Journal of
Marine Science and Engineering

MDPI

Article

ULOTrack: Underwater Long-Term Object Tracker for Marine Organism Capture

Ju He [1,*,†], Yang Yu [1,2,*,†], Hongyu Wei [1] and Hu Xu [1]

[1] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China; hongyuwei@mail.nwpu.edu.cn (H.W.); xuhu@mail.nwpu.edu.cn (H.X.)
[2] Shenzhen Research & Development Institute, Northwestern Polytechnical University, Shenzhen 510085, China
* Correspondence: hj@mail.nwpu.edu.cn (J.H.); nwpuyuy@nwpu.edu.cn (Y.Y.)
† These authors contributed equally to this work.

Abstract: Underwater object tracking holds considerable significance in the field of ocean engineering. Additionally, it serves as a crucial component in the operations of autonomous underwater vehicles (AUVs), particularly during tasks associated with capturing marine organisms. However, the attenuation and scattering of light result in shortcomings such as poor contrast in underwater images. Additionally, the motion deformation of marine organisms poses a significant challenge. Therefore, existing tracking algorithms face difficulty in direct application to underwater object tracking. To overcome this challenge, we propose a novel tracking architecture for the marine organism capturing of AUVs called ULOTrack. ULOTrack is based on a performance discrimination and re-detection framework and constitutes three modules: (1) an object tracker, which can extract multi-feature information of the underwater target; (2) a multi-layer tracking performance discriminator, which serves the purpose of evaluating the stability of the current tracking state, thereby reducing potential model drift; and (3) lightweight detection, which can predict the candidate boxes to relocate the lost tracked underwater object. We conduct comprehensive experiments to validate the efficacy of the designed modules. Finally, the results of the experimentation demonstrate that ULOTrack significantly outperforms existing approaches. In the future, we aim to carefully scrutinize and select more suitable features to enhance tracking accuracy and speed.

Keywords: visual perception; underwater observation; underwater object tracking; autonomous underwater vehicle; tracking discriminator

Citation: He, J.; Yu, Y.; Wei, H.; Xu, H. ULOTrack: Underwater Long-Term Object Tracker for Marine Organism Capture. *J. Mar. Sci. Eng.* **2024**, *12*, 2092. https://doi.org/10.3390/jmse12112092

Academic Editor: Rafael Morales

Received: 20 October 2024
Revised: 11 November 2024
Accepted: 17 November 2024
Published: 19 November 2024

1. Introduction

Autonomous underwater vehicles (AUVs) play a crucial role in various marine applications [1] such as ocean environment monitoring [2,3], object tracking [4,5], underwater terrain mapping [6], and marine search and rescue [7]. Using an AUV enables the safe, stable, and efficient tracking and capturing of marine organisms, thereby reducing the dependency on human resources. Object tracking can be used to determine the specific position and motion trajectory of an object; therefore, the accuracy of the tracker is crucial in executing subsequent capture actions in complex underwater environments.

Underwater object tracking relies on various sensors that address the unique challenges posed by the underwater environment, including limited visibility, light attenuation, and turbidity. Sonar, particularly side-scan and multibeam sonar, provides effective tracking over greater distances and can penetrate turbid water, making it ideal for detecting larger underwater structures or organisms. Acoustic sensors are also popular, especially in low-visibility conditions, as they can detect and localize objects by measuring sound reflections, which are less affected by water turbidity. However, the low resolution of sonar limits the accuracy of detecting and tracking specific targets. Optical cameras, with their high-resolution imaging capabilities, offer detailed visual tracking but are limited by light

attenuation and perform best in clear, shallow waters. Unlike water surface visual perception for unmanned surface vehicles [8–10], long-term underwater object tracking often encounters various severe challenges. More specifically, regarding the inherent limitations of underwater optical imaging, the attenuation and scattering of light underwater result in reduced clarity and contrast in images. Different wavelengths of light are absorbed to various extents in water, which causes color distortion in underwater images. Uneven illumination and lighting conditions, as well as suspended particles and water movement, restrict the depth of light penetration in water. This limitation constrains the visibility of images and often leads to a loss of details.

Furthermore, the movement properties of marine organisms pose a series of challenges in object tracking. On the one hand, the non-rigid motions, deformations, and out-of-plane rotations of marine organisms, along with the changes in the perspective of AUVs, challenge the accuracy and stability of object tracking. On the other hand, the motion blur of targets affects their clarity and outlines in images. Given the complexity of the marine environment, addressing these challenges requires more precise tracking methods.

The mainstream object tracking algorithms can be divided into two categories: traditional methods and deep learning methods. Ref. [11] proposed underwater object tracking based on underwater image enhancement. Ref. [12] proposed an improved KCF tracker and a novel fuzzy controller. Ref. [13] proposed a lightweight Siamese network that enhances the capability of feature extraction. Ref. [14] proposed a fish tracking method based on adaptive multi-appearance models. Ref. [15] designed a novel fish tracking algorithm based on deformable multiple kernels. Ref. [16] proposed a multiscale underwater tracker using the adaptive feature. Ref. [17] incorporated an implicit motion modeling module into a tracker, enhancing the ability to distinguish the tracked target from similar interferences.

The aforementioned trackers lack the capability of re-detection after target loss. Recently, trackers incorporating re-detection mechanisms have emerged. Ref. [18] introduced a high-precision target tracking method that includes anomaly tracking status detection and recovery. Ref. [19] proposed a joint local–global search mechanism, while Ref. [20] focused on a coarse-to-fine re-detection and spatial–temporal reliability evaluation. Ref. [21] explored the exploitation of both local and global properties. In scenarios where marine objects exhibit rapid movements, encounter obstruction by aquatic vegetation, or undergo significant scale variations, inappropriate sampling and model updates lead to template drift issues [22,23], as illustrated in Figure 1. This template drift ultimately results in tracking failures. The attenuation and scattering of light result in shortcomings such as poor contrast in underwater images. Additionally, the motion deformation of marine organisms poses a significant challenge. However, the above mentioned methods rarely focus on addressing the deformation of marine organisms or the degradation of underwater images. Therefore, existing tracking algorithms face difficulties in direct application to underwater object tracking.

(a) (b)

Figure 1. (**a**,**b**) Object tracking failure due to water weed occlusion. The yellow boxes denote the fish tracking results.

We observed that, when a target is lost, people change their field of view to broaden the tracking scope. Additionally, they quickly scan or focus on different areas to re-detect visual cues of the target. Once the target is rediscovered, eye movements and gaze changes are minimized, maintaining visual focus on the target to ensure continuous stable tracking. So,

inspired by the human eye tracking system, our approach focuses on accurately discerning object tracking performance to enable adaptive template and search area updates. Furthermore, the re-detection mechanism excels in resisting interference in complex underwater scenarios, achieving the stable detection and tracking of marine organisms. Our lightweight long-term tracking algorithm is designed to run in real time on AUV low-power computing devices, such as a mobile phone with diverse sensors. The mobile phone also integrates key sensors, including GPS and MEMS, for the autonomous navigation of the AUV. This allows for the continuous, long-term tracking of underwater organisms while maintaining minimal energy consumption. Optimizing the algorithm for efficiency ensures that the system can operate for extended periods without overburdening the limited resources of the device. This capability is crucial for autonomous underwater vehicles (AUVs) that rely on low-power sensors and computational hardware to monitor and track marine life over long durations, enabling consistent and accurate tracking in real-world, energy-constrained environments.

In summary, the main contributions of this work can be summarized as follows:

(1) We propose a novel underwater long-term object tracking architecture named ULO-Track, enabling consistent and accurate tracking on a low-power computing AUV platform.
(2) We propose a multi-layer object tracking performance discriminator that evaluates the current tracking state's stability and suppresses the model drift caused by rapid target movement. The layers are structured as follows: the top layer measures the maximum response score, the second layer evaluates the average peak correlation energy, and the third layer counts the number of multiple peaks.
(3) We design a multiscale space filter and calculate scale responses to address the significant scale variations encountered with marine organisms. Extensive experiments on real-world datasets demonstrate that our algorithm not only achieves greater robustness across various target types but also outperforms other algorithms in tracking performance.

2. Related Work

2.1. Object Tracking

Existing object tracking algorithms are mainly divided into three categories: generative object tracking, discriminative object tracking, and deep learning-based object tracking. Generative algorithms rely on the construction of the target feature subspace, while discriminative algorithms are built upon classification or regression methods to discriminate between the target and background. Classic generative tracking algorithms include Kalman filtering [24], particle filtering [25], and mean shift [26]. Bolme first introduced the correlation filter into object tracking and proposed the Minimum Output Sum of Squared Error (MOSSE) [27] algorithm, achieving a processing speed of up to 669 Frames Per Second (FPS). Henriques [28] proposed the Circulant Structure of Tracking-by-Detection with Kernels (CSK) method. Martin Danelljan introduced the Discriminatiive Scale Space Tracker (DSST) [29] algorithm, pioneering the combination of translation and scale filtering algorithms. Henriques [30] proposed the Kernel Correlation Filter (KCF) algorithm, and, based on the KCF framework, Martin Danelljan proposed the learning spatially regularized correlation filters for visual tracking (SRDCF) [31] algorithm. The algorithm improved the robustness in fast-changing scenes but could not meet real-time requirements. In 2017, the CSR_DCF (Channel Spatial Reliability for DCF) algorithm [32] was proposed, utilizing color histograms of the foreground and background, as well as response map information from different channels to enhance spatial and channel reliability. Background-Aware Correlation Filters (BACFs) [33] use real background information displacement to obtain negative samples, thereby expanding the target search area. Spatial–Temporal Regularized Correlation Filters (STRCFs) [34] add temporal and spatial regularization terms to the DCF-based framework. Wen et al. [35] proposed the enhanced robust spatial feature selection tracker. The learning of adaptive sparse spatially regularized correlation filters

(AS2RCF) [36] was used to design an adaptive sparse spatially regularized correlation filter. SOCF [37] is a real-time tracker with spatial disturbance suppression.

Though the above algorithms significantly enhance model discrimination, they involve extensive training and parameter tuning, which makes it challenging to conveniently apply them to underwater mobile platforms. In contrast, ULOTrcker's low computational requirements allow it to be conveniently and reliably deployed on AUVs.

2.2. Underwater Object Tracking for AUVs

Certain underwater tracking systems are required to operate on AUVs for object perception tasks. However, on the one hand, they face constraints due to limited memory space and computing resources. On the other hand, underwater image degradation and low quality directly impact tracking accuracy. To address the aforementioned issues, existing algorithms have made two major improvements to traditional correlation filter trackers, focusing on feature extraction and template update mechanisms: (1) Regarding enhanced feature extraction, the focus has been on refining the process of feature extraction from the target model. This improvement aids in better target recognition and tracking. (2) Regarding template update mechanisms, such methods employ template update strategies to improve the adaptability of the tracking algorithm, allowing it to adjust to changes in the target's appearance.

Several tracking systems have benefited from these enhancements. For instance, Li et al. proposed a real-time fish tracking method based on novel adaptive multi-appearance models and tracking strategies. To address the issue of target occlusion, Ref. [12] designed an improved KCF tracker, which incorporates a self-discrimination mechanism based on the uncertainty of system confidence. Ref. [38] proposed an improved anti-occlusion object tracking algorithm using an unscented Rauch–Tung–Striebel smoother and kernel correlation filter. Ref. [39] added a fusion correction mechanism (FCM) to the KCF tracking algorithm to improve tracking performance. Ref. [40] explored real-time object tracking methods applied to underwater robotics platforms. Additionally, they evaluated color restoration algorithms suitable for enhancing the quality of images. Ref. [41] presented a multiple-fish tracking system for low-contrast and low-frame-rate stereo videos. Ref. [15] proposed a novel tracking algorithm based on the deformable multiple kernels. Furthermore, there are some TLD-based trackers and particle-filter-based trackers. Rout et al. designed Walsh–Hadamard-kernel-based features in a particle filter framework for underwater object tracking [42]. Ref. [43] proposed multi-feature fusion in a particle filter framework for visual tracking. Ref. [44] designed UOSTrack for marine organism grasping. Ref. [13] designed a hybrid excitation model-based lightweight Siamese network. However, most of the aforementioned methods overlook the spatial restriction issue in correlation filters and lack the re-detection capability to refine unreliable tracking results. They also lack a discriminative mechanism and adaptive template updating; template updating is indispensable when the underwater target is undergoing rapid movement and deformation.

3. Experimental Autonomous Underwater Vehicle Platform

In this section, we present our design for a low-cost micro-AUV system. This includes an overview of the mechanical, electronic, and control architecture. The proposed AUV is characterized by its compact size, simplicity, and affordability while offering a range of essential functionalities such as autonomous navigation, object detection, and object tracking. As a result, it can be easily integrated into various projects for intelligent systems and applications. Similar to a conventional AUV, our micro-AUV comprises the following key modules: a water-resistant shell, a navigation and control module, a communication module, an energy management module, a propulsion module, and an external payload interface module, which can accommodate various accessories like side-scan sonar (SSS) and the Doppler Velocity Log (DVL). Supported by all the above-mentioned modules, the micro-AUV design structure is presented in Figure 2.

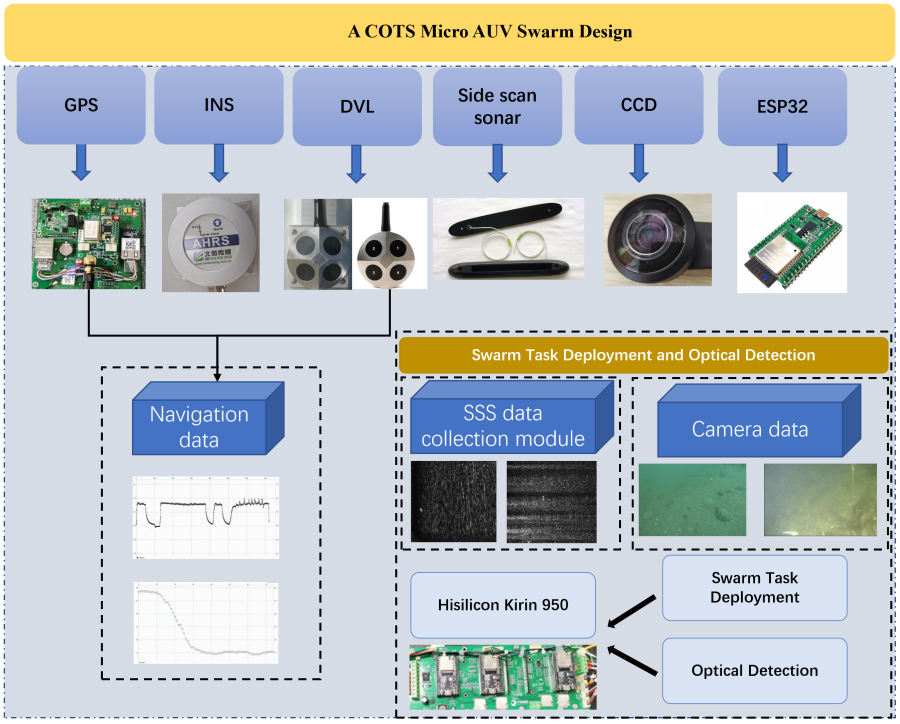

Figure 2. A flowchart of the COTS micro-AUV design.

The navigation and control module: The navigation and control module serves as the central component of the AUV, leveraging the mobile phone's internal components, including the GPU, GPS, and MEMS compass sensor. As previously mentioned, only the pressure sensor unit is not available on the mobile phone. Consequently, we specifically installed a miniature pressure sensor unit with a WiFi interface to help the mobile phone obtain real-time depth data. The module is responsible for the AUV steering, elevation, and rolling control, and a PID control method is used for the process. More interestingly, as a bonus, the mobile phone screen is used as a debug monitor, as well as a flashlight for AUV positioning at night.

The payload interfacing module: A payload interface board is developed, and any necessary payloads can be further installed in the micro-AUV via the WiFi interface. The commonly used payloads like side-scan sonar (SSS), the Doppler Velocity Log (DVL), and Forward-Looking Sonar (FLS), have already been tested in our prototype successfully.

4. Underwater Long-Term Object Tracker

The integrated algorithm for object detection and tracking based on the AUV platform adopts a primarily tracking-oriented approach with detection as a supplementary method. It combines improved correlation filter algorithms and lightweight detection algorithms to create an excellent long-term tracking system, significantly enhancing algorithm performance. A processing flowchart of the algorithm is shown in Figure 3. The long-term tracker consists of three main components: a lightweight object detector, an improved kernel correlation filter algorithm, and a multi-layer tracking confidence discriminator. Initially, a lightweight object detector is used to rapidly and accurately detect the ocean target to be captured. The detected target's position in the first frame is transmitted to the tracker, which then starts tracking. Additionally, a three-level tracking confidence discriminator assesses the target's tracking status. If it is determined that the target is in a non-stationary tracking state, the lightweight object detector is activated, and the repositioning results are sent to the tracker to continue tracking the target. The overall algorithm structure is depicted in Figure 4, and the specific process is detailed below.

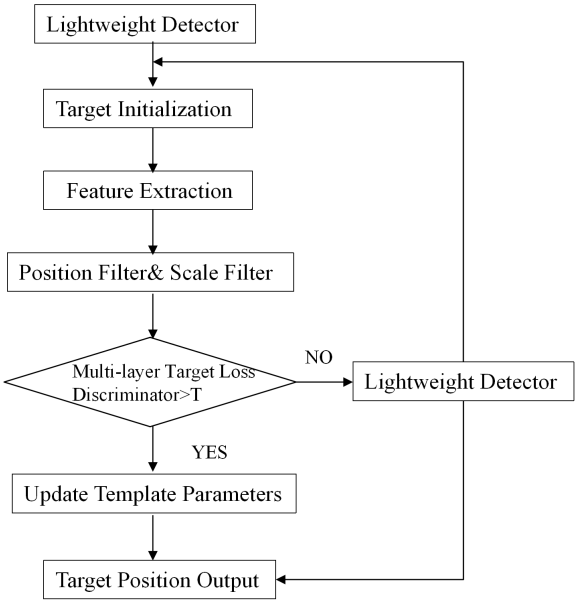

Figure 3. Processing flowchart of the underwater long-term object tracker.

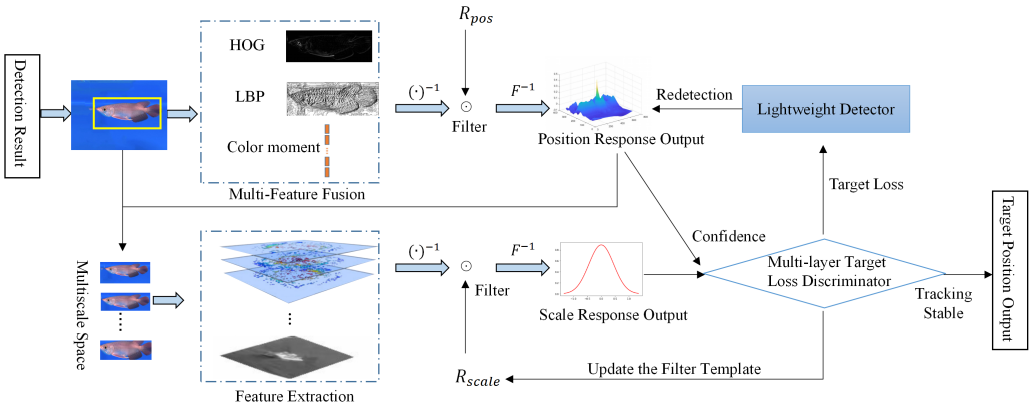

Figure 4. Overall architecture of the underwater long-term object tracker.

Step 1: The underwater camera saves videos as image sequences. A lightweight detection algorithm identifies the target position in the first frame and extracts the search area from the current image.

Step 2: Features from the first-frame target and the search area are extracted, obtaining the region covariance of the Histogram of Oriented Gradient (HOG) [45], Local Binary Patterns (LBPs) [46], and color moment features. These features are used to train the initial position filter R_{pos} and the scale filter R_{scale}.

Step 3: In the subsequent frames, regional cyclic sampling is performed to obtain positive and negative samples, resulting in a position response map and the maximum scale response value.

Step 4: A three-level tracking confidence discrimination mechanism is employed to achieve a combined assessment of the current tracking state. The confidence score for each frame is calculated to determine whether the tracking state is stable.

Step 5: Based on the results of the three-level discrimination, the learning rate is dynamically adjusted, and adaptive template updates are performed. When tracking is unstable, template updates are halted, the lightweight detector is activated, the lost target is re-located, and the positioning results are sent to the tracker.

4.1. Lightweight Object Detection Method for Marine Organisms

A lightweight underwater object detection model will effectively reduce the cost of computational resources for the easier deployment of the AUV. In our work, we design a lightweight underwater object detection model for low-power mobile phone platforms in AUVs and take both the accuracy and speed requirements into account in real-world scenarios.

To improve the quality of underwater images and achieve better detection performance, we first process the input images with image preprocessing. The image preprocessing operations adopt the classical Single-Scale Retinex (SSR) image processing algorithm, which enhances the color and brightness of input images through the stable reflectance of the object in the image, adjusting its pixel values that preserve the overall lightness information while eliminating the effects of non-uniform illumination.

In the lightweight design of detection models, model quantization is an effective way to speed up the interference of the model. However, this often leads to a decline in detection accuracy. To address this issue, we draw inspiration from YOLOv6-3.0, the state-of-the-art detection model in COCO datasets, and replace the conventional CNN block module with a quantization-friendly module called Quantization-Aware RepVGG (QARepVGG). The QARepVGG module ensures model variance stability and delivers outstanding detection accuracy during quantization.

Figure 5 presents the overall architecture of the lightweight underwater object detection model. In the backbone of our model, we employ six QARepVGG blocks and four downsampling layers to extract features from underwater images. These extracted features are then fed into Spatial Pyramid Pooling (SPP) and Feature Pyramid Network (FPN) modules, which enable a larger receptive field and facilitate global context linkage. This approach helps mitigate the impact of water reflections and enhances the model's robustness. Furthermore, the detection results are generated through a detection head based on an anchor-free detection scheme. Unlike the traditional YOLO series object detectors that use an anchor-based detection scheme, which can introduce complexity in embedded computing applications, the anchor-free detection scheme has gained popularity due to its strong generalization capability and faster decoder speed. Therefore, we adopted the anchor-free detection scheme in the detection head to optimize performance.

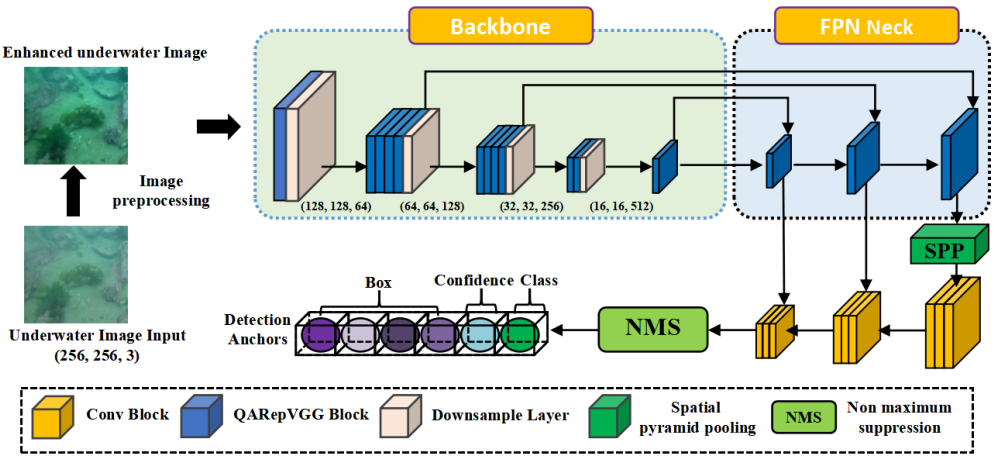

Figure 5. Overall architecture of the lightweight underwater object detection model.

4.2. Underwater Long-Term Object Tracker

The camera captures images and reads them in a sequence. Using the lightweight detection algorithm described in Section 4.1, the automatic acquisition of the target's size and position from the first-frame image is performed. Subsequently, these data are fed into the object tracking stage. The underwater object tracking procedure involves the improved KCF method. The kernel correlation filter possesses rapid and efficient

tracking performance. Firstly, leveraging the circulant matrix properties enables dense sampling, significantly enhancing the number of training samples. Secondly, leveraging Fourier transform and the kernel function notably reduces inference time, optimizing the algorithm's efficiency.

4.2.1. Circulant Matrix

The tracker adopts dense sampling to obtain a large number of positive and negative training samples. It cleverly uses circulant matrix shifting instead of traversing the search area, significantly reducing training time. The initial training samples are obtained by circularly shifting the base sample (the target in the first frame). Taking a one-dimensional matrix as an example, if the base sample is represented as $x = [x_1, x_2, \ldots, x_n]^T$, then the circulant matrix P is defined as follows:

$$P = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{1}$$

Moreover, it is calculated as $Px = [x_n, x_1, x_2, \ldots, x_{n-1}]^T$, which represents the result after one cycle of x; then, after $n-1$ cycles, the result is $\{P^n x | n = 1, 2, \ldots, n-1\}$. This is obtained by combining all cyclic samples to form the circulant matrix X, which constitutes the training sample set for the target.

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \tag{2}$$

The circulant matrix X can be diagonalized using discrete Fourier transformation.

$$X = C(x) = F \cdot diag(\hat{x}) \cdot F^H, \tag{3}$$

where F is the constant matrix of the Fourier transform, F^H is the conjugate transpose matrix of F, and $\hat{x}$ is the discrete Fourier transform of the base vector x, $\hat{x} = \mathcal{F}(x)$.

$$F = \frac{1}{\sqrt{n}} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & \cdots & w^{n-1} \\ 1 & w^2 & w^4 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{n-1} & w^{2(n-1)} & \cdots & w^{(n-1)^2} \end{bmatrix}, \tag{4}$$

where $w = e^{-2\pi i/n}$.

From the above definition, other properties of the circulant matrix can be derived:

$$X^H = F \cdot diag((\hat{x})^*) \cdot F^H. \tag{5}$$

4.2.2. The Training of Classifier

The classifier outputs correlation values for all potential regions and identifies the maximum response as the tracked object. The KCF initially applies ridge regression to train the data. The ridge regression method extends the least squares regression by incorporating a regularization term, effectively handling ill-conditioned data and yielding more stable computational outcomes. In a linear space, assuming a training sample set

of $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, in order to train and obtain a regression model $f(x_i) = w^T x_i$ with the aim of minimizing the error between the predicted and ground truth, the following loss function is defined:

$$\begin{aligned} L(w) &= \min_{w} \sum_{i=1}^{m} (y_i - w^T x_i)^2 + \lambda \|w\|^2 \\ &= min \|Xw - y\|^2 + \lambda \|w\|^2 \end{aligned}, \tag{6}$$

where y_i represents the label values of sample x_i, X denotes the circulant matrix of the input samples, w is the coefficient matrix to be determined, and λ is the regularization term coefficient used to regulate the complexity of the system.

The analysis above is based on a linear space. However, the actual tracking scene is complex. To further address classification issues in nonlinear spaces, the principle of the kernel function in support vector machines is employed. This principle involves mapping the training samples from a lower-dimensional space to a higher-dimensional one, thereby transforming the nonlinear problem into a linearly separable one.

Suppose that the nonlinear mapping function is denoted as $\varphi(x)$ and that the regression model after mapping is $f(x_i) = w^T \varphi(x_i)$. Representing w as a linear combination of the training samples, the question of solving w is transformed into the question of solving α:

$$w = \sum_{i} \alpha_i \varphi(x_i) \tag{7}$$

The kernel function is the kernel correlation matrix of the training sample set, defined as follows:

$$K = \kappa(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \tag{8}$$

The mapping function of the nonlinear space is converted into an inner product calculation in a higher-dimensional space, so the regression model can be further represented as

$$f(z) = w^T \varphi(z) = \sum_{i=1}^{n} \alpha_i \kappa(z, x_i) \tag{9}$$

This is solved to obtain filter parameters:

$$\alpha = (K + \lambda I)^{-1} y, \tag{10}$$

where K is the circulant matrix obtained by cyclically shifting the autocorrelation vector k^{xx} of the base sample, and $k^{xx} = \kappa(x, x)$. Utilizing the properties of the circulant matrix, K is diagonalized to yield the following equation:

$$K = C(k^{xx}) = F \cdot diag(\hat{k}^{xx}) \cdot F^H \tag{11}$$

By substituting Equation (11) into Equation (10), we can obtain

$$\begin{aligned} \alpha &= (K + \lambda I)^{-1} y \\ &= (F \cdot diag(\hat{k}^{xx}) \cdot F^H + \lambda I)^{-1} y \\ &= F \cdot diag((\hat{k}^{xx}) + \lambda)^{-1} \cdot F^H \cdot y \end{aligned} \tag{12}$$

By performing Fourier transformation on both sides of the equation, we can obtain

$$\hat{\alpha} = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{13}$$

4.2.3. Kernel Correlation Filter

By operating the filter parameters with the test samples, the region with the maximum response value in the response map represents the output of the object tracking. The response results are presented as follows:

$$f(z) = (K^z)^T \alpha, \tag{14}$$

where K^z denotes the kernel correlation matrix between the detection sample and the training sample, calculated as follows:

$$K^z = C(k^{xz}), \tag{15}$$

where k^{xz} represents the kernel correlation between the training base sample x and the detection sample z. K^z is also a circulant matrix, subjected to diagonalization:

$$K^z = F \cdot diag(\hat{k}^{xz}) \cdot F^H \tag{16}$$

By substituting Equation (16) into Equation (14) and performing Fourier transformation on both sides, we can obtain

$$\hat{f}(z) = \hat{k}^{xz} \odot \hat{\alpha}. \tag{17}$$

During the tracking process, both the target and background undergo dynamic changes. To enhance model stability, a linear interpolation method is introduced in this paper for template updates. This method involves updating the model parameters α and target templates x of the classifier using the following strategy:

$$\begin{matrix} \alpha_t = (1-\eta)\alpha_{t-1} + \eta\alpha'_t \\ x_t = (1-\eta)x_{t-1} + \eta x'_t \end{matrix}, \tag{18}$$

where η represents the model update rate; t denotes the number of frames in the video sequence; and α_{t-1} and α'_t represent the model parameters of the $t-1$ frame image and the t frame image while signifying the target templates of the $t-1$ frame image and the t-th frame image, respectively.

Based on the derivation of the above formulas, it can be determined that, during the process of object tracking, the rapid acquisition of the current target appearance status is achieved through the update of filter parameters, which enhances the algorithm's adaptability. However, the algorithm updates parameters for each frame of the image, and the learning rate η remains constant. When the underwater target is occluded or there is a sudden change in lighting, background noise is introduced, leading to model drift. Therefore, it is necessary to design a reasonable mechanism for updating the model parameters. Hence, we propose a multi-layer target loss discrimination mechanism in Section 4.3.

4.3. Design of the Multi-Layer Target Loss Discriminator

The KCF algorithm lacks a target loss discrimination mechanism, resulting in template updates even when the target is occluded. This introduces background noise, leading to target localization failures. This paper introduces a three-level object tracking confidence discriminator to act as an early warning for target loss. The first-level discrimination employs the filtering maximum response score, the second-level discrimination uses the average peak-to-correlation energy, and the third-level discrimination is based on multi-peak counts.

For each frame in the image sequence, object tracking calculations are made. If target loss is detected, then the template updates in the tracking algorithm are halted, and a lightweight object detector is activated for target repositioning.

① Top layer: maximum response score. The top-layer discrimination utilizes the filtering maximum response score. The response score, denoted as F, is the peak value

resulting from convolving the input image with the filter template. $Fmax$ represents the maximum response score, $Fmin$ represents the minimum response score, and $Fthre$ is the threshold response score. The maximum peak set is represented as $\{Fmax_i | i = 1, 2, \ldots n\}$.

$$\begin{array}{l} u_1 = \sum\limits_{i=1}^{n} Fmax_i / n \\ {\sigma_1}^2 = \sum\limits_{i=1}^{n} (Fmax_i - m_1)^2 \Big/ n \end{array}, \tag{19}$$

where u_1 signifies the mean of the maximum response scores over a recent period of time before the current frame, and ${\sigma_1}^2$ represents their variance.

② Second layer: average peak correlated energy. The second-layer discrimination employs the average peak-to-correlation energy (APCE) [47], which reflects the fluctuation in response maps and the confidence level of detecting the target. When the target is lost, there is significant fluctuation. The formula is as follows:

$$APCE = \frac{|F_{max} - F_{min}|^2}{mean(\sum\limits_{w,h} (F_{w,h} - F_{min})^2)}, \tag{20}$$

$$\begin{array}{l} u_2 = \sum\limits_{i=1}^{n} APCE_i / n \\ {\sigma_2}^2 = \sum\limits_{i=1}^{n} (APCE_i - \mu_2)^2 \Big/ n \end{array}, \tag{21}$$

where u_2 represents the mean of the APCE over a recent period of time before the current frame, and ${\sigma_2}^2$ represents its variance.

③ Third layer: the number of multiple peaks. The third-level discrimination employs a multi-peak count-based criterion. Experimental results have shown that, when there are multiple peaks in the response map, the object tracking performance tends to be non-stationary. n represents the number of response peaks as follows:

$$n = num(F > Fthre), \tag{22}$$

It is assumed that the stability discrimination factor is ξ, $Fmax_p$ is the maximum response score for the p-th frame image, $APCE_p$ is the average peak-to-correlation energy for the p-th frame image, and γ is a positive real number. The current tracking state is considered stable when the following conditions are satisfied and ξ is set to 1:

$$\begin{cases} Fmax_p > |u_1 \pm \gamma\sigma_1|^2 \\ APCE_p > |u_2 \pm \gamma\sigma_2|^2 \\ n < 3 \end{cases} \tag{23}$$

If the conditions mentioned above are not met, it signifies that the present tracking status is unsuccessful, and $\xi = 0$. When $\xi = 0$, the re-detection function is activated, which is discussed in Section 4.1 regarding the lightweight detection network.

According to the multi-layer discrimination mechanism, the learning rate is dynamically adjusted, and adaptive template updates are performed. As for the adaptive adjustment parameter ω_t, it can take on two different values depending on the tracking state, as shown in Equation (24). When the target is in an unobstructed state and the tracking state is good, it follows an exponential distribution. However, when the target is under the circumstances of severe occlusion or appearance change, ω_t is set to 0 to avoid model contamination.

$$\omega_t = \begin{cases} 2^{Fmax} - 1 & (Fmax_p < Fmax_{TH}) \; and \; (APCE_p < APCE_{TH}) \; and \; (n < 3) \\ 0 & else \end{cases} \tag{24}$$

After obtaining the adaptive adjustment parameters, the parameters and target template information update formula are given as shown in Equation (25):

$$\begin{aligned} \alpha_t &= (1-\eta\omega_t)\alpha_{t-1} + \eta\omega_t\alpha'_t \\ x_t &= (1-\eta\omega_t)x_{t-1} + \eta\omega_t x'_t \end{aligned}, \tag{25}$$

where η is the model update rate. t is the number of frames in the video sequence. α_{t-1} is the model parameters of the $t-1$ frame. α'_t is the model parameters of the t frame. x_{t-1} is the target template of the $t-1$ frame. x'_t is the target template of the t frame.

5. Experimental Analysis

5.1. Lightweight Object Detection Experiment

5.1.1. Dataset and Evaluation Metrics

To comprehensively evaluate the proposed lightweight underwater detection model, we leverage several public underwater detection datasets designed for AUVs. The Target Recognition Group of China Underwater Robot Professional Competition (URPC) [48] serves as an essential underwater object detection benchmark dataset, encompassing diverse and challenging underwater detection scenes. The URPC is specialized in detecting underwater marine organisms like holothurian, echinus, scallop, and starfish. Additionally, we incorporate the Real-World Underwater Object Detection (RUOD) dataset [49], which comprises 14,000 high-resolution images, 74,903 labeled objects, and 10 common aquatic categories; this dataset was captured in the wild and is widely recognized in the underwater object detection community. Additionally, the URPC and RUOD datasets are utilized in their standard format for model training and evaluation.

For the evaluation metric of the multi-class object detection task, we adopt the mean average precision (mAP), which is commonly used to evaluate overall model performance. Average precision is the area under the precision–recall (PR) curve, representing the average precision for that category. The PR curve is the curve formed by calculating the precision and recall rates, and the formulas for precision and recall are as follows:

$$Precision = \frac{TP}{TP+FP}, \tag{26}$$

$$Recall = \frac{TP}{TP+FN}, \tag{27}$$

where TP represents the true-positive instances, which are positive samples correctly predicted as positive by the model. FP represents the false-positive instances, which are negative samples incorrectly predicted as positive by the model. FN represents the false-negative instances, which are positive samples incorrectly predicted as negative by the model.

5.1.2. Experimental Settings

Our model is implemented using PyTorch 1.8.0 and CUDA 10.2 on Nvidia GTX 3090 GPUs equipped with 128 GB of RAM. During the training phase, we initialize the model with random parameters and employ a batch size of 16 during training. The initial learning rate is set to $3e^{-5}$, and we train the network for 100 epochs using the SGD optimizer and the mini-batch StepLR descent algorithm. To enhance the model's robustness, we apply various data augmentation techniques during training, including random flipping (with a 50% probability), global random scaling (ranging from 0.90 to 1.10 percent of the range), and hue adjustment (with a 30% probability). During the testing phase, we quantize our trained model using INT8 inference accuracy to ensure deployment on common embedded mobile phone platforms for AUV applications.

5.1.3. Experimental Results and Analysis

In this section, to verify the detection accuracy and speed performance of our proposed lightweight underwater object detection model, we compare our proposed detection methods with other well-known lightweight detection models. As YOLO is the most classical one-stage object method, we select YOLOv5-n [50], YOLOv7-tiny [51], and YOLOv8-n [52] as the representative comparison baselines. All the baseline models and our proposed model are trained using the recommended training settings to ensure fair comparisons. Tables 1 and 2 show the detection performance of each class using our model and the baselines in detail. Large or dark targets like echinus are easily detected, whereas small and light-colored targets like scallops are more likely to be missed during detection. The experiment results demonstrate that our proposed lightweight underwater object detection method outperforms the baselines in terms of both accuracy and inference speed. While the mAP@0.50 of our model achieves 0.753 on the URPC datasets and 0.813 on the RUOD datasets, the single model achieves an inference time of 38 ms in Hisilicon Kirin 950 with the deployment of Android Neural Networks API (NNAPI).

Table 1. Per-class AP on the URPC dataset for different object detection methods.

Network	AP				Evaluation Index
	Echinus	Holothurian	Scallop	Starfish	mAP
YOLOv5-n	0.8651	0.6465	0.6116	0.7421	0.716
YOLOv7-tiny	0.8592	0.6639	0.6003	0.7074	0.708
YOLOv8-n	0.8891	0.6832	0.6237	0.7281	0.731
Ours	0.9002	0.7035	0.6355	0.773	0.753

Table 2. Per-class AP on the RUOD dataset for different object detection methods.

Network	AP										Evaluation Index
	Holothurian	Echinus	Scallop	Starfish	Fish	Corals	Diver	Cuttlefish	Turtle	Jellyfish	mAP
YOLOv5-n	0.61	0.89	0.84	0.86	0.68	0.62	0.73	0.65	0.80	0.91	0.759
YOLOv7-tiny	0.63	0.91	0.79	0.83	0.65	0.63	0.73	0.63	0.74	0.88	0.742
YOLOv8-n	0.68	0.95	0.93	0.82	0.71	0.72	0.71	0.68	0.82	0.93	0.795
Ours	0.71	0.96	0.92	0.86	0.75	0.70	0.75	0.70	0.85	0.94	0.813

5.2. Long-Term Tracking Experiment

5.2.1. Evaluation Metrics

We employ the One-Pass Evaluation (OPE) [28] with precision and success plot metrics to assess tracking performance.

The evaluation of video tracking performance is based on accuracy and the success rate. Accuracy is defined as the ratio of frames where the target center position error is less than 20 pixels. The center position error is calculated as the average Euclidean distance between the algorithm-derived center position and the true center position, with a defined error threshold of 20 pixels. The center position error can be calculated as follows:

$$CLE = \sqrt{(x_{pre} - x_{gt})^2 + (y_{pre} - y_{gt})^2} \tag{28}$$

The success rate is used as another metric to assess the overall effectiveness of the tracking algorithm. The success rate of the algorithm calibration is defined as the proportion of frames with an overlap ratio greater than the threshold within the total number of frames, where the threshold is set to 0.5. The intersection over union (IoU) [53] refers to the area between the algorithm-calibrated bounding box and the ground truth bounding box, which is calculated as follows:

$$S = \frac{|A_{pre} \cap A_{gt}|}{|A_{pre} \cup A_{gt}|} \tag{29}$$

5.2.2. Experiment Analysis of the Multi-Layer Target Loss Discriminator

In practical underwater scenarios, autonomous underwater vehicles (AUVs) capture multiple sets of tracking videos of fish targets, conducting tracking tests. In the 131th frame in the experimental results, the fish's motion deformation is significant, leading to delayed template updates and target loss. At this point, the number of peaks increases significantly. The maximum peak value and APCE (average peak-to-correlation energy) value sharply decrease, with significant numerical fluctuations. Figure 6a–d correspond to the filtering response graphs of frame 5, frame 99, frame 131, and frame 143, respectively. The experiment obtained the third-layer target loss discriminator: the number of multiple peaks. In Figure 6, the number of red ellipses indicates the number of peaks (n). It is obvious that n = 1 (in frame 99) when the tracking state is stable and n = 6 (in frame 131) when it is unstable.

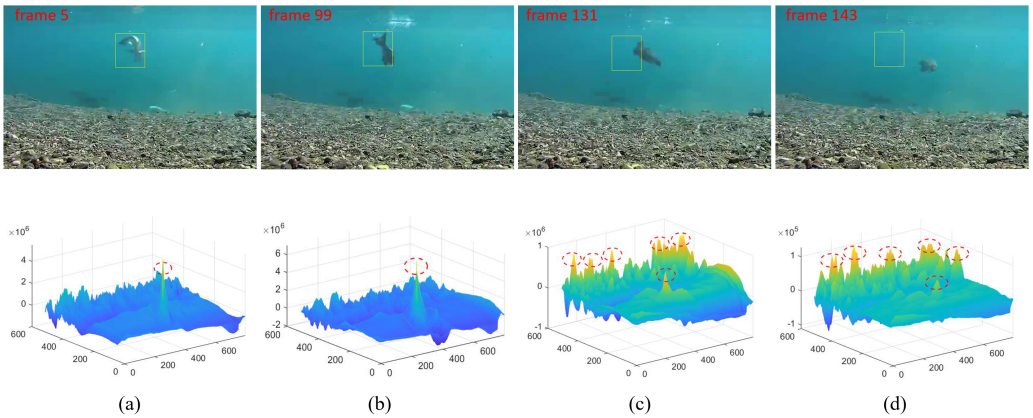

Figure 6. Examples of multiple numbers of peaks for target loss discrimination. (**a**) Frame 5. (**b**) Frame 99. (**c**) Frame 131. (**d**) Frame 143. The yellow boxes denote the fish tracking results.

The experiment obtained the top-layer and second-layer target loss discriminators. Figure 7a shows the discriminative results of the maximum response score, where the horizontal axis represents the frame number, and the vertical axis represents the maximum response score calculated for each frame. Figure 7b presents the results of the average peak correlation energy (APCE), with the horizontal axis denoting the frame number and the vertical axis indicating the average peak correlation energy value. Through the analysis of the experiments, the following conclusions are drawn:

(1) In the initial stable state of object tracking, the maximum response score (Fmax) and average peak correlation energy (APCE) values are relatively high. In the 5th frame, the experiment calculates Fmax = 0.391 and APCE = 29.07.
(2) In subsequent scenarios where object tracking is successful, the Fmax and APCE values remain high. In the 99th frame, the experiment calculates Fmax = 0.397 and APCE = 27.84.
(3) When the target is lost, both the Fmax and APCE values sharply decrease, exhibiting significant fluctuations. In the 131th frame when the target is lost, Fmax = 0.287 and APCE = 17.78.
(4) After the target is lost, the correlation filter tracker introduces background error information, treating the background as the target and continuing tracking, leading to a lower value. In the 143th frame, Fmax = 0.237 and APCE = 13.34.

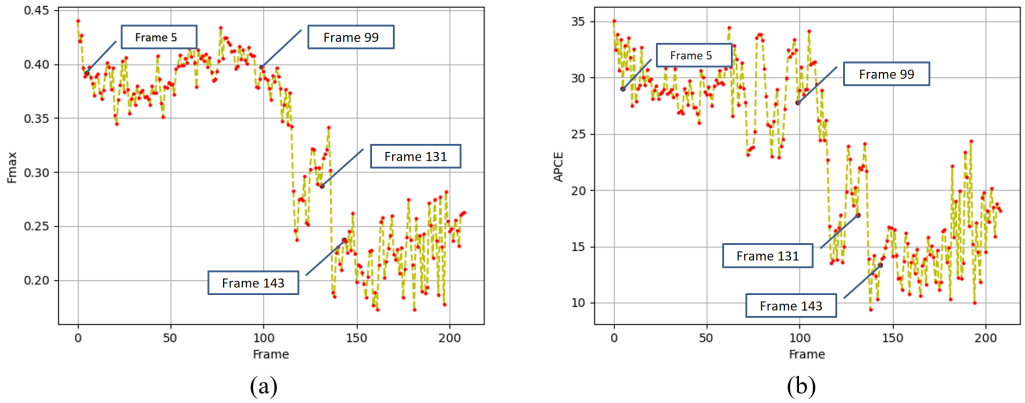

Figure 7. (**a**) The result of maximum response score discrimination. (**b**) The result of average peak-to-correlation energy.

From the above experiments, it can be concluded that the maximum response scores, average peak correlation energy values, and multi-peak number values can serve as discriminative criteria for object tracking states. Adaptive template and parameter updates can be performed based on different tracking scenarios.

5.2.3. Tracking Model performance comparison

We conducted tracking result testing and visualization based on publicly available underwater scene datasets and actual measurement data. The tracking videos include marine organisms such as fish, turtle, jellyfish, cuttlefish, starfish, and octopus, as well as divers. The challenges of video encompass the motion deformation of targets and variations in target scale and size. The underwater scene tracking video contains a total of 35 sequences, and each sequence has a length of approximately 100 to 1000 frames.

We conducted performance comparison evaluations using 13 algorithms, namely, CF2, ECO, BACF, SiamFC, CSR_DCF, SRDCF, KCF, EFSCF, AS2RCF, SOCF, OTFUR [54], IMKCF [55], and ours. As illustrated in Table 3, the proposed algorithm achieved favorable results across various challenges, with accuracy and success rates of 88.3% and 64.2%, respectively. This represents an improvement of 16.4% and 11.9% compared to the KCF algorithm. This represents an improvement of 2.8% and 5.83% compared to the CF2 algorithm. Additionally, OPE precision plots and OPE success rate plots are shown in Figure 8a and Figure 8b, respectively. The long-term tracking performance of our algorithm in a challenging underwater environment is demonstrated.

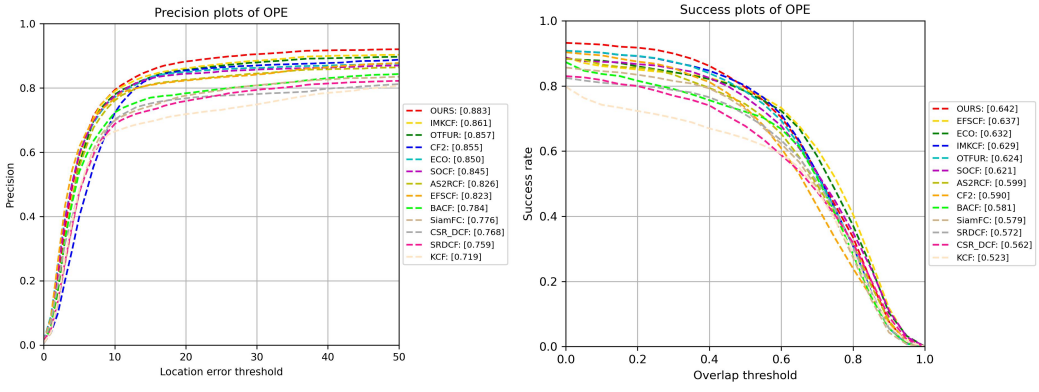

Figure 8. (**a**) OPE precision plots for the underwater visual data. (**b**) OPE success rate plots for the underwater visual data.

For a more intuitive display of the tracking results, we employed multiple algorithms to assess performance and visualize the tracking results for some typical videos. As depicted in Figure 9, when faced with fast motion challenges (Figure 9a) and low-light scenarios (Figure 9b), the proposed ULOTrack exhibits robust performance. Additionally, ULOTrack is better at handling scale variation (Figure 9c) and challenges involving deformation (Figure 9d).

(a) (b) (c) (d) (e)

Figure 9. The visualization perception results of our proposed model in various scenes; the box denotes the tracked marine organisms. (**a**) Example of low-light challenge (fish 1). (**b**) Example of fast motion challenge (fish 2). (**c**) Example of the motion deformation challenge (octopus). (**d**) Example of the challenge in target size (turtle). (**e**) Example of complex scene, including several potential tracking objects (fish 3). The boxes denote the fish tracking results.

The experiments demonstrated the effectiveness of the multi-layer tracking performance discriminator, one-dimensional scale filter, and re-detection algorithm. Notably, the proposed tracking algorithm is capable of handling challenging scenarios, such as complex backgrounds, scale variations, changes in lighting conditions, fast motion, deformation, and in-plane rotations. This substantiates the robustness of the tracker in challenging underwater environments.

Table 3. The accuracy and success rates for different trackers. Bold value means the best performance.

Tracker	Accuracy (Acc)	Success Rate (SR)	Frame Rate (FPS)
CF2 [2015]	0.855	0.590	43 (GPU)
ECO [2017]	0.850	0.632	50 (GPU)
BACF [2017]	0.784	0.581	35 (CPU)
SiamFC [2016]	0.776	0.579	58 (GPU)
CSR_DCF [2017]	0.768	0.562	13 (CPU)
SRDCF [2018]	0.759	0.572	5 (CPU)
KCF [2015]	0.719	0.523	172 (CPU)
EFSCF [2023]	0.823	0.637	18 (CPU)
AS2RCF [2023]	0.826	0.599	20 (GPU)
OTFUR [2023]	0.857	0.624	61 (GPU)
SOCF [2024]	0.845	0.621	48 (GPU)
IMKCF [2024]	0.861	0.629	16 (CPU)
Ours	**0.883**	**0.642**	42 (CPU)

5.2.4. Ablation Experiment of Different Modules

We completed an ablation experiment based on ULOTrack to showcase the efficacy of the different modules. The experiment focused on analyzing the effects of the object re-detection mechanism, multi-layer tracking discriminator, adaptive template updates, and multi-feature fusion on the overall performance of the algorithm. Table 4 presents the tracking results of the ablation experiment.

Table 4. Ablation experiment of the proposed modules on the test set. Bold value means the best performance.

Model	Modules				Evaluation Index	
	Re-Detection	Multi-Layer Tracking Discriminator (MTD)	Adaptive Template Updates (ATUs)	Multi-Feature Fusion (MFF)	Acc	SR
Without re-detection	✘	✔	✔	✔	0.802	0.568
Without MTD	✔	✘	✔	✔	0.821	0.617
Without ATU	✔	✔	✘	✔	0.856	0.629
Without MFF	✔	✔	✔	✘	0.834	0.590
OURS	✔	✔	✔	✔	**0.883**	**0.642**

Table 4 reflects the impact of re-detection on tracking accuracy using ULOTrack. When the re-detection mechanism is not utilized, ULOTrack degrades into a short-term tracker, and the experimental results show that it cannot independently handle model drift. Specifically, the tracking accuracy and success rate are improved by 8.1% and 7.4%, respectively. When the multi-feature fusion strategy is not utilized, the method is degraded to a single HOG feature extract, causing unstable feature extraction. The algorithm employs an adaptive template update mechanism that can improve accuracy by 2.7%. Additionally, the multi-layer tracking discriminator can directly assess the current tracking state and guide subsequent decisions accordingly.

In addition, we provide the tracking results obtained under challenges such as lighting variations, motion blur, fast motion, and deformation in the benchmark tests. Figure 10 demonstrates the tracking results under different environmental conditions. In the first sequence, significant lighting changes occur in frames 100 and 206. Other algorithms struggle to adapt to these lighting changes, causing the position of the rectangular target box to gradually shift to the left. Our algorithm introduces a multi-feature fusion strategy, which effectively handles lighting variations. To address the issue of cluttered backgrounds, we employ an adaptive template update mechanism that avoids introducing background noise due to overly fast or slow model updates, allowing for robust target tracking. In the second and third sequences, the target undergoes fast motion and deformation, resulting in substantial tracking errors with other methods. Our method successfully relocates the

target benefit from the re-detection mechanism, leading to more stable tracking results. The right figure shows the tracking confidence curves.

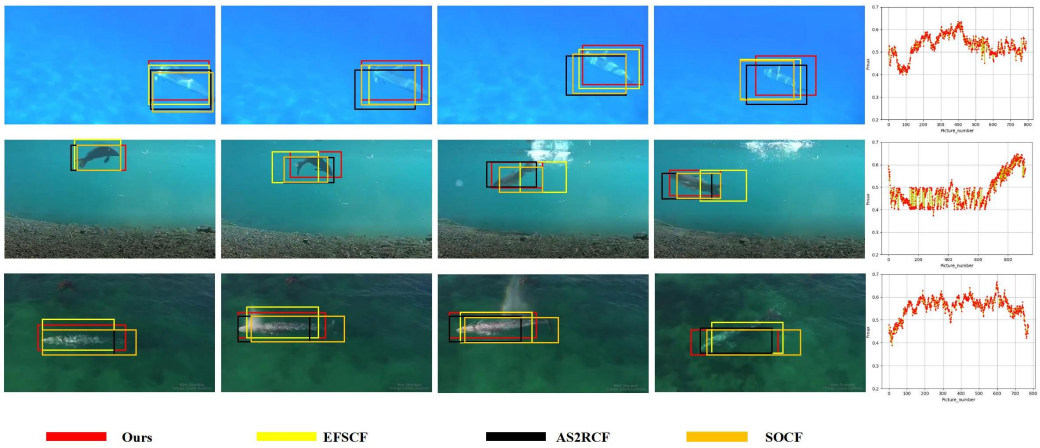

Figure 10. The tracking results of different models in various scenes.

Figure 11 provides a detailed illustration of how ULOTrack influences the ability to localize targets. The heat map represents the level of attention that the model assigns to different search regions. We observed that the activation regions encompass broader global areas of the marine organisms. For example, in Figure 11b, ULOTrack focuses on the global features of the octopus, while UOSTrack maintains the stable tracking of the fish even in complex underwater environments. In contrast, other methods tend to have more dispersed heat maps, activating multiple regions in scenarios involving motion deformation and complex backgrounds, as shown in Figure 11c,d.

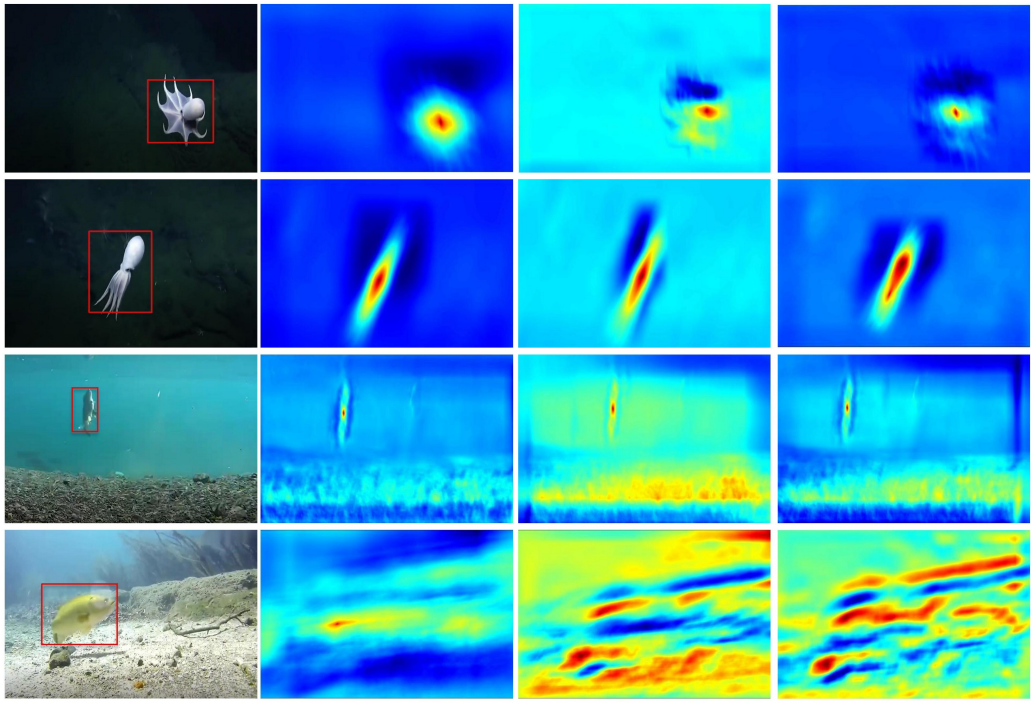

Figure 11. Examples of feature visualization on the different sequences. The red box is the ground truth of the target. (**a**) Underwater image and ground truth. (**b**) Ours. (**c**) EFSCF. (**d**) AS2RCF. The red areas represent high correlations.

6. Conclusions

In this study, we propose ULOTrack, an underwater long-term object tracking algorithm for capturing marine organisms. Experimental results demonstrate that ULOTrack excels in challenging scenarios such as variations in lighting, scale changes, and target deformation. Compared to existing methods, our approach achieves higher tracking accuracy and success rates. The multi-layer tracking performance discriminator in ULOTrack plays a crucial role in reducing template drift. This multi-layer mechanism allows the tracker to pause template updates when target loss is detected, thereby activating a lightweight detector for target re-localization. Additionally, the multi-feature fusion and adaptive template update mechanisms prove effective in adapting to the rapid movements and appearance changes of marine organisms.

However, our study has some limitations. Although ULOTrack performed well in experiments, it faced certain challenges in handling extreme occlusions and highly complex backgrounds. Furthermore, while the lightweight detector effectively balances accuracy and computational efficiency, its performance could be further enhanced for more complex targets. Future work will focus on improving the robustness of the detection module in noisy and dynamic backgrounds and further optimizing computational efficiency for extended continuous tracking.

Author Contributions: Conceptualization, Y.Y. and J.H.; methodology, Y.Y. and J.H.; validation, J.H. and H.X.; writing—original draft preparation, J.H., Y.Y., H.X. and H.W.; writing—review and editing, J.H., Y.Y., H.X. and H.W.; supervision, Y.Y.; project administration, Y.Y.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program under grants 2021YFC2803000 and 2021YFC2803001.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the author (J.H.).

Acknowledgments: The authors acknowledge editors and reviewers for comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. He, J.; Xu, H.; Li, S.; Yu, Y. Efficient SonarNet: Lightweight CNN Grafted Vision Transformer Embedding Network for Forward-Looking Sonar Image Segmentation. *IEEE Trans. Geosci. Remote. Sens.* **2024**, *62*, 4210317. [CrossRef]
2. Whitt, C.; Pearlman, J.; Polagye, B.; Caimi, F.; Muller-Karger, F.; Copping, A.; Spence, H.; Madhusudhana, S.; Kirkwood, W.; Grosjean, L.; et al. Future vision for autonomous ocean observations. *Front. Mar. Sci.* **2020**, *7*, 697. [CrossRef]
3. Zhang, J.; Liu, M.; Zhang, S.; Zheng, R.; Dong, S. Multi-AUV adaptive path planning and cooperative sampling for ocean scalar field estimation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–14. [CrossRef]
4. Yu, J.; Wu, Z.; Yang, X.; Yang, Y.; Zhang, P. Underwater target tracking control of an untethered robotic fish with a camera stabilizer. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, *51*, 6523–6534. [CrossRef]
5. Prasad, D.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1993–2016. [CrossRef]
6. Melo, J.; Matos, A. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Eng.* **2017**, *139*, 250–264. [CrossRef]
7. Li, J.; Zhang, G.; Jiang, C.; Zhang, W. A survey of maritime unmanned search system: Theory, applications and future directions. *Ocean Eng.* **2023**, *285*, 115359. [CrossRef]
8. Xu, H.; Zhang, X.; He, J.; Geng, Z.; Yu, Y.; Cheng, Y. Panoptic Water Surface Visual Perception for USVs using Monocular Camera Sensor. *IEEE Sens. J.* **2024**, *15*, 24263–24274. [CrossRef]
9. Xu, H.; Zhang, X.; He, J.; Yu, Y.; Cheng, Y. Real-time Volumetric Perception for unmanned surface vehicles through fusion of radar and camera. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–12. [CrossRef]
10. Xu, H.; Zhang, X.; He, J.; Geng, Z.; Pang, C.; Yu, Y. Surround-view Water Surface BEV Segmentation for Autonomous Surface Vehicles: Dataset, Baseline and Hybrid-BEV Network. *IEEE Trans. Intell. Veh.* **2024**, *10*, 1–15. [CrossRef]
11. Panetta, K.; Kezebou, L.; Oludare, V.; Agaian, S. Comprehensive underwater object tracking benchmark dataset and underwater image enhancement with GAN. *IEEE J. Ocean. Eng.* **2021**, *47*, 59–75. [CrossRef]

12. Sun, C.; Wan, Z.; Huang, H.; Zhang, G.; Bao, X.; Li, J.; Sheng, M.; Yang, X. Intelligent target visual tracking and control strategy for open frame underwater vehicles. *Robotica* **2021**, *39*, 1791–1805. [CrossRef]
13. Wu, X.; Han, X.; Zhang, Z.; Wu, H.; Yang, X.; Huang, H. A hybrid excitation model based lightweight siamese network for underwater vehicle object tracking missions. *J. Mar. Sci. Eng.* **2023**, *11*, 1127. [CrossRef]
14. Li, X.; Wei, Z.; Huang, L.; Nie, J.; Zhang, W.; Wang, L. Real-time underwater fish tracking based on adaptive multi-appearance model. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018 ; IEEE: Piscataway, NJ, USA, 2018; pp. 2710–2714.
15. Chuang, M.C.; Hwang, J.N.; Ye, J.H.; Huang, S.C.; Williams, K. Underwater fish tracking for moving cameras based on deformable multiple kernels. *IEEE Trans. Syst. Man, Cybern. Syst.* **2016**, *47*, 2467–2477. [CrossRef]
16. Lu, Y.; Wang, H.; Chen, Z.; Zhang, Z. Multi-scale underwater object tracking by adaptive feature fusion. In Proceedings of the International Symposium on Artificial Intelligence and Robotics 2021, Fukuoka, Japan, 21–22 August 2021; SPIE: Bellingham, WA, USA, 2021; Volume 11884, pp. 346–357.
17. Mayer, C.; Danelljan, M.; Paudel, D.P.; Van Gool, L. Learning target candidate association to keep track of what not to track. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 13444–13454.
18. Huang, Y.; Huang, H.; Niu, M.; Miah, M.S.; Wang, H.; Gao, T. UAV Complex-Scene Single-Target Tracking Based on Improved Re-Detection Staple Algorithm. *Remote Sens.* **2024**, *16*, 1768. [CrossRef]
19. Gao, Z.; Zhuang, Y.; Gu, J.; Yang, B.; Nie, Z. A joint local–global search mechanism for long-term tracking with dynamic memory network. *Expert Syst. Appl.* **2023**, *223*, 119890. [CrossRef]
20. Li, G.; Nai, K. Robust tracking via coarse-to-fine redetection and spatial-temporal reliability evaluation. *Expert Syst. Appl.* **2024**, *256*, 124927. [CrossRef]
21. Liu, C.; Zhao, J.; Bo, C.; Li, S.; Wang, D.; Lu, H. LGTrack: Exploiting Local and Global Properties for Robust Visual Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 8161–8171. [CrossRef]
22. Fan, B.; Cong, Y.; Du, Y. Discriminative multi-task objects tracking with active feature selection and drift correction. *Pattern Recognit.* **2014**, *47*, 3828–3840. [CrossRef]
23. Zhang, Y.; Gao, X.; Chen, Z.; Zhong, H.; Li, L.; Yan, C.; Shen, T. Learning salient features to prevent model drift for correlation tracking. *Neurocomputing* **2020**, *418*, 1–10. [CrossRef]
24. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
25. Zhou, S.K.; Chellappa, R.; Moghaddam, B. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Process.* **2004**, *13*, 1491–1506. [CrossRef]
26. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
27. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 2544–2550.
28. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 702–715.
29. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014; Bmva Press: Durham, UK, 2014.
30. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef]
31. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
32. Lukezic, A.; Vojir, T.; Cehovin Zajc, L.; Matas, J.; Kristan, M. Discriminative correlation filter with channel and spatial reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.
33. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
34. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning spatial-temporal regularized correlation filters for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4904–4913.
35. Wen, J.; Chu, H.; Lai, Z.; Xu, T.; Shen, L. Enhanced robust spatial feature selection and correlation filter learning for UAV tracking. *Neural Netw.* **2023**, *161*, 39–54. [CrossRef]
36. Zhang, J.; He, Y.; Wang, S. Learning adaptive sparse spatially-regularized correlation filters for visual tracking. *IEEE Signal Process. Lett.* **2023**, *30*, 11–15. [CrossRef]
37. Ma, S.; Zhao, B.; Hou, Z.; Yu, W.; Pu, L.; Yang, X. SOCF: A correlation filter for real-time UAV tracking based on spatial disturbance suppression and object saliency-aware. *Expert Syst. Appl.* **2024**, *238*, 122131. [CrossRef]

38. Xia, R.; Chen, Y.; Ren, B. Improved anti-occlusion object tracking algorithm using Unscented Rauch-Tung-Striebel smoother and kernel correlation filter. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6008–6018. [CrossRef]
39. Cui, S.; Wang, Y.; Wang, S.; Wang, R.; Wang, W.; Tan, M. Real-time perception and positioning for creature picking of an underwater vehicle. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3783–3792. [CrossRef]
40. Lee, D.; Kim, G.; Kim, D.; Myung, H.; Choi, H.T. Vision-based object detection and tracking for autonomous navigation of underwater robots. *Ocean Eng.* **2012**, *48*, 59–68. [CrossRef]
41. Chuang, M.C.; Hwang, J.N.; Williams, K.; Towler, R. Tracking live fish from low-contrast and low-frame-rate stereo videos. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 167–179. [CrossRef]
42. Rout, D.K.; Subudhi, B.N.; Veerakumar, T.; Chaudhury, S. Walsh–Hadamard-kernel-based features in particle filter framework for underwater object tracking. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5712–5722. [CrossRef]
43. Bhat, P.G.; Subudhi, B.N.; Veerakumar, T.; Laxmi, V.; Gaur, M.S. Multi-feature fusion in particle filter framework for visual tracking. *IEEE Sens. J.* **2019**, *20*, 2405–2415. [CrossRef]
44. Li, Y.; Wang, B.; Li, Y.; Liu, Z.; Huo, W.; Li, Y.; Cao, J. Underwater object tracker: UOSTrack for marine organism grasping of underwater vehicles. *Ocean Eng.* **2023**, *285*, 115449. [CrossRef]
45. He, J.; Chen, J.; Xu, H.; Yu, Y. SonarNet: Hybrid CNN-Transformer-HOG Framework and Multifeature Fusion Mechanism for Forward-Looking Sonar Image Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–17. [CrossRef]
46. Li, W.; Chen, C.; Su, H.; Du, Q. Local Binary Patterns and Extreme Learning Machine for Hyperspectral Imagery Classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3681–3693. [CrossRef]
47. Wang, M.; Liu, Y.; Huang, Z. Large margin object tracking with circulant feature maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4021–4029.
48. Liu, C.; Li, H.; Wang, S.; Zhu, M.; Wang, D.; Fan, X.; Wang, Z. A dataset and benchmark of underwater object detection for robot picking. In Proceedings of the 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 5–9 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
49. Fu, C.; Liu, R.; Fan, X.; Chen, P.; Fu, H.; Yuan, W.; Zhu, M.; Luo, Z. Rethinking general underwater object detection: Datasets, challenges, and solutions. *Neurocomputing* **2023**, *517*, 243–256. [CrossRef]
50. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
51. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
52. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 1 June 2023).
53. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
54. Lee, M.F.R.; Chen, Y.C. Artificial intelligence based object detection and tracking for a small underwater robot. *Processes* **2023**, *11*, 312. [CrossRef]
55. Yue, W.; Xu, F.; Yang, J. Tracking-by-Detection Algorithm for Underwater Target Based on Improved Multi-Kernel Correlation Filter. *Remote Sens.* **2024**, *16*, 323. [CrossRef]