that mAP50 increases by 1.8% to 7%(Yu et al., 2024). Active (moving) and passive (static) obstacles affect USV operations. The object-recognition method combines Convolutional Neural Network (CNN) based visual classification and Support Vector Machines (SVM) to generate collision-avoidance scenarios. Simulation and experimental results demonstrate that the algorithm can effectively identify situations and safely avoid collisions (Niu et al., 2025).

Collision avoidance problems for USVs enable safe navigation and more efficient energy consumption. Some deep reinforcement learning frameworks for collision avoidance decision-making in complex maritime environments. For USVs, an improved Deep Q-Network (DQN)-based method use NoisyNets, prioritized experience replay, parallel network architecture, Double Q-learning, with dynamic area restriction and state pruning to reduce computational cost and training complexity. A Unity-based test model with scenario diversification enhances the ability to avoid collisions (Fan et al., 2023). Additionally, other studies propose a new collision avoidance decision-making system based on relative velocity to obstacles. A multi-stage optimization model with numerous constraints, including maneuverability and COLREGs (Shaobo et al., 2020). This paper proposes an Optimal Collision Avoidance Point to determine the time and direction of collision avoidance for USVs. The method combines the USV's 2-DOF kinematics and a probabilistic assessment of collision risk within the admissible trajectory to select the direction. Experimental results show high real-time performance (H. Zhu & Ding, 2023a).

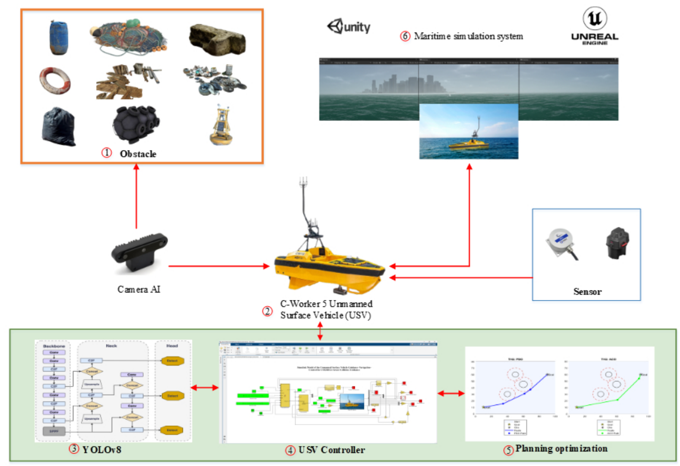
Other dynamic collision avoidance methods are based on fuzzy rules combined with PSO algorithms to optimize collision avoidance parameters. The study addresses the problem of detecting and avoiding collisions with moving objects (Song et al., 2021). The human-like collision avoidance method for USVs combines a deep reinforcement learning model based on the human observational mechanism of seafarers when facing multiple ships. Convention on the International Regulations for Preventing Collisions at Sea (COLREGs) are integrated into the calculation of the Navigation Impact Factor (NIF), and the Velocity Obstacle (VO) is included in the reward function, with carefully designed reward components that balance safety and steering ability (Yang et al., 2024). However, studies often separate vision from planning and rarely integrate the pipeline from recognition to optimal route planning for collision-avoidance for USVs (Guan & Wang, 2023; Ren et al., 2021). Therefore, the new contribution of this study is to combine YOLOv8, for object recognition, with the COLREGs to plan and generate collision avoidance routes for USVs at ports.

The remainder of the paper is organized in Section 2, which outlines the research methodology, and Section 3 analyzes tests of collision detection and avoidance scenarios at ports. Conclusions and further research directions are summarized in Section 4.

## 2. System Architecture and Methodology.

### 2.1. System Architecture.

Figure 1: Integrated USV Perception?Planning?Control Architecture.




Source: Authors.

Figure 1 shows the system architecture with the following steps:

- Block 1: Static and mobile obstacles such as Blue plastic barrel, Fishing Bouys, Rotten Wood,... are considered as input data.
- Block 2: The USV model is built on Unity 3D software with C-Worker 5 parameters.
- Block 3: YOLOv8 model to distinguish obstacles in an underwater environment. Additionally, there is a tracker function to retrieve the target ID and velocity.
- Block 4: USV controller built on Matlab with direction control and collision avoidance functions obtained from optimal scenarios. Waypoint tracking PID/MPC controllers simulate 3-DOF dynamics.
- Block 5: Optimal routes help USVs avoid collisions with multiple waypoints generated.
- Block 6: Test environment in UNITY 3D for real-life scenarios. Weather, wind, and wave data are generated Hardware-in-the-Loop (HIL) simulates in real-time.

### 2.2. Datasets and Preprocessing.

#### 2.2.1. Datasets.

The data used for training the YOLOv8 model was collected by the authors, focusing on real-life obstacles that directly affect the operations of USVs in the port. Figure 2 depicts the basic obstacles in the port.

Figure 2: Integrated USV Perception–Planning–Control.


Source: Authors.

The dataset is labeled into nine classes for collision - avoidance objectives. The images and videos are collected through cameras and simulated data from Unity 3D. In addition, environmental conditions can be simulated, including rain, storms, obstacle density, and varying wave heights. Image denoising is crucial for object classification and detection problems. The data are divided three splits: train/val/test 70/15/15%.

*2.2.2. Preprocessing.*

During the data collection process, numerous cases of duplicate or invalid photos and videos were encountered. Therefore, the Author used some of the following methods:

- Clean and standardize videos and photos: sample at 2–5 fps while maintaining tracking. Filter out blurry, distorted images that are difficult to recognize.
- Size normalization: YOLOv8 uses a 640×640 input size. At the same time, increase the information density for obstacles and the water surface. Combine methods of image color correction and contrast adjustment for different weather conditions.
- Noise reduction: There are many different noise reduction methods, such as Denoising Convolutional Neural Network (DnCNN), UnderWater Convolutional Neural Network (UwCNN), and U-net. In this study, the Author utilizes the DnCNN model for noise reduction, which offers the advantage of processing complex images with high resolution and efficiency. The structure of the DnCNN model is shown in Figure 3.

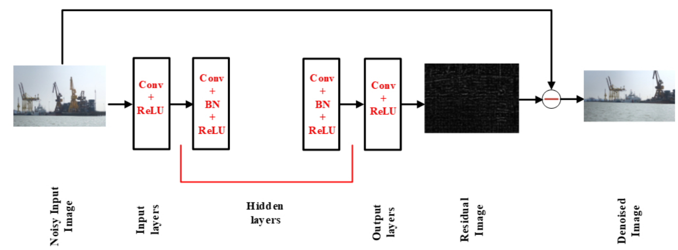
Degradation model (additive noise):

$$y = x + v \tag{1}$$

where: $x$ is the clean image, $y$ the observed noisy image, and $v$ the noise.

Layer-wise propagation (typical DnCNN):

Figure 3: The DnCNN Network.




Source: Authors.

- First layer (Conv + ReLU, no BN).

$$h_1 = ReLU(\mathrm{W}_1 * h_0 + b_1) \tag{2}$$

where: $h_0 = y$

- Middle layers $l = 2, \ldots, D-1$ (Conv + BN + ReLU).

$$h_l = ReLU(BN(\mathrm{W}_l * h_{l-1} + b_l)) \tag{3}$$

- Final layer (linear, no BN/ReLU) outputs the estimated noise.

$$\hat{v} = \mathrm{W}_D * h_{D-1} + b_D \tag{4}$$

*2.3. YOLOv8 Model Structure.*

Figure 4 depicts the structure of the YOLOv8 model with three parts as follows (Elhanashi et al., 2024):

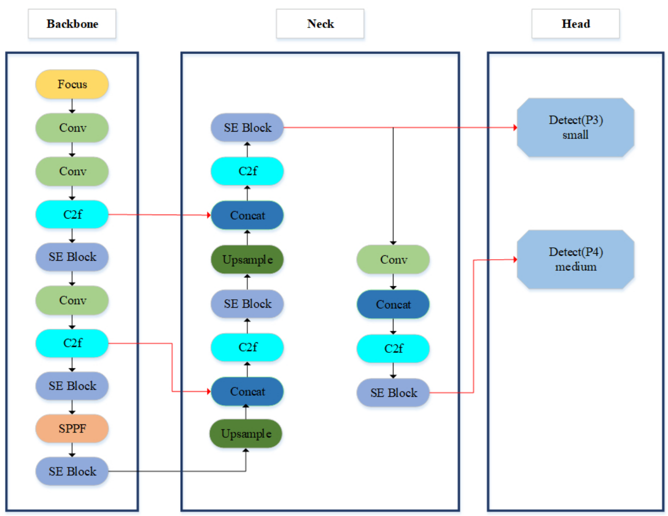
- Backbone: has feature extraction function including Focus, Conv, C2f blocks and SE blocks to create features in different layers.
- Neck: a proportional fusion function with the goal of generating information from space and labels at multiple sizes.
- Head: prediction function with 2 outputs, Detect (P3) for small obstacles and Detect (P4) for medium obstacles. In addition, for large obstacles, there will be Detect (P5) (Yao et al., 2024).

*2.4. Collision Avoidance Situations.*

Figure 5 depicts the situational awareness-based collision avoidance architecture for USVs, in which the inputs are risk indicators, such as $d_{CPA}$, $t_{CPA}$ . (H. Zhu & Ding, 2023)

The sensor system block receives data from the external environment, including environmental conditions and obstacles, as well as feedback from the USV. The encounter-situation module has the function of classifying scenarios according to COLREGs and selecting a reasonable maneuvering scenario. The obstacle avoidance algorithm block calculates the trajectory and controls safety. Finally, the self-adaptive controller provides the

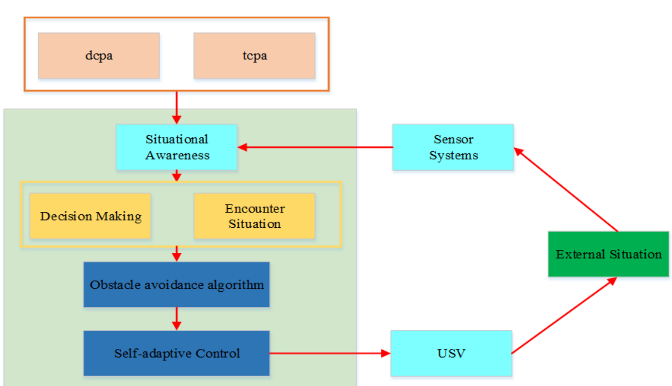
Figure 4: YOLOv8 model structure.




Source: Authors.

ship with a steering angle and speed, and adjusts the heading the direction according to the feedback (Zhuang et al., 2021). The Closest Point of Approach (CPA) is the point at which the distance between the ship and another target object will reach its minimum value. The CPA distance geometry is illustrated in Figure 6.

Figure 5: USV Collision-Avoidance Architecture with Situational Awareness and Self-Adaptive Control.




Source: Authors.

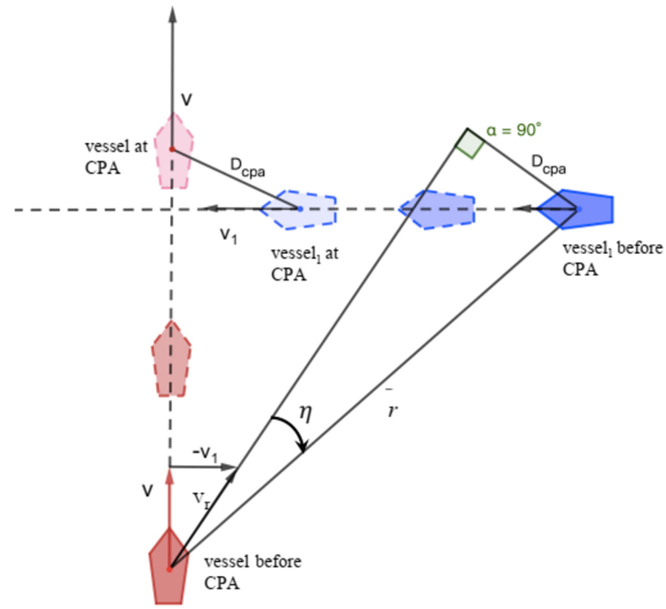
The equation describing the distance between the USV and obstacles can be represented as (Ha et al., 2021):

$$r = \sqrt{(x - x_o)^2 + (y - y_o)^2} \tag{5}$$

$$v_r = v - v_o - \dot{d} \tag{6}$$

where: $r$ is the Euclidean distance between the center points of the two ships.

Figure 6: Distance at the closest point of approach.




Source: H. Zhu & Ding, 2023b.

CPA consists of two parameters: the distance at the closest point of approach ($d_{CPA}$) and the time to the closest point of approach ($t_{CPA}$);$\alpha$ is the angle between the relative bearing of the obstacle ship and heading of the USV (Song et al., 2019; Zhuang et al., 2021):

$$d_{CPA} = r \times \sin(\alpha) \tag{7}$$

$$t_{CPA} = r \times \cos(\alpha)/v_r \tag{8}$$

### 2.5. *PSO collision avoidance.*

The PSO algorithm is applied in waypoint and trajectory optimization as follows:

With the seed $p$at the loop $t$(Xia et al., 2020):

$$\begin{aligned} v_p^{t+1} &= \omega v_p^t + c_1 r_1 (x_{p,best} - x_p^t) \\ &+ c_2 r_2 (x_{gbest} - x_p^t) \end{aligned} \tag{9}$$

$$x_p^{t+1} = x_p^t + v_p^{t+1} \tag{10}$$

where:

$x_p^t$: particle $p$'s **position** (decision vector) at iteration *t;*

$v_p^t$ : particle $p$'s **velocity** (decision vector) at iteration *t;*

$\alpha$: inertia weight: scales the previous velocity; higher values favor exploration, lower values favor exploitation;

$c_1$: cognitive (self) coefficient: how strongly the particle is pulled toward its own best-found position;

$c_2$: social coefficient: how strongly the particle is pulled toward the global (swarm) best position;

$r_1, r_2$: independent uniform random factors that add stochasticity to the cognitive;

$X_{p,\text{best}}$: particle $p$'s personal best position discovered so far;

$X_{g\text{best}}$: swarm's global best position discovered so far.

The USV's objective function with small goals and weights is as follows (Gu et al., 2022):

$$\min_{W} J = \omega_1 J_{length} + \omega_2 J_{risk} + \omega_3 J_{rule} + \omega_4 J_{smooth} \tag{11}$$

where: $J_{length}$ is total distance traveled by USV from starting point to destination (Papadimitrakis et al., 2021);

$J_{risk}$: is the safety function calculated according to the risk index in formulas (7), (8);

$J_{rule}$: COLREGs violation penalty (head-on / starboard cut / overtake), right-of-way priority;

$J_{smooth}$: trajectory and steering angle variation.

Algorithm 1 below shows waypoint optimization with PSO algorithm.

| **Algorithm 1: PSO_Optimize_Waypoints for USV Collision Avoidance** |
|---|
| **Input:** S, G, Obstacles, Targets, K, Bounds B, Params, Weights, Safety |
| **BEGIN** |
| 1: Encode x = [x1,y1,...,xK,yK], x ∈ B<br>2: Initialize swarm {Xp,Vp}p=1..nPop uniformly in B; set pBest=Xp; eval Fp=Cost(Xp,...)<br>3: (gBest,Jbest) ← argminp Fp<br>4: FOR t = 1..Params.itMax DO<br>5: FOR each particle p DO<br>6: r1,r2 ← U(0,1);<br>7: Vp ← ωVp + c1 r1 (pBestp−Xp) + c2 r2 (gBest−Xp)<br>8: Vp ← Clamp(Vp, ±vClamp); Xp ← ProjectToBox(Xp + Vp, B)<br>9: Fp ← Cost(Xp, S,G, Obstacles,Targets, Weights,Safety) // length + CPA/TCPA + COLREGs + smooth<br>10: IF Fp < pBestF[p] THEN pBestp ← Xp; pBestF[p] ← Fp ENDIF<br>11: ENDFOR<br>12: (cand,Jcand) ← argminp Fp; IF Jcand < Jbest THEN gBest←cand; Jbest←Jcand ENDIF<br>13: IF Converged(Jbest) OR FeasibleEnough(gBest,Safety) THEN BREAK ENDIF<br>14: ENDFOR<br>15: W* ← reshape(gBest,K,2); Path* ← [S → W* → G]<br>END |
| **END** |
| **Output:** Optimized waypoints W*, collision-safe path Path*, best cost gBestF |

After calculating the waypoint and optimal path, the author proposes Algorithm 2 to evaluate according to the safety index, regulations and smoothness when steering along the trajectory (Song et al., 2019; H. Zhu & Ding, 2023b; Zhuang et al., 2021).

| **Algorithm 2: EvaluateCost(W, S, G, Obstacles, Targets, Weights, Safety)** |
|---|
| **Input:** W={w1..wK}, S, G, Obstacles(cj,Rj), Targets(pi,vi). |
| **BEGIN** |
| 1: P ← [S, w1..wK, G]; J_len=0; J_static=0; J_dyn=0; J_rule=0; J_smooth=0<br>2: FOR m = 0..K DO<br>3: seg ← (P[m], P[m+1]); L ← length(seg); ψ ← heading(seg); J_len += L<br>4: FOR each (cj,Rj) IN Obstacles DO<br>5: d ← dist_segment_circle(seg, cj); J_static += penalty(d - Rj)<br>6: ENDFOR<br>7: Δt ← L / v_nominal; v_usv ← v_nominal * unit(seg)<br>8: FOR each target (pi,vi) IN Targets DO<br>9: (tcpa, dcpa) ← CPA(P[m], v_usv, pi, vi, Δt)<br>10: J_dyn += penalty(dcpa - R_safe) + η * penalty(T_look - tcpa)<br>11: θ ← wrap(angle(pi - P[m]) - ψ)<br>12: IF 0 < θ < (π/2 + δ) THEN J_rule += soft_penalty(turning_left_amount()) ENDIF<br>13: ENDFOR<br>14: IF m>0 THEN Δψ ← wrap(ψ - ψ_prev); J_smooth += Δψ∧2 + penalty(\|Δψ\| - Δψ_max) ENDIF<br>15: ψ_prev ← ψ<br>16: ENDFOR<br>17: J_safe ← J_static + J_dyn<br>18: J ← λ_len*J_len + λ_safe*J_safe + λ_rule*J_rule + λ_smooth*J_smooth<br>19: RETURN J |
| **END** |
| **Output:** Optimized J |

## 3. Experiments and Implementation.

### 3.1. Simulation HIL setup.

The authors conducted multiple scenarios using the HIL setup, as shown in Figure 7. The Unity 3D simulation environment was built with multiple objects, including ships, waves, wind, and different types of obstacles in the port. The weather scenarios were customized to meet the testing requirements (Hashali et al., 2024).

In this study, the authors used YOLOv8, a deep learning technology implemented on an Intel (R) Core (TM) i7-10700 CPU @ 2.90 GHz, 16 cores, 40 logical processors, and NVIDIA's GeForce RTX 3060Ti Ram 128Gb.

### 3.2. Scenarios.

To test the collision avoidance and waypoint calculation capabilities of the algorithm, the authors built four test scenarios as follows:

Scenario 1 (S1): Natural collision avoidance (Baseline tracking) with the objective of the controller moving from the start point to the destination point and avoiding collisions reflexively with LO and PID without optimizing the waypoints. The aim of the scenario is to compare the minimum milestones in terms

Figure 7: HIL Environment for USV Navigation and COLREGs Evaluation.


Source: Authors.

of safety and travel time. Table 1 shows the input parameter settings (Guan & Wang, 2023; Liang et al., 2025).

Table 1: S1 – No PSO (Baseline tracking).

| Parameter | Typical Value | Notes |
|---|---|---|
| Planner | LOS + reactive collision avoidance | Straight line S→G, no waypoint optimization |
| Safety | $d_{safe} = 30m; t_{safe} = 30s$ | Thresholds for near-miss and evaluation |
| LOS tracking | $L = 8 - 12m$ | Scaled to low harbor speeds |
| Harbor limits | $v_{max} = 4 - 6kn$ | delta |
| PID controllers | Yaw $K_p = 1.2; K_i = 0.05; K_d = 0.2$ Surge $K_p = 0.6; K_i = 0.02; K_d = 0.05$ | Keep fixed across scenarios for fair comparison |
| Repeats & environment | (N=20) per scenario; Day/Night/Fog | Different random seeds |

Source: Authors.

Scenario 2 (S2): Waypoint optimization is performed using the PSO algorithm. The experimental method is based on an optimal waypoint for K waypoints, considering cost and length functions, safety, and smoothness, while disregarding the COLREGs rule. The experimental results show that it is more effective than S1. Table 2 shows the input setting values for the scenario (Guan & Wang, 2023; Liang et al., 2025; Ren et al., 2021).

Table 2: S2 — PSO (Waypoint optimization).

| Parameter | Typical Value | Notes |
|---|---|---|
| Planner | PSO-optimized waypoints + LOS tracking | Cost = path length + safety + smoothness (no COLREGs penalty) |
| Safety | $d_{safe} = 30m; t_{safe} = 30s$ | Thresholds for near-miss and evaluation |
| PSO settings | $K = 5 - 9$ waypoint; $n_{pop} = 24$ $iters = 80; \omega = 0.72; c_1 = c_2 = 1.6$ v-clamp =0.1 – 0.2 span | Bounds from workspace B |
| LOS tracking | $L = 8 - 12m$ | PID yaw/surge same as S1 for fairness |
| Harbor limits | $v_{max} = 4 - 6kn$ | delta |
| Repeats & environment | (N=20) per scenario; Day/Night/Fog | Different random seeds |

Source: Authors.

Scenario 3 (S3): The authors implemented another algorithm, ACO, to compare with the selected algorithm. This testing method helps identify the optimal path on the grid, considering distances and gaps. The results found the optimal paths on the grid and compared the time and performance between the two algorithms. Table 3 presents the algorithm's parameters (Guan & Wang, 2023; Liang et al., 2025).

Table 3: S3 — ACO (Ant Colony Optimization, grid-based).

| Parameter | Typical Value | Notes |
|---|---|---|
| Planner | ACO path on 2D grid + LOS tracking | ACO path on 2D grid + LOS tracking |
| Safety | $d_{safe} = 30m; t_{safe} = 30s$ | Used for near-miss/evaluation; inflate obstacles by |
| ACO settings | $ants = 40, iters = 100; \alpha = 1.0$ $\beta = 2.0, \rho = 0.4, Q = 1.0$ | $\alpha$: pheromone weight; $\beta$: heuristic weight; $\rho$: evaporation |
| Grid / heuristic | cell = 5–10 m; 8-connected moves; $\eta = 1/(L + k\max(0, d_{safe} - d)^2)$ $, k = 0.5$ | PID yaw/surge same as S1 for fairness |
| Harbor limits | $v_{max} = 4 - 6kn$ | delta |
| Repeats & environment | (N=20) per scenario; Day/Night/Fog | Different random seeds |

Source: Authors.

Scenario 4 (S4): applies the PSO algorithm and COLREGs to reduce violations and approach the optimal solution. The experimental method utilizes the PSO algorithm and incorporates COLREGs, CPA, and TCPA rules into the cost function. The result increases the possibility of completing the route a shorter time. Table 4 shows the setting parameter settings for Scenario 4 (Guan & Wang, 2023; Liang et al., 2025; Ren et al., 2021).

Table 4: S4 — PSO + COLREGs (COLREGs-aware waypoint optimization).

| Parameter | Typical Value | Notes |
|---|---|---|
| Planner | PSO-optimized waypoints + COLREGs-aware cost + LOS tracking | Waypoints optimized, controller follows via LOS |
| Safety | $d_{safe} = 30m; t_{safe} = 30s$ | Near-miss/evaluation thresholds (CPA/TCPA) |
| PSO & cost weights | $K = 5 - 9$ waypoint; $n_{pop} = 24$ $iters = 80; \omega = 0.72;$ $c_1 = c_2 = 1.6$ v-clamp =0.1 – 0.2 span $\lambda_{len} = 10; \lambda_{safe} = 4; \lambda_{smooth} = 0.8;$ $\lambda_{rule} = 8; \lambda_{CPA} = 3$ | Bounds from workspace B |
| COLREGs logic | Head-on (R14): alter to starboard; Crossing (R15): give-way passes astern; Overtaking (R13): overtaker keeps clear | Count violation if rule not satisfied within $t_{safe}$ or $d_{CPA} < d_{safe}$ |
| LOS tracking & PID | $L = 8 - 12m$ Yaw $K_p = 1.2; K_i = 0.05; K_d = 0.2$ Surge $K_p = 0.6; K_i = 0.02; K_d = 0.05$ | PID yaw/surge same as S1 for fairness |
| Limits & runs | $v_{max} = 4 - 6kn$ | delta |
| Repeats & environment | (N=20) per scenario; Day/Night/Fog | Different random seeds |

Source: Authors.

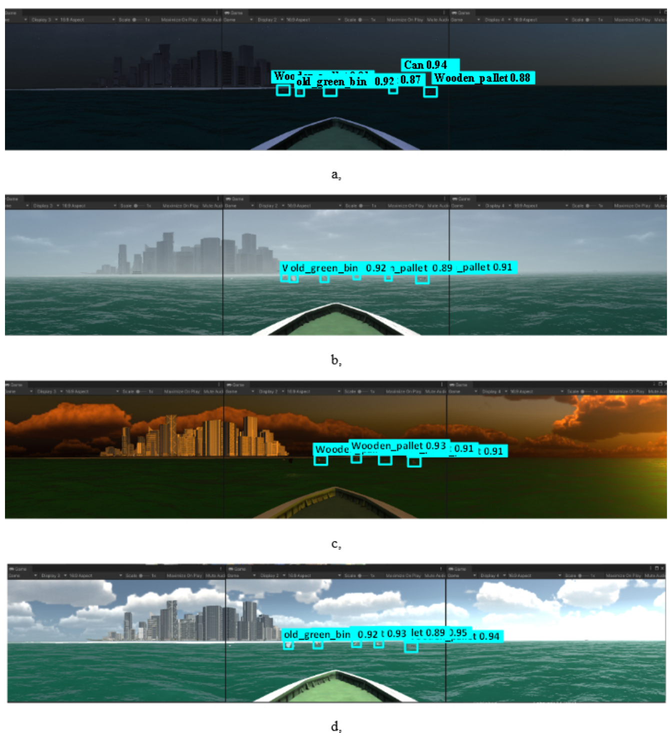
### *3.3. Result.*

#### *3.3.1. Classification test results.*

Figure 8.a shows the YOLOv8 scenario for detection in night-time weather conditions. The results show that the model can detect small, distant obstacles. However, it still misses

some obstacles with confidence scores in the 0.67–0.75 range. In this environment, the scene introduces noise and low-light regions around objects (Lee et al., 2018; Nguyen et al., 2021; Yu et al., 2024).

Figure 8: a, Night/Low-Light Scene; b, Fog/Haze Conditions; c, Sunset/Backlit Scene; d, Clear Daylight Scene.




Source: Authors.

Figure 8.b shows that the object detection ability is better at 0.90. However, the environment with low contrast makes it more challenging to detect the objects. Figure 8.c has a high confidence score of approximately 0.94, and the object detection is relatively uniform. Figure 8.d shows high stability, and this is the clearest test environment. The model detects objects across many positions. The image results show that the ability is clearly different across different test environments. Table 4 shows the results of the evaluation indexes with the YOLOv8 model.

Table 5: YOLOv8 Detection Results.

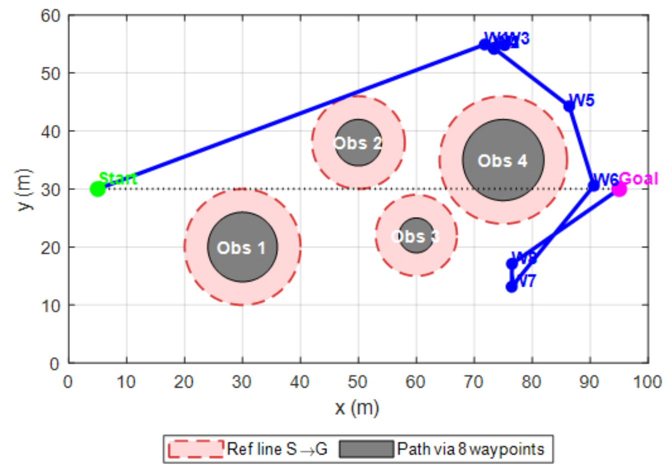
| | Condition | mAP@50 | mAP@50:95 | FPS |
|---|---|---|---|---|
| 1 | Daylight | 0.92 | 0.72 | 40 |
| 2 | Sunset/Backlit | 0.9 | 0.69 | 39 |
| 3 | Fog/Haze | 0.86 | 0.64 | 36 |
| 4 | Night/Low-light | 0.84 | 0.61 | 35 |

Source: Authors.

### 3.3.2. *Collision avoidance results.*

The results of Scenario 1 with the random scenario are shown in Figure 8 (Hu et al., 2022).

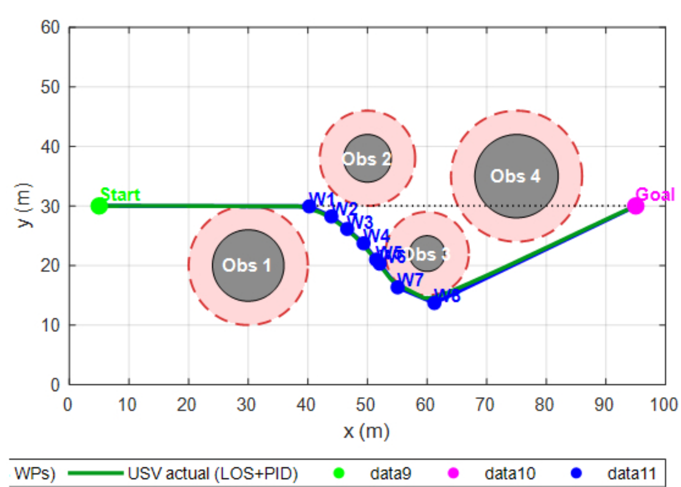
Figure 9: Scenario 1 Random Waypoint Avoidance (4 Obstacles)




Source: Authors.

Figure 9 shows the collision avoidance results of the USV going from the starting point to the destination with 4 obstacles with the red safety zone. With this scenario, the USV meets the requirement of avoiding all obstacles through waypoints W3, W5, and W6 without crossing the safety zone. However, with random collision avoidance, the smoothness through waypoints is low and difficult to follow the route (Guan & Wang, 2023; Liang et al., 2025; Ren et al., 2021).

Figure 10: Scenario 2: PSO Waypoints + Actual USV Path (LOS/PID)

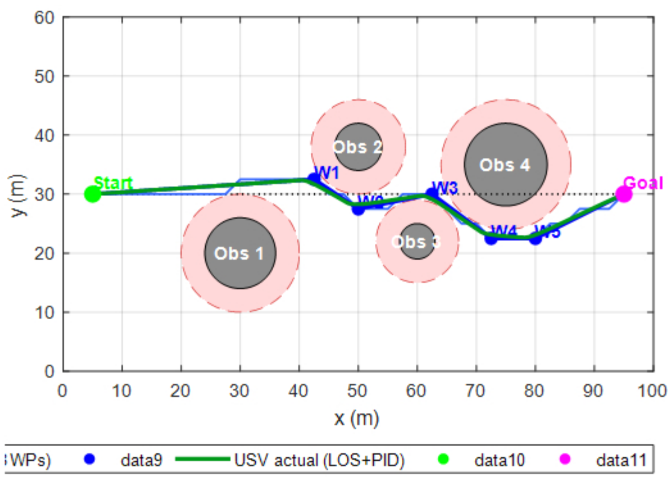



Source: Authors.

Figure 10 illustrates the collision avoidance results of the USV navigating from the starting point to the destination, with four obstacles, utilizing a combination of PSO and the actual USV controllers (LOS and PID) collision avoidance algorithms.

The trajectory results show that the actual USV and the proposed algorithm's paths remain closely aligned. Waypoint W3 - W8 show that the USV tracks well and exhibits less oscillation than Scenario 1 (Guan & Wang, 2023; Liang et al., 2025; Ren et al., 2021).

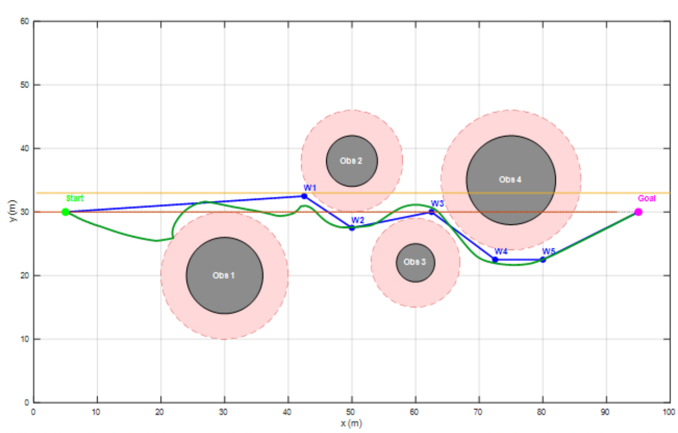
Figure 11: Scenario 3 Grid Path + Simplified Waypoints + USV Actual.




Source: Authors.

Figure 11 illustrates the collision avoidance results obtained using the ACO algorithm, demonstrating a relatively strong obstacle - avoidance capability. The actual trajectory of the USV does not intersect the red safe zone between W2 and W3. However, the passage between W3 and W4 is lengthy, which increases the total route length (Guan & Wang, 2023; Liang et al., 2025; Ren et al., 2021; Wang et al., 2024).

Figure 12: Scenario 4: Dynamic COLREGs Avoidance + PSO and LOS-PID.




Source: Authors.

Figure 12 shows the collision avoidance results with PSO and COLREGs algorithms, showing better collision avoidance performance. The generated waypoint trajectories and the actual trajectory of the USV avoided the objects while complying with the COLREGs rules (Guan & Wang, 2023; Liang et al., 2025; Ren et al., 2021; Wang et al., 2024).

## 4. Conclusion.

Collision avoidance and maintaining stable operations in ports for USVs are important tasks that require high stability and predictability. This paper presents a method of applying YOLOv8 to detect objects in the port area under different weather conditions. At the same time, the PSO algorithm in avoidance and COLREGs rule combination to find the waypoints to create the most optimal path. The experimental research results in the simulation environment show that the ability to avoid collisions and detect objects is relatively stable and has high efficiency. This research result initially shows that the ability to combine multiple USVs in ports and coordinate operations will be effective in the future.

## Acknowledgment.

This research is funded by Vietnam Maritime University

## References.

Boadu, S., Anu Thompson, E., Godavarthy, R., Boateng, A., & Penneigh, R. (2025). A decade of Smart Port Concept: a comprehensive review of past, present, and future implications. *Australian Journal of Maritime and Ocean Affairs*, 1–27. https://doi.org/10.1080/18366503.2025.2546729.

Elhanashi, A., Dini, P., Saponara, S., & Zheng, Q. (2024). TeleStroke: real-time stroke detection with federated learning and YOLOv8 on edge devices. *Journal of Real-Time Image Processing*, 21(4), 1–16. https://doi.org/10.1007/s11554-024-01500-1.

Fan, Y., Sun, Z., & Wang, G. (2023). A novel intelligent collision avoidance algorithm based on deep reinforcement learning approach for USV. *Ocean Engineering*, 287, 115649. https://doi.org/10.1016/j.oceaneng.2023.115649.

Gu, Y., Rong, Z., Tong, H., Wang, J., Si, Y., & Yang, S. (2022). Unmanned surface vehicle collision avoidance path planning in restricted waters using multi-objective optimisation complying with COLREGs. *Sensors*, 22(15). https://doi.org/10.3390/s22155796.

Guan, W., & Wang, K. (2023). Autonomous collision avoidance of unmanned surface vehicles based on improved A-Star and dynamic window approach algorithms. *IEEE Intelligent Transportation Systems Magazine*, 15(3), 36–50. https://doi.org/10.1109/MITS.2022.3229109.

Ha, J., Roh, M.-I., & Lee, H.-W. (2021). Quantitative calculation method of the collision risk for collision avoidance in ship navigation using the CPA and ship domain. *Journal of Computational Design and Engineering*, 8(3), 894–909. https://doi.org/10.1093/jcde/qwab021.

Hashali, S. D., Yang, S., & Xiang, X. (2024). Route planning algorithms for Unmanned Surface Vehicles (USVs): A comprehensive analysis. *Journal of Marine Science and Engineering*, 12(3). https://doi.org/10.3390/jmse12030382.

Hu, L., Hu, H., Naeem, W., & Wang, Z. (2022). A review on COLREGs-compliant navigation of autonomous surface vehicles: From traditional to learning-based approaches. *Journal of Automation and Intelligence*, 1(1), 100003. https://doi.org/10.1016/j.jai.2022.100003.

Lee, S.-J., Roh, M.-I., Lee, H., Ha, J.-S., & Woo, I.-G. (2018). *Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks*.

Liang, Z., Yang, Y., Huang, Z., & Zhou, S. (2025). USV path planning algorithm based on 2D-A* and IDWA algorithms under COLREGs. *Ocean Engineering*, 331, 121389. https://doi.org/10.1016/j.oceaneng.2025.121389.

Lyu, H., Hao, Z., Li, J., Li, G., Sun, X., Zhang, G., Yin, Y., Zhao, Y., & Zhang, L. (2023). Ship autonomous collision-avoidance strategies—A comprehensive review. *Journal of Marine Science and Engineering*, 11(4). https://doi.org/10.3390/jmse11040830.

Nguyen, T.-T., Tran, N.-H., Ho, T.-M.-D., & Nguyen, H. (2021). Path planning for Unmanned Surface Vehicle (USV) in obstacle-filled environments. *2021 International Conference on Advanced Technologies for Communications (ATC)*, 104–108. https://doi.org/10.1109/ATC52653.2021.9598281.

Niu, L., Fan, Y., Liu, T., & Han, Q. (2025). Developing lightweight object detection models for USV with enhanced maritime surface visible imaging. *Journal of Visual Communication and Image Representation*, 110, 104477. https://doi.org/10.1016/j.jvcir.2025.104477.

Papadimitrakis, M., Stogiannos, M., Sarimveis, H., & Alexandridis, A. (2021). Multi-ship control and collision avoidance using MPC and RBF-based trajectory predictions. *Sensors*, 21(21). https://doi.org/10.3390/s21216959.

Ren, J., Zhang, J., & Cui, Y. (2021). Autonomous obstacle avoidance algorithm for unmanned surface vehicles based on an improved velocity obstacle method. *ISPRS International Journal of Geo-Information*, 10(9). https://doi.org/10.3390/ijgi10090618.

Santos, T. A. (2025). Sustainable port operations: Pollution prevention and mitigation strategies. *Sustainability (Switzerland)*, 17(11). https://doi.org/10.3390/su17114798.

Shaobo, W., Yingjun, Z., & Lianbo, L. (2020). A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method. *Ocean Engineering*, 215, 107910. https://doi.org/10.1016/j.oceaneng.2020.107910.

Shen, J., Ren, X., Feng, Z., & Nie, J. (2024). The impact of public environmental concerns on port sustainability: Evidence from 44 port cities in China. *Frontiers in Marine Science*, 11(January), 1–16. https://doi.org/10.3389/fmars.2024.1454242.

Song, L., Chen, Z., Dong, Z., Xiang, Z., Mao, Y., Su, Y., & Hu, K. (2019). Collision avoidance planning for unmanned surface vehicle based on eccentric expansion. *International Journal of Advanced Robotic Systems*, 16(3), 1–9. https://doi.org/10.1177/1729881419851945.

Song, L., Shi, X., Sun, H., Xu, K., & Huang, L. (2021). Collision avoidance algorithm for USV based on rolling obstacle classification and fuzzy rules. *Journal of Marine Science and Engineering*, 9(12). https://doi.org/10.3390/jmse9121321.

Wang, D., Chen, H., & Wu, C. (2024). A path planning and obstacle avoidance method for USV based on dynamic-target APF algorithm in edge. In Z. Tari, K. Li, & H. Wu (Eds.), *Algorithms and Architectures for Parallel Processing* (pp. 21–39). Springer Nature Singapore.

Xia, G., Han, Z., Zhao, B., & Wang, X. (2020). Local path planning for unmanned surface vehicle collision avoidance based on modified quantum particle swarm optimization. *Complexity*, 2020(1), 3095426. https://doi.org/10.1155/2020/3095426.

Yang, X., Lou, M., Hu, J., Ye, H., Zhu, Z., Shen, H., Xiang, Z., & Zhang, B. (2024). A human-like collision avoidance method for USVs based on deep reinforcement learning and velocity obstacle. *Expert Systems with Applications*, 254, 124388. https://doi.org/10.1016/j.eswa.2024.124388.

Yao, G., Zhu, S., Zhang, L., & Qi, M. (2024). HP-YOLOv8: High-precision small object detection algorithm for remote sensing images. *Sensors*, 24(15). https://doi.org/10.3390/s24154858.

Yu, C., Yin, H., Rong, C., Zhao, J., Liang, X., Li, R., & Mo, X. (2024). YOLO-MRS: An efficient deep learning-based maritime object detection method for unmanned surface vehicles. *Applied Ocean Research*, 153, 104240. https://doi.org/10.1016/j.apor.2024.104240.

Zhu, H., & Ding, Y. (2023a). Optimized dynamic collision avoidance algorithm for USV path planning. *Sensors*, 23(9), 1–18. https://doi.org/10.3390/s23094567.

Zhu, H., & Ding, Y. (2023b). Optimized dynamic collision avoidance algorithm for USV path planning. *Sensors*, 23(9). https://doi.org/10.3390/s23094567.

Zhu, Q., Xi, Y., Weng, J., Han, B., Hu, S., & Ge, Y.-E. (2024). Intelligent ship collision avoidance in maritime field: A bibliometric and systematic review. *Expert Systems with Applications*, 252, 124148. https://doi.org/10.1016/j.eswa.2024.124148.

Zhuang, J., Zhang, Y., Xu, P., Zhao, Y., Luo, J., & Song, S. (2021). Multiple moving obstacles avoidance for USV using velocity obstacle method. *Proceedings of 2021 IEEE International Conference on Unmanned Systems, ICUS 2021*, 140–145. https://doi.org/10.1109/ICUS52573.2021.9641331.