applied sciences

MDPI

Article

Multi-Object Detection for Inland Ship Situation Awareness Based on Few-Shot Learning

Junhui Wen [1,2], Maciej Gucma [3], Mengxia Li [4,5] and Junmin Mou [1,2,*]

[1] Hubei Key Laboratory of Inland Shipping Technology, Wuhan University of Technology, Wuhan 430063, China; junhuiwen@whut.edu.cn
[2] School of Navigation, Wuhan University of Technology, Wuhan 430063, China
[3] Faculty of Navigation, Maritime University of Szczecin, 70-500 Szczecin, Poland; m.gucma@am.szczecin.pl
[4] Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China; limengxia@whut.edu.cn
[5] State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan 430063, China
* Correspondence: moujm@whut.edu.cn

Abstract: With the rapid development of artificial intelligence technology and unmanned surface vehicle (USV) technology, object detection and tracking have wide applications in marine monitoring and intelligent ships. However, object detection and tracking tasks on small sample datasets often face challenges due to insufficient sample data. In this paper, we propose a ship detection and tracking model with high accuracy based on a few training samples with supervised information based on the few-shot learning framework. The transfer learning strategy is designed, innovatively using an open dataset of vehicles on highways to improve object detection accuracy for inland ships. The Shuffle Attention mechanism and smaller anchor boxes are introduced in the object detection network to improve the detection accuracy of different targets in different scenes. Compared with existing methods, the proposed method is characterized by fast training speed and high accuracy with small datasets, achieving 84.9% (mAP@0.5) with only 585 training images.

Keywords: ship object detection; multi-object tracking (MOT); few-shot learning (FSL); transfer learning

check for updates

Citation: Wen, J.; Gucma, M.; Li, M.; Mou, J. Multi-Object Detection for Inland Ship Situation Awareness Based on Few-Shot Learning. *Appl. Sci.* **2023**, *13*, 10282. https://doi.org/10.3390/app131810282

Academic Editor: Andrea Prati

Received: 30 August 2023
Revised: 10 September 2023
Accepted: 11 September 2023
Published: 13 September 2023

1. Introduction

Intelligent shipping has become the main direction of the development of the shipping industry, and the autonomous navigation technology of ships is the key to realizing intelligent shipping. As one of the core technologies, situational awareness is the basis for realizing the autonomous navigation of ships. Particularly, detecting and tracking the target ships with collision risks are essential to situational awareness.

In the recent decade, the rapid development of artificial intelligence technology and high-speed processors have promoted ship object detection tracking methods based on computer vision. However, existing high-precision ship object detection and object tracking models rely on vast amounts of data with high-quality labelling [1]. However, samples for inland ships are relatively few, and labeling is time consuming. Moreover, a large amount of data brings a long training time. In addition, the trained models suffer from poor generalization capability. The model application scenario is affected by the annotation of the dataset [2].

1.1. Contributions

Based on the FSL framework, this paper designed a method for inland ship object detection and object tracking using the YOLOv5s lightweight detection network and DeepSORT tracking network. The proposed method is characterized by fast training speed and high accuracy with small datasets. The main contribution of this manuscript is as follows:

- Multi-object detection and tracking for inland ship situation awareness based on FSL are proposed, which achieve 84.9% (mAP@0.5) with only 585 training images.
- The transfer learning strategy is designed by innovatively using an open dataset of vehicles on highways to improve object detection accuracy for inland ships.
- Introducing the Shuffle Attention mechanism and smaller anchor boxes in the object detection network can somewhat improve the detection accuracy of different targets in different scenes.

1.2. Organization

The rest of this paper is organized as follows. Section II briefly introduces the basic method and content of this paper and describes the framework of the whole system. Section III of this paper introduces the related contents, methods, and improvements of object detection. In section IV, this paper introduces the relevant contents, methods, and modifications of target tracking. Section V presents the experimental setup and results. Finally, Section VI summarizes this paper.

2. Related Works

This section aims to provide a comprehensive review of recent studies that are related to object detection and object tracking. Additionally, this section will also cover some of the applications of object detection and object tracking in marine environments.

2.1. Object Detection

Object detection is an essential computer vision task, which is considered the cornerstone of many advanced artificial intelligence tasks. Many object detection algorithms based on deep learning have been applied to the field of ships. Common deep learning detection algorithms include Faster R-CNN [3], MASK-RCNN [4,5], SSD (Single Shot MultiBox Detector) [6], and YOLO (You Only Look Once) [7]. However, surveillance images (such as photographs or videos) for ships are rare, while Synthetic Aperture Radar (SAR) is available all day under all weather. Thus, for ship object detection, methods for detection with SAR images are proposed, such as DDNet [8], Saliency-Based Centernet [8], and Expansion Pyramid Network (SEPN) [9]. For general ship surveillance images, K-means clustering prior box combined with the yolov4 network [10] and SSD_MobilenetV2 [11] have been applied to improve ship detection performance. In these studies, the NMS (Non-Maximum Suppression) effect and detection speed of the network for rectangular boxes are improved, but the detection accuracy and accuracy are decreased. In one study [12], the authors constructed ship datasets in the form of COCO datasets and adopted YOLOX, which introduced residual structure and CIoU loss function. The performance of the algorithm is significantly better than the original YOLOX algorithm. In ref. [13], an improved CenterNet network for ship detection in scale-changing images has been proposed, making the network more sensitive to small objects.

2.2. Object Tracking

Tracking in deep learning is the task of predicting the positions of objects throughout a video using their spatial and temporal features. Traditional object tracking methods can be broadly categorized into several types, including machine-learning-based methods [14–16], Markov random field-based methods [17], and optical flow and background subtraction-based methods [18–21]. For ship object tracking, due to the complexity of the water surface environment, variability in ship scales, and the occlusion of ships, ship object tracking is more challenging. In addition to the above methods, Kaid used particle filtering to track ships based on color histograms within bounding rectangles [22]. Chen et al. proposed an automatic ship object detection and tracking method based on a mean shift to improve the robustness of real-time detection and tracking [23].

One of the challenges for object tracking is the Multi-Object Tracking (MOT) problem. The trackers must track multiple objects and even different classes simultaneously while maintaining high speed. Currently, mainstream MOT methods mainly include detection-based tracking methods (TBD), joint detection and tracking methods (JDT), and transformer-based tracking methods. Deep-learning-based models, such as RAN [24], DMAN [25], TRACK R-CNN [26], STAM [27], and MOT-RNN [28], have also been applied to multi-object tracking. Although MOT research has been widely applied, it has mainly focused on pedestrians and vehicles. For ship tracking, neural-network-based multi-object tracking methods (NN-MOT) have gained much attention from researchers due to their powerful feature extraction capabilities. For example, Tang et al. [29] first introduced deep matching, which uses a deep learning framework to calculate optical flow features and achieved promising tracking results. Hang et al. [30] proposed a fusion network that combines image and point cloud features captured from different modalities to improve the reliability and accuracy of the tracker. Xi et al. proposed an enhanced SiamMask network for coastal ship tracking, which makes the tracking results more subjective. Wen et al. proposed RoDAN (Robust Deep Affinity Network) [31]. Based on DAN (Deep Affinity Network), the ASPP multi-scale fusion module extracts semantic information at different feature scales.

2.3. Literature Summary

To conclude, existing learning-based methods have been highly successful in data-intensive object detection and tracking, but they are often hampered when the dataset is small. Moreover, multi-object detection and tracking for ships still face many challenges: (1) Most object detection and tracking methods focus on data association problems, which overly rely on the quality of detection results. Ship object detection often has small datasets with low annotation quality, which can frequently result in issues, such as frequent ID switches due to poor detection quality. (2) When tracking ships, the large-scale variation in ship scales can easily affect the accuracy of affine similarity in the network output, leading to frequent ID switches. (3) Long-term occlusion problems are that certain types of ships, such as Very Large Ore Carriers (VLOCs), often move at extremely low speeds, which may result in prolonged occlusion duration in ship multi-object detection.

3. Detection and Tracking for Inland Ship Based on Few-Shot Learning Framework

To tackle the problems caused by the small dataset, long training time, and poor generalization, this paper proposes a multi-object detection and tracking approach based on the FSL framework. The multi-object detection method combines YOLOv5 and transfer learning strategies. The DeepSORT tracking algorithm is applied to achieve high detection and tracking quality in small sample data. The FSL framework enables a pre-trained model to generalize over new categories of data using only a few labeled samples per class.

The proposed framework comprises two modules (see Figure 1): detection (including pre-training and fine-tuning) and tracking (including prediction and matching). The pre-training process uses a large volume of high-speed vehicle images as the source domain for training the detection framework. The fine-tuning stage involves using our collected data of Yangtze River ship images to create a few-shot subset for fine-tuning the detection framework. The tracking module combines the detection framework with the DeepSORT tracking framework to achieve effective tracking of inland ships.

YOLO integrates object region prediction and object class prediction into a single neural network model. YOLOv5 has four different versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, with YOLOv5s being the lightest. The motivation for using the YOLOv5s network in this paper is to make the entire ship detection and tracking network more lightweight and faster. In addition, this paper improves the original YOLOv5s object detection network by using the Shuffle Attention mechanism and smaller anchor boxes. Humans can quickly learn about an object with very few samples and apply this to new tasks. Transfer learning partially mimics this characteristic of the human brain: the human genome carries vast knowledge that spans various domains and helps humans quickly adapt to various tasks. Due to the similarity between images of high-speed vehicles and distant ship images, this paper pre-trains the YOLOv5s network using a public high-speed vehicle dataset, UA-DETRAC, to obtain a pre-trained model. Then, the transfer learning strategy is used to fine-tune the model with a small sample ship image dataset collected and well-annotated. In the fine-tuning process, we first freeze the feature extraction part of the network and only update the convolutional layers in the network head to retain some of the network's features from the high-speed image dataset. Then, we unfreeze all network layers to update the weights of all network layers, allowing the model to fit the data better. Next, the detected ship bounding boxes and their class information are inputted into the DeepSORT module, which matches the bounding boxes to tracks. Finally, data association and ID assignment are performed on the successfully matched bounding boxes using a Kalman filter update to obtain the final ship tracking results in the video.

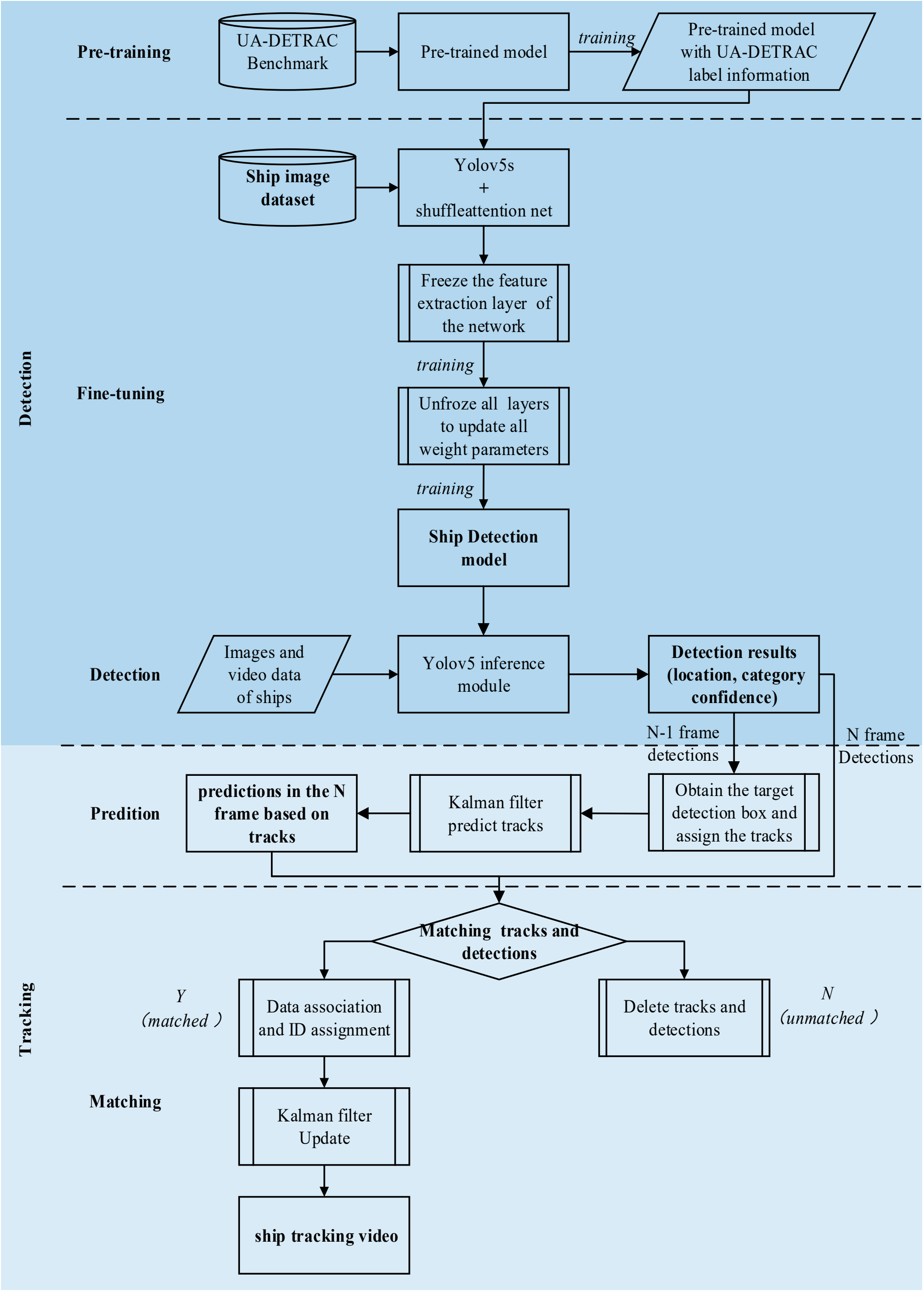

Detection
Pre-training
UA-DETRAC Benchmark
Pre-trained model
training
Pre-trained model with UA-DETRAC label information
Ship image dataset
Yolov5s + shuffleattention net
Fine-tuning
Freeze the feature extraction layer of the network
training
Unfroze all layers to update all weight parameters
training
Ship Detection model
Detection
Images and video data of ships
Yolov5 inference module
Detection results (location, category confidence)
N-1 frame detections
N frame Detections
Predition
predictions in the N frame based on tracks
Kalman filter predict tracks
Obtain the target detection box and assign the tracks
Tracking
Matching tracks and detections
Y (matched)
Data association and ID assignment
Delete tracks and detections
N (unmatched)
Matching
Kalman filter Update
ship tracking video

Figure 1. Ship detection and tracking framework.

In the realm of detection and tracking, various integrated frameworks, such as Hydro-3D, have emerged. Hydro-3D [32] combines advanced object detection features from the V2X-ViT algorithm with historical object tracking data, enabling object inference. It employs a novel 3D neural network for global and local manipulation of tracking data. Hydro-3D strategically utilizes 3D LiDAR for precise object detection and tracking. LiDAR's distance measurements are consistently reliable, regardless of the lighting conditions, making it ideal for both daytime and nighttime operations. However, LiDAR-based detection may be affected by weather-induced inaccuracies and water surface reflections when tracking vessels. Additionally, point cloud data processing involves significant computational demands and real-time constraints.

In our proposed framework, we leverage YOLOv5s as our primary detector, complemented by an enhanced network architecture, aimed at elevating the precision and robustness of ship detection. This profound fusion of technologies equips us with the capability to better adapt to the multifaceted challenges posed by ship tracking across diverse environmental conditions. Furthermore, we employ transfer learning, harnessing the knowledge gleaned from extensive annotated datasets, and seamlessly integrating it into our ship detection and tracking paradigm. This strategic approach serves to enhance the model's generalization prowess, resulting in superior performance across specific task domains. The amalgamation of the DeepSORT algorithm further augments our system's capabilities, enabling real-time target tracking within streaming video feeds, while concurrently affording the provision of highly accurate trajectory information. Our proposed framework demonstrates a remarkable ability to achieve high precision and recall rates even with a limited dataset, attributed in part to its foundation on the YOLOv5s network, renowned for its swift inferencing capabilities and relatively compact parameter size. It is worth noting that in comparison to LiDAR-based detection and tracking, our image-based approach is less susceptible to errors arising from water surface reflections. However, it is prudent to acknowledge that image-based detection may encounter reduced precision and recall rates under deteriorating lighting conditions, stemming from inherent hardware limitations within image sensors. Nevertheless, this limitation can be ameliorated through the application of various image processing techniques and technologies in subsequent phases.

4. Multi-Object Detection for Inland Ships

This section proposes the multi-object detection framework for inland ships based on the YOLOv5s architecture. The Shuffle Attention mechanism, transfer learning training strategy, and small object detection anchor are introduced to improve the method.

4.1. YOLOv5s Architecture

YOLOv5 adjusts the depth and width of its backbone network using two parameters: depth_Multiple and width_Multiple. YOLOv5s, the most miniature volume network model, is chosen as the basic architecture. We enhance the basic model architecture by adding the Shuffle Attention mechanism and optimizing the training process using a transfer learning strategy to train the network with limited ship image data effectively. The resulting detection model achieves improved accuracy and can be used for ship detection in various scenarios. The network structure of YOLOv5s consists of an input end, backbone, neck, and head detection module. The original network module of YOLOv5s includes CBS (Conv + BN + SiLU), BOTTLENECKCSP, CSP1_X, CSP2_X, SPPF, and other modules. For expression, CBS modules are merged into CSP1_X during visualization. In the YOLOv5s model, there are three detection layers. When the input image size is 640 × 640, the Neck network performs down sampling by 8×, 16×, and 32×, respectively. As a result, the dimensions of the feature maps in the corresponding Detect layers are 80 × 80, 40 × 40, and 20 × 20, respectively. These feature maps are used for detecting small, medium, and large targets, respectively.

4.2. Shuffle Attention Mechanism Network

In convolutional neural networks (CNNs), the attention mechanism operates on the feature map to capture the relevant attention within the feature map. In recent years, the attention mechanism has made significant breakthroughs in image and natural language processing and has been proven to be beneficial in improving model performance. There are mainly two categories of attention mechanisms in current research: namely, spatial attention mechanisms and channel attention mechanisms. The former focuses on capturing pixel-level relationships at spatial locations, locating the target, and performing spatial transformations or obtaining spatial weights, while the latter captures inter-channel dependencies. Shuffle Attention is a model that efficiently combines these two types of attention mechanisms. It first groups channel features to obtain sub-features from multiple groups and then applies the spatial and inter-channel attention mechanisms to each sub-feature using Shuffle Attention units. Finally, the features from different groups are fused using the Channel Shuffle operation. Adding an attention mechanism to the detection network can improve detection accuracy. The network structure of Shuffle Attention is illustrated in Figure 2.

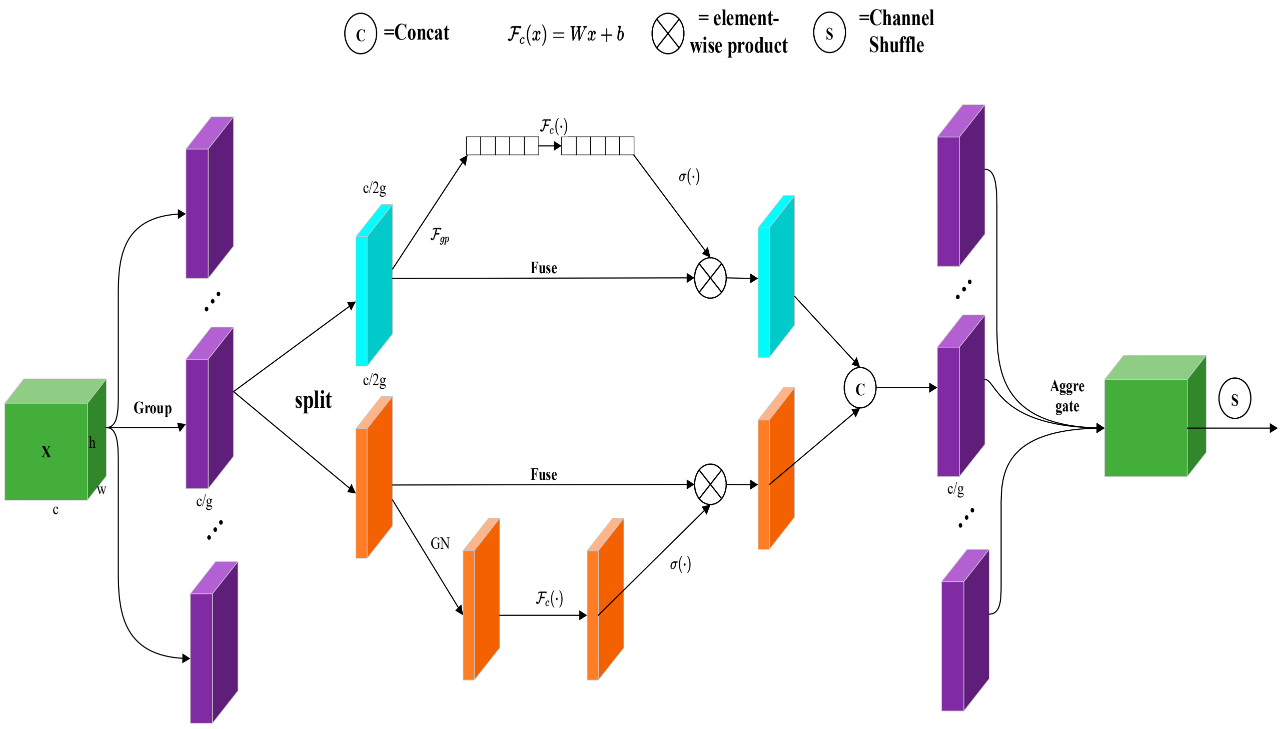


Figure 2. The Shuffle Attention Network.

The Shuffle Attention structure is in front of the SPPF module in the last layer of the BackBone, as well as the corresponding detection layers in the Head part of YOLOv5s. The Shuffle Attention structure helps the model better resist confusing information and focus on relevant target objects. Shuffle Attention structure is depicted in Figure 3.

Considering the potential risk of over-complexity in the network that may hinder the model's ability to fit the training data, a Yolov5s architecture with the Shuffle Attention only to the backbone part (ShuffleBackBone) is also designed, as depicted in Figure 4.

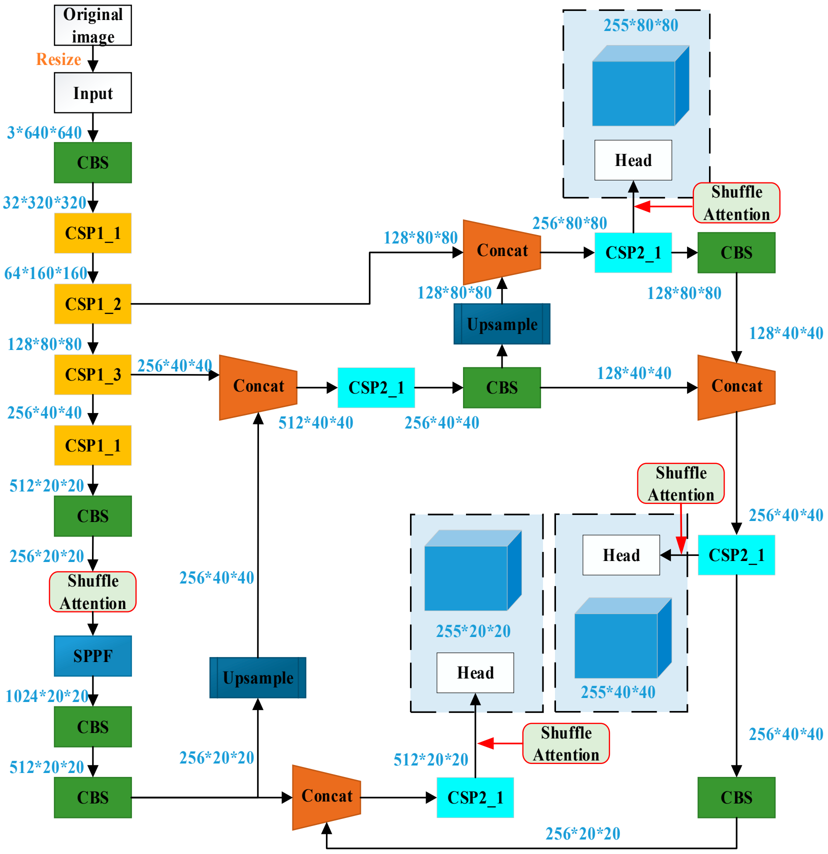


Figure 3. YOLOv5s Shuffle Attention.

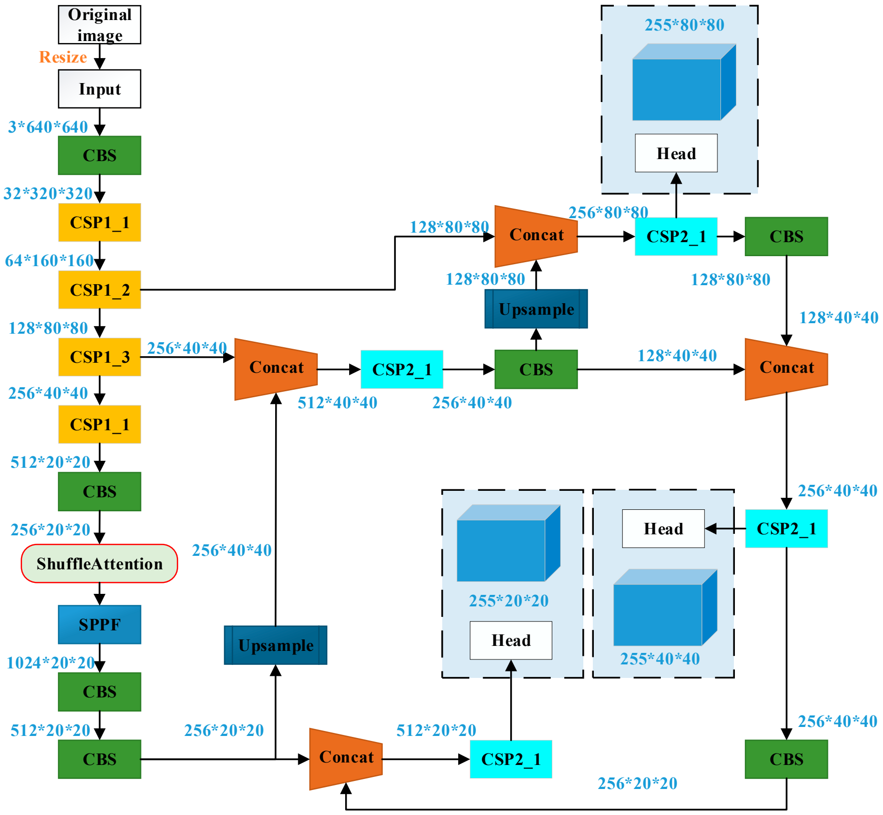


Figure 4. YOLOv5s Shuffle BackBone.

4.3. Transfer Learning Training Strategy

Transfer learning [33] is a machine learning technique that facilitates the transfer of knowledge from a source domain to a target domain, thereby improving the learning performance in the target domain. The quality of data annotation during training significantly impacts the training effectiveness. Feature transfer enhances the model's generalization performance, even when dealing with significantly large target datasets. In various studies related to transfer learning, DANN [34] introduces adversarial network concepts into the field, optimizing feature mapping parameters. In conventional training, a large amount of labeled data is typically required, and the manual data labeling process is labor intensive and costly. Due to the scarcity of ship image datasets with well-annotated information, ship detection training datasets often rely on ship SAR data, which are also limited in quantity.

There are two common approaches for applying transfer learning. The first approach is fine-tuning, which involves modifying a pre-trained network obtained from another source for the specific learning task. Usually, the weights of the network are initialized with pre-trained weights instead of random initialization. The second approach is using a fixed feature extractor, where the pre-trained network is used as a feature extractor for new tasks. Typically, the earlier layers of the network are frozen, and only the last fully connected layer is trained. Subsequently, the pre-trained network acts as a feature extractor, and the later layers are unfrozen to better learn the feature extraction parameters for the task at hand. The strategy employed in this paper involves both using a fixed feature extractor and adopting the fine-tuning approach.

This study chooses the UA-DETRAC high-speed dataset due to the similarity of target vehicle characteristics with ships. The high-speed dataset presents varying sizes of objects based on distance, which aligns with our target network. In the initial 30 epochs, all feature extraction layers (i.e., all network layers except the last Head layer) are frozen, and only the parameters in the Head prediction part are updated. This allows the network to first update the parameters of the ship detection module without affecting the previously trained feature extraction layers. The UA-DETRAC dataset is used to train YOLOv5s and obtain a pre-trained weight model. In the subsequent 100 epochs, all feature extraction layers are unfrozen to enable the YOLOv5s network to update all weight parameters. The transfer learning strategy employed in this paper allows for training a ship detection model with satisfactory accuracy and robustness, despite the limited availability of training samples in the target dataset.

4.4. Small Object Detection Anchor

In the ordinary YOLOv5 detection network, there are three rows of anchors parameters; each row has six values, and the values are (10, 13, 16, 30, 33, 23), (30, 61, 62, 45, 59, 119), (116, 90, 156, 198, 373, 326). Each row represents a different feature map of the application; among them, the sizes of the three different feature maps are 80 × 80, 40 × 40, and 20 × 20, respectively. In the object detection task, detecting small targets on a large feature map is generally desirable. Because the large feature map contains more small target information, the anchor value on the large feature map is usually set to a small value, while the value on the small feature map is set to a large value to detect large objects. Due to the large number of buoys in the ship dataset, the buoys are likely to occupy a very small proportion of the image in the distant images due to the different image sizes in the ship detection. The labeling of small target data is often inaccurate, and the overall detection effect is significantly lower than other types of ships. In this paper, a smaller anchor is selected in the detection network. Thus, the corresponding larger feature map can be obtained to detect smaller targets. We add a smaller anchor (5, 6, 7, 9, 12, 10), thus adding a feature map of 160 × 160 to improve the detection ability of buoys in object detection.

4.5. Implementation Details

Intersection over Union (IoU) is a widely used evaluation metric in object detection, which measures the spatial overlap between the predicted bounding box and the ground truth bounding box of an object. The IoU is computed by dividing the intersection area between the predicted and ground truth bounding boxes by the area of their union. A higher IoU value indicates a better localization accuracy of the predicted bounding box. Typically, a threshold value of 0.5 or 0.7 is used to determine whether the predicted bounding box is a true positive or a false positive. A new loss function called Alpha-IoU is proposed to replace the original Intersection over Union (IoU) loss. The Alpha-IoU loss consists of a Power IoU term and an additional Power regular term, both controlled by a single parameter, α. The experiments conducted in this paper demonstrate that using the Alpha-IoU loss can significantly outperform the current IoU loss.

The Power IoU term in the Alpha-IoU loss is designed to improve the accuracy of high IoU targets, which are typically objects with tight bounding box annotations. The Power IoU term helps the network to localize objects better with high precision. The Power regular term is an additional regularization term that encourages smooth regression of bounding box coordinates. Together, these two terms in the Alpha-IoU loss help improve the accuracy of gradient adaptive weighted box regression.

The choice of α, the power parameter in the Alpha-IoU loss, is important. This paper suggests that selecting $\alpha > 1$ can improve the loss of high IoU targets and the robustness of the network against noise, especially in small datasets. The α parameter can be adjusted to fine-tune the regression accuracy for different bounding box sizes, but it is not highly sensitive to different models and datasets. This paper recommends using $\alpha = 3$ as a parameter, as it performs well in the experiments.

Overall, the proposed Alpha-IoU loss with a selected α value of 3 effectively improves object detection accuracy, especially for small datasets and high IoU targets, providing a more robust and accurate loss function for training object detectors. The formula of Alpha-IoU is as follows:

$$\mathcal{L}_{\alpha-\mathrm{IoU}} = \frac{1 - \mathrm{IoU}^{\alpha}}{\alpha}, \alpha > 0 \tag{1}$$

According to the experiment and the characteristics of the relative loss weight and relative gradient weight of Alpha-IoU, Alpha-IoU can train a better detector than ordinary IoU. For the specific formula derivation, please refer to ref. [35].

Due to the insufficient sample size for some categories in our data, we have introduced sample weights to reduce the issue of low detection accuracy for a particular category caused by too few samples.

In other parts, Mosaic data enhancement, adaptive anchor box calculation, and adaptive image scaling used in YOLOv4 are also used. For other formula improvements and features, please refer to ref. [36].

5. Multi-Object Tracking for Inland Ships

Deep Sort [37] is an improvement based on Sort object tracking. The core of the Sort algorithm is Hungarian matching and Kalman filtering. In addition to the core part of the Sort module, DeepSORT also introduces a deep learning model ReID module. In the process of real-time object tracking, the appearance features of the target are extracted for nearest proximity matching. In the current frame, the minimum cosine distance between all Feature vectors of the *i*th object tracking and the detection of the *j*th object are calculated. The object tracking match can achieve a real-time tracking effect and reduce ID switching, which can be applied to industrial development. In this paper, to achieve a relatively continuous tracking effect for the input ship video data, we introduced the DeepSORT algorithm to combine it with the YOLOv5 model.

The general process of the DeepSORT algorithm is as follows: First, the prediction video or continuous image is input, and the confidence after prediction and position information obtained from our improved YOLOv5 detection network are taken as input. The Kalman filter first determines whether the track exists. If a track exists, the prior probability prediction is made for its position information. After that, the prior probability prediction obtained is cascaded matching and IoU matching. Finally, the matching list is obtained. The specific process of the algorithm is shown in Figure 5.

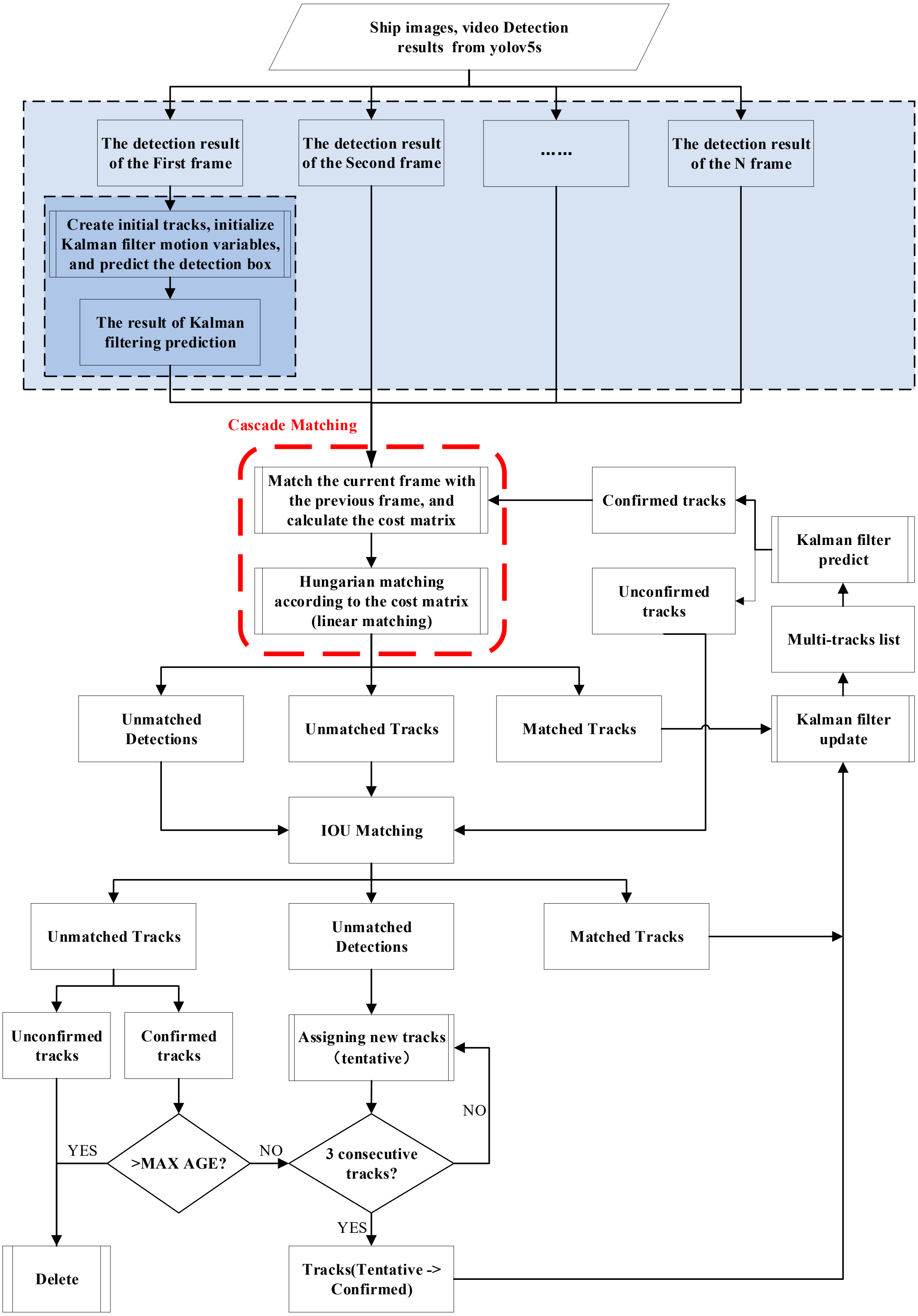

Ship images, video Detection results from yolov5s
The detection result of the First frame
The detection result of the Second frame
……
The detection result of the N frame
Create initial tracks, initialize Kalman filter motion variables, and predict the detection box
The result of Kalman filtering prediction
Cascade Matching
Match the current frame with the previous frame, and calculate the cost matrix
Hungarian matching according to the cost matrix (linear matching)
Confirmed tracks
Unconfirmed tracks
Kalman filter predict
Multi-tracks list
Kalman filter update
Unmatched Detections
Unmatched Tracks
Matched Tracks
IOU Matching
Unmatched Tracks
Unmatched Detections
Matched Tracks
Unconfirmed tracks
Confirmed tracks
Assigning new tracks (tentative)
NO
YES
>MAX AGE?
NO
3 consecutive tracks?
YES
Delete
Tracks(Tentative -> Confirmed)

Figure 5. DeepSORT framework.

The specific algorithm steps are divided into the following steps:

First, the original video frame is input into our YOLOv5 detection network, from which we can obtain the location of the detection result and its confidential information. The Kalman filter determines whether the track exists. It makes a prior probability prediction for its position information if it exists and abandons it if it does not exist.

Secondly, based on the mean value and variance of the previous moment, the forecast track predicts the mean value and variance information of the track at the next moment and updates the position and speed. The Kalman filter update formula is as follows:

$$\hat{\chi}_k^- = A\hat{\chi}_{k-1} + B\mu_k \tag{2}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{3}$$

where $\hat{\chi}_k^-$ represents the state at time k predicted from time $k-1$.

Thirdly, after the prior prediction is obtained, the detection that predicted the location of YOLOv5 and the predicted location of the track are matched. Use the cascade matching policy. The matching policy divides the tracks into Confirmed, Tentative, and Deleted types.

Cascade matching only matches the confirmed track. The cost matrix of cascade matching combines the cosine similarity distance and the Mahalanobis distance, and the measurement formula of the Mahalanobis distance matching is as follows:

$$d^{(1)}(i,j) = \left(d_j - y_i\right)^T S_i^{-1}\left(d_j - y_i\right) \tag{4}$$

where d_j represents the position of the jth detection box, y_i represents the predicted position of the target by the ith tracker, and S_i represents the covariance matrix between the detection position and the average tracking position.

The Mahalanobis distance takes the uncertainty of state measurement into account by calculating the standard deviation between the detection and the mean tracking positions. Because the Mahalanobis distance association method will be invalid when the camera moves, this will lead to the rapid switching of the ID of the tracking target. To make the ID switching stable, cosine distance matching is also introduced, which combines the appearance features of the object detection frame with the features extracted from the REID network. The formula is as follows:

$$d^{(2)}(i,j) = min\left\{1\middle|-r_j^T r_k^{(1)}\middle|r_k^{(i)} \in R_i\right\} \tag{5}$$

where R stands for appearance feature vector library and r stands for feature vector extracted for d detection blocks.

Finally, the two features are set a certain weight to obtain our final cost matrix:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda)d^{(2)}(i,j) \tag{6}$$

where λ stands for two metric weights.

The cost matrix of the determined track and corresponding detection is calculated by the Hungarian matching algorithm. The matching is successful if the cost matrix is smaller than the threshold. The matched track and observation value detection are obtained, and the unmatched track and detection are obtained.

Fourthly, combining the failed track with the tentative track and using the IoU matching strategy, IoU matching directly builds all track and detection from IoU as elements into the IoU cost matrix and uses the Hungarian algorithm for matching. The matching method is similar to that of cascade matching.

Then, after matching, the system obtains matched, unmatched_track, and unmatched_detection. It modifies the successfully matched track successively, updates the status of the failed track, converts the unmatched detection into a track, and updates the feature set of the successfully matched track.

Finally, the Kalman filter gain is used to correct and update the successfully matched track. The Kalman filter gain update formula is shown as follows:

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R} \tag{7}$$

$$P_k = (I - K_k C) P_k^- \tag{8}$$

$$\hat{\chi}_k = \hat{\chi}_k^- + K_k \left(y_k - C \hat{\chi}_k^- \right) \tag{9}$$

where C is the observation matrix, R represents the observation noise covariance matrix, and K_k is the Kalman coefficient.

The integration of the traditional Kalman filter (KF) into the DeepSORT framework for ship detection and tracking in our maritime context is driven by its core advantages, which include simplicity, effective state estimation, robust noise handling, and cost effectiveness. KF's simplicity and adaptability make it a practical choice for ship tracking in real-world scenarios where achieving reliable results without excessive resource expenditure is essential.

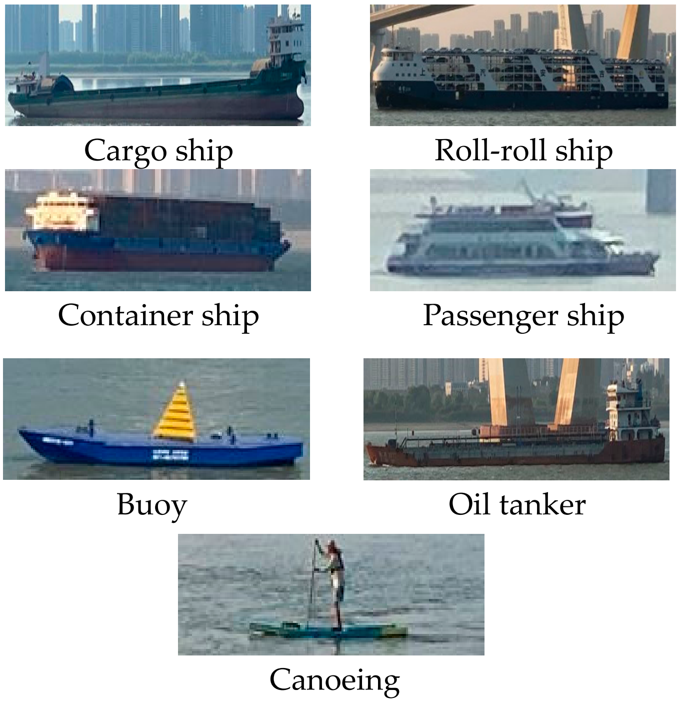
After updating the status of each track, deleting the dead track, and updating the feature set of the confirmed track, the process of this frame is finished, and the detection and tracking of the next frame start. In this way, the task of ship tracking with videos is completed.

6. Experimental Results and Discussion

In this section, we analyze the influence of three factors: the Shuffle Attention mechanism, the transfer learning strategy, and a smaller anchor box we added on the accuracy of ship object detection. The experimental environment is constructed on a laptop whose hardware configuration is as follows: AMD(R) Ryzen 7 4800h with Radeon graphics * 16 @ 2.9 GHz, NVIDIA Geforce RTX 3060 Laptop GPU/PCle/SSE2, 15.6 G running memory. The software configuration is as follows: Ubuntu 18.04.6 LTS, CUDA 11.1 version, cudnn 8.2.4 version. The project code is based on the PyTorch deep learning framework.

6.1. Setup

Ship image data along the Yangtze River are collected to make the supporting dataset. Each picture is labelled with corresponding ship categories. The training set is divided into seven categories: cargo ship, roll-roll ship, container ship, passenger ship, buoy, oil tanker, and canoeing. As the cargo ships are the majority on the Yangtze River, they occupy a relatively large proportion of the dataset, accounting for about 40% of the dataset. The remaining categories (buoy, container ship, passenger ship, and oil tanker) all account for about 50% of the dataset. The amount of roll-roll ship and canoeing in the dataset is small, accounting for only about 10 percent of the dataset. The seven categories are shown in Figure 6 below. In the following Sensitivity analysis experiment and other comparative experiments, we conducted five repeated experiments for each network that participated in the experiment and selected the best result among them.



Cargo ship

Roll-roll ship

Container ship

Passenger ship

Buoy

Oil tanker

Canoeing

Figure 6. Different types of ships in training datasets.

The Intersect Over Union (IoU) threshold parameter needs to be set for the model prediction. The Precision is the True positives (the correct ratio of all the recognized images). The recall is the ratio of correctly recognized objects to the total number of objects in the test set. The formulas are shown as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} \ = \frac{TP}{TP + FN} \tag{11}$$

where *TP* is the number of positive classes that are correctly predicted. *FP* is the number of negative classes that are predicted as positive classes, and *FN* is the number of positive classes that are predicted as negative classes.

AP means to average the precision values on the *PR* (precision and recall) curve. For the PR curve, we use the integral to compute it. The *AP* formula is shown below:

$$AP = \int_0^1 P(R)dR \tag{12}$$

In the experiment, when evaluating the detection accuracy of the algorithm, we used the *mAP* (Mean Average Precision) index commonly used in object detection to evaluate the model. mAP@0.5 is a *mAP* score obtained for detections corresponding to IoU > 0.5. The formula for *mAP* is as follows:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} \int_0^1 P(R)dR \tag{13}$$

where *n* represents the number of categories detected.

Cosine annealing is used to reduce the rate of learning by using cosine functions. In the cosine function, the cosine goes down slowly as x goes up, and then it goes down faster and goes down slowly again. This decline mode can cooperate with the learning rate and produce a good effect. The more the cosine annealing learning rate algorithm is trained, the smaller the learning rate will be. In this experiment, our initial learning rate (lr0) is set to 0.01, and our cosine annealing learning rate (lrf) is also set to 0.01.

6.2. Sensitivity Analysis of Object Detection Method

In this section, we conduct an ablation experiment to compare the effects of Shuffle Attention, transfer learning, and smaller anchor boxes on the network training strategy we have constructed. In this section, we use 585 images as the training set and 1614 images as the validation set.

6.2.1. Influence of Shuffle Attention

In object detection, we introduce the Shuffle Attention mechanism to improve the object detection of the network precision and recall. We train 130 epochs simultaneously in the original YOLOv5s network and networks with the ShuffleAttention mechanism and networks with ShuffleBackBone. And, we compare the results. The experimental results show that adding ShuffleBackBone and adding the ShuffleAttention mechanism can improve the overall mAP@0.5 value to a certain extent. The details are shown in Table 1. In the table footers, the use of bold text and underlining is employed to highlight the best-performing values within the experimental data.This formatting convention is consistently applied to subsequent tables (Tables 2–7) as well.

Table 1. Algorithm performance comparison of mechanism of attention (mAP@0.5 value).

Types	Cargo Ship	Roll-Roll Ship	Container Ship	Passenger Ship	Buoy	Oil Tanker	Canoeing	ALL
Original	0.776	0.647	0.814	0.671	0.638	0.757	0.836	0.734
Original + ShuffleBackBone	0.804	**0.678**	0.796	0.697	0.664	0.728	0.851	0.745
Original + ShuffleAttention	**0.81**	0.667	**0.821**	**0.74**	**0.685**	**0.808**	**0.877**	**0.773**

6.2.2. Influence of Transfer Learning

In ship object detection, the transfer learning method is introduced to use the UA-DETRAC high-speed vehicle dataset to reduce the demand for training data for object detection for inland ships. The high-speed dataset consists of 82,085 images with relatively good manual labeling information. The YOLOv5s network is trained using a large number of high-speed vehicle data with specific characteristics similar to the distant images of ships as pre-training weights. Obviously, training can be improved when using UA-DETRAC data pre-training data as training weights. We combine the transfer learning strategy data with ShuffleAttention and ShuffleBackBone, respectively, and compare the detection accuracy of the backbone without the pre-training weight. After using transfer learning, the network has a noticeable improvement in detection performance. The results are shown in Table 2.

In this study, we utilized a variety of datasets, including ImageNet, COCO, ABOships, Seaships, and UA-DETRAC. These datasets were partitioned into training and validation sets with a 7:3 ratio. The hyperparameters used for training were also kept consistent with those used in the previous experiments. To alleviate the impact of class imbalance, we also introduced the use of image weights during the training process. We conducted training on these datasets for 100 epochs, resulting in the generation of pre-trained weights. Subsequently, we employed these pre-trained weights to perform transfer learning on the ShuffleAttention network architecture proposed in this paper. The transfer learning was conducted on our dataset (consisting of 585 images for training and 1614 images for validation). Finally, we compared the detection performance in terms of the mAP@0.5 value obtained from this transfer learning approach against the baseline results. The results are shown in Table 3. In the comparative analysis, we evaluated the performance of the proposed method by conducting training experiments on our target dataset.

In the comparative analysis, we have found that both the COCO and ImageNet datasets exhibit favorable performance in ship detection. These datasets, enriched with numerous maritime-related data, demonstrate impressive results in detecting ships. However, despite their strong performance in specific scenarios, considering the overall mAP@0.5 value, we maintain that employing the UA-DETRAC dataset as the pretrained dataset is better suited for our specific task. On our target dataset, utilizing UA-DETRAC as the pretrained weight yields detection results that align more closely with our requirements.

Table 2. Algorithm performance comparison of transfer learning (mAP@0.5 value).

Types	Cargo Ship	Roll-Roll Ship	Container Ship	Passenger Ship	Buoy	Oil Tanker	Canoeing	ALL
Original	0.776	0.647	0.814	0.671	0.638	0.757	0.836	0.734
Original + transfer learning	0.871	0.737	**0.856**	**0.835**	0.792	0.801	0.73	0.803
ShuffleBackBone + transferlearning	0.873	0.721	0.854	0.82	0.783	0.865	0.863	0.826
ShuffleAttention + transferlearning	**0.885**	**0.753**	0.854	0.833	**0.794**	**0.869**	**0.957**	**0.849**

Table 3. Comparison of mAP values of different pretrained datasets and the proposed dataset (mAP@0.5 value).

Pretrained Datasets	Cargo Ship	Roll-Roll Ship	Container Ship	Passenger Ship	Buoy	Oil Tanker	Canoeing	ALL
ImageNet	**0.894**	0.738	0.886	0.841	0.767	0.849	0.651	0.803
COCO	0.885	**0.797**	**0.887**	**0.854**	0.778	0.836	0.77	0.829
ABOships	0.85	0.717	0.831	0.773	0.761	0.793	0.745	0.781
Seaships	0.884	0.712	0.864	0.796	0.744	0.825	0.697	0.789
UA-DETRAC	0.885	0.753	0.854	0.833	**0.794**	**0.869**	**0.957**	**0.849**

6.2.3. Influence of Smaller Anchor Box

In object detection, we introduce a smaller anchor box, which uses the feature map of 160 × 160 to improve the detection of small objects. The data reflect the improvement of the detection accuracy of the buoy. However, adding the detection box for small targets may decrease the detection accuracy and recall rate of large ship targets. We conduct experiments on the original yolo network, the network adding a small object detection layer, the network adding ShuffleBackBone and the small object detection network attention mechanism, and the network adding ShuffleAttention and small object detection attention mechanism, respectively. The results are shown in Table 4.

The detection capability of the network for the small target buoy is improved to a certain extent. However, the accuracy of other categories and the overall detection network is decreased. When small object detection (+1 anchor) is added to the strategy (ShuffleBackBone + transferlearning) that we think can obtain the best detection result, the overall mAP@0.5 value will decrease. However, the detection rate of the buoy increased. Thus, a network without small object detection would work better. However, for specific small object detection tasks, such as buoy detection, we can try adding small object detection into the network to improve detection accuracy.

Table 4. Algorithm performance comparison of smaller anchor box (mAP@0.5 value).

Types	Cargo Ship	Roll-Roll Ship	Container Ship	Passenger Ship	Buoy	Oil Tanker	Canoeing	ALL
Original	0.776	0.647	**0.814**	0.671	0.638	0.757	**0.836**	0.734
Original + 1anchor	0.719	0.63	0.683	0.614	0.687	0.566	0.79	0.67
Shuffleonlybackbone + 1anchor	0.737	0.561	0.694	0.609	0.692	0.619	0.796	0.672
Shuffleattention + 1anchor	0.72	0.611	0.657	0.589	0.721	0.584	0.809	0.67
ShuffleBackBone + transferlearning + 1anchor	0.811	0.718	0.795	0.776	0.813	0.768	0.808	0.784
ShuffleAttention + transferlearning + 1anchor	**0.827**	**0.739**	0.813	**0.785**	**0.813**	**0.78**	0.818	**0.796**

6.3. Setup

To validate the superior performance of the object detection network, we conduct a series of experiments on our dataset. We introduce other state-of-the-art (SOTA) object detection models and train and validate them on our dataset. These competing methods use the same training and validation sets and are trained for 130 epochs with original pretrain weights. We compare the method (ShuffleAttention + transferlearning) with other competing methods on our test datasets (145 images). The specific comparative data are shown in Table 5. The proposed framework shows the best detection performance for all types of objects except the oil tanker. Still, for oil takers, the mAP@0.5 value of the proposed framework is the second. It is worth noting that we have also included two additional metrics, FPS and params, to evaluate our model's performance. FPS refers to the number of frames per second that the model can process during image or video inference (object detection and localization); it can be influenced by various factors, such as the resolution of input images, model complexity, and the availability of computational resources. In this section, FPS calculations for the experiments were consistently conducted on the NVIDIA GeForce RTX 3090, with validation performed on our test datasets consisting of 145 images. The size of params indicates the model's parameter count, which is influenced by both the network architecture and the number of classes in the training dataset (nc). The size of 'params' reflects the complexity of the model. The results are shown in Table 6. The results indicate that our proposed framework achieves the highest level of accuracy on the test dataset while maintaining faster detection speeds.

Table 5. Algorithm performance comparison with SOTA methods (mAP@0.5 value).

Pretrained Datasets	Cargo Ship	Roll-Roll Ship	Container Ship	Passenger Ship	Buoy	Oil Tanker	Canoeing	ALL
Faster-RCNN [3]	0.73	0.749	0.813	0.702	0.28	0.861	0.683	0.688
Yolov3 [38]	0.636	0.4	0.673	0.513	0.496	0.638	0.226	0.51
Yolov4-tiny [39]	0.76	0.705	0.836	0.835	0.431	0.796	0.538	0.70
Yolov4 [40]	0.787	0.833	0.845	0.838	0.639	**0.952**	0.565	0.778
Yolov5s [41]	0.809	0.737	0.821	0.718	0.675	0.753	0.726	0.749
Yolov7 [42]	**0.914**	0.575	**0.923**	0.803	0.818	**0.894**	0.613	0.791
Yolov8s [41]	0.869	0.688	0.849	0.782	0.724	0.827	0.658	0.771
Proposed framework	0.866	**0.995**	0.851	**0.925**	**0.819**	0.878	**0.995**	**0.904**

Table 6. Algorithm performance comparison with FPS and Params.

Pretrained Datasets	FPS	Params
Faster-RCNN [1]	3.75	136,811,934
Yolov3 [38]	82.6	61,529,740
Yolov4-tiny [39]	409.22	5,887,976
Yolov4 [40]	66.42	6,042,3500
Yolov5s [41]	144.9	7,029,004
Yolov7 [42]	120.5	37,227,020
Yolov8s [41]	417.7	11,138,309
Proposed framework	128.2	7,029,532

6.4. Comparison with Other Papers

This section compares the proposed multi-object detection and tracking framework with popular multi-type object ship detection papers. The key index is the amount and type of data they use with the value of mAP@0.5.

Paper [1], using YOLOv4-tiny and YOLOv4-tiny-3l for ship object detection, xxx in yolov4-tiny-xxx or yolov4-tiny-3l-xxx in the table below, represents the input image size. The author uses the SeaShips dataset [43] and the ABOships dataset [44]. The two datasets are divided into 70 percent training and 30 percent validation sets. The SeaShips dataset contains 31,455 images taken from video segments acquired from coastline video surveillance systems. The ABOships dataset contains 9,880 images. ABOships dataset images, on the other hand, were obtained through a camera mounted on a ferry, providing footage from the point of view of the vehicle and providing annotations for nine types of vessels, seamarks, and miscellaneous floaters. ABOships datasets contain many small objects, similar to the small buoys in our dataset. Too many small objects will cause the mAP@0.5 value of the overall dataset to decrease significantly. In ref. [45], an enhanced CNN-enabled learning method was proposed. Combined with yolov3, the accuracy of ship multi-type object detection is improved. The dataset used in this article is 7000 from SeaShips.

The original YOLOv5s network and our proposed framework (transfer learning + improved YOLOv5s network) were employed to train and evaluate their performance on the SeaShips (7000) and ABOships (9880) datasets, essential datasets in the maritime domain. To ensure a comprehensive evaluation, we adopted two distinct data partitioning strategies for training. The first approach adhered to the standard convention used in related works, where the ABOships and SeaShips datasets were partitioned into a 70% training set and a 30% validation set. Subsequently, we employed the second approach to investigate the efficacy of transfer learning with a limited training set. Specifically, we utilized a mere 10% of the data from each dataset for training, reserving the remaining 90% for validation purposes. This approach facilitated a robust comparison between the results of our proposed framework and other existing approaches cited in the literature. Through meticulous analysis and comparison of the final mAP@0.5 values, our proposed method, which integrates transfer learning with our improved YOLOv5s-based network, demonstrated exceptional proficiency in effectively training on both the ABOships and SeaShips datasets. Notably, our proposed framework achieved high levels of accuracy and recall on the validation sets. These results not only showcase the superiority of our proposed framework but also highlight its capability to address the challenges presented by these specific datasets.

By incorporating the two metrics, FPS and params, introduced in Table 6 of Section 6.3, these metrics have also been included in the table below for our data comparison. From the comparative data, it can be observed that our model achieves a high detection mAP@0.5 while simultaneously demonstrating faster detection speeds and smaller model parameters.

The results are shown in Table 7.

Table 7. Performance comparison with other papers.

Model	Datasets				mAP@0.5 (Best) (%)	FPS	Params
	SeaShips (31,455 Images)	ABO Ships (9880 Images)	Sea Ships (7000 Images)	Ours (585 Training Images)			
Yolov4-tiny-352	☑	☑			84.37 (SeaShips) 35.24 (ABOships)	20.6 (Darknet) 30.5 (TensorRT)	5,882,562 (SeaShips) 5,894,112 (ABOships)
Yolov4-tiny-416	☑	☑			82.63 (SeaShips) 38.33 (ABOships)	16.1 (Darknet) 24.8 (TensorRT)	
Yolov4-tiny-480	☑	☑			85.63 (SeaShips) 40.02 (ABOships)	13.8 (Darknet) 21.6 (TensorRT)	
Yolov4-tiny-544	☑	☑			85.31 (SeaShips) 42.07 (ABOships)	9.3 (Darknet) 15.7 (TensorRT)	
Yolov4-tiny-608	☑	☑			84.63 (SeaShips) 42.89 (ABOships)	8.3 (Darknet) 14.3 (TensorRT)	
Yolov4-tiny-3l-352	☑	☑			84.06 (SeaShips) 37.39 (ABOships)	18.7 (Darknet) 27.9 (TensorRT)	6,124,579 (SeaShips) 6,138,064 (ABOships)
Yolov4-tiny-3l-416	☑	☑			85.21 (SeaShips) 40.67 (ABOships)	14.4 (Darknet) 22.9 (TensorRT)	
Yolov4-tiny-3l-480	☑	☑			83.98 (SeaShips) 41.54 (ABOships)	12.3 (Darknet) 19.3 (TensorRT)	
Yolov4-tiny-3l-544	☑	☑			84.58 (SeaShips) 41.30 (ABOships)	8.4 (Darknet) 14.5 (TensorRT)	
Yolov4-tiny-3l-608	☑	☑			83.45 (SeaShips) 42.88 (ABOships)	7.5 (Darknet) 12.9 (TensorRT)	
eYOLOv3-416			☑		85.62	35 (Nvidia GeForceGTX 1080TI GPU)	61,524,355 (SeaShips) 61,551,280 (ABOships)
eYOLOv3-512			☑		87.28	22 (Nvidia GeForceGTX 1080TI GPU)	
eYOLOv3-608			☑		87.74	30 (Nvidia GeForceGTX 1080TI GPU)	
YOLOv5s		☑			84.2 (SeaShips-%10 training)	142.8 (NVIDIA GeForce RTX 3090)	7,026,307 (SeaShips) 7,039,792 (ABOships) 7,029,004 (Ours)
					98.40 (SeaShips-%70 training)	149.2 (NVIDIA GeForce RTX 3090)	
			☑		40.1 (ABOships-%10 training)	142.9 (NVIDIA GeForce RTX 3090)	
					62.2 (ABOships-%70 training)	147.1 (NVIDIA GeForce RTX 3090)	
				☑	74.9	144.9 (NVIDIA GeForce RTX 3090)	

Table 7. *Cont.*

Model	**Datasets**				**mAP@0.5 (Best) (%)**	**FPS**	**Params**
	SeaShips (31,455 Images)	**ABO Ships (9880 Images)**	**Sea Ships (7000 Images)**	**Ours (585 Training Images)**			
Proposed framework		☑			**<u>92.90</u>** (SeaShips-%10 training)	153.8 (NVIDIA GeForce RTX 3090)	7,026,835 (SeaShips) 7,040,320 (ABOships) 7,029,532 (Ours)
					<u>98.90</u> (SeaShips-%70 training)	163.9 (NVIDIA GeForce RTX 3090)	
			☑		**<u>45.10</u>** (ABOships-%10 training)	149.3 (NVIDIA GeForce RTX 3090)	
					<u>63.40</u> (ABOships-%70 training)	144.9 (NVIDIA GeForce RTX 3090)	
				☑	**<u>86.4</u>**	138.9 (NVIDIA GeForce RTX 3090)	

6.5. Experiments on Multi-Object Detection and Tracking for Inland Ships

In the following experiment, we combined the object tracking algorithm DeepSORT to carry out the test and analyzed the effectiveness of the proposed framework under good visibility and restricted visibility. As cargo ships have the largest proportion among the ships sailing on the Yangtze River, we consider cargo detection accuracy and recall rate to be the most important indicators. Therefore, while ensuring a certain level of detection accuracy for other categories, we ultimately chose the ShuffleAttention + transfer learning strategy model with the highest detection accuracy for cargo ships for our multi-object detection and tracking experiments.

6.5.1. Object Tracking in Good Visibility

We set up our detection and tracking model to track in the daytime with good visibility. As shown in Figure 7, the proposed model can accurately track the target cargo ship and buoy. Figure 8 shows that the model automatically deletes the tracking track when the ship is out of the detection range. In general, after testing with 144 images in good visibility, the mAP@0.5 of the proposed method is 0.904.

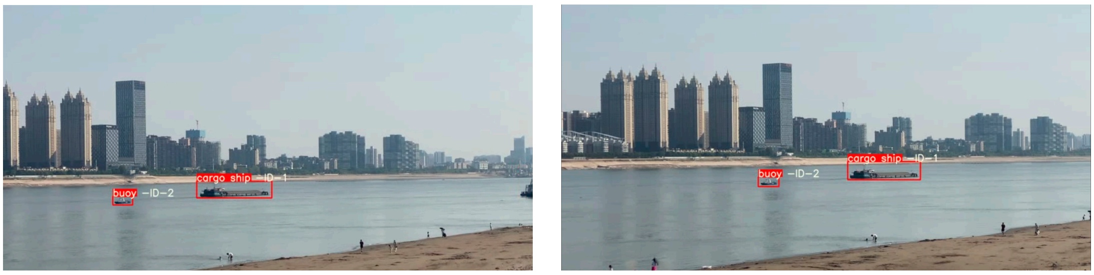


Figure 7. Cargo ship and buoy tracking in good visibility.

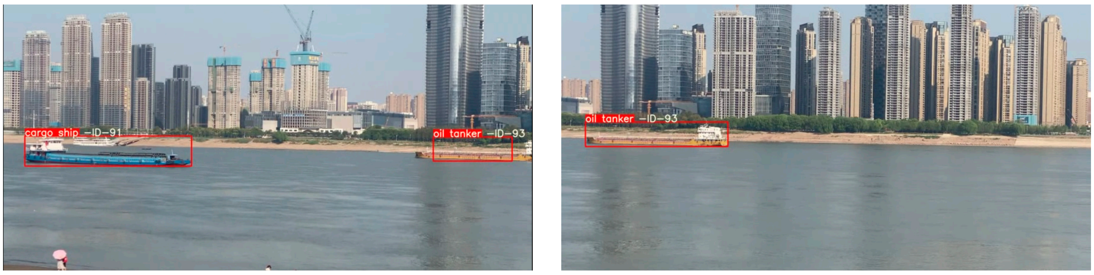


Figure 8. Cargo ship and oil tanker tracking in good visibility.

6.5.2. Object Tracking in Restricted Visibility

The video of the ship collected on rainy and foggy days with poor visibility is used to test the performance of the proposed framework in restricted visibility. As shown in Figure 9, the cargo ship was tracked steadily.

However, the ship image is blurred due to the rain and fog. Thus, some problems exist, such as undetected targets and wrong target identification. As shown in Figure 10, the ship can complete the tracking and detection in rainy and foggy days, but the oil tanker in the figure below is detected as a container ship. In Figure 11, the object tracking in the near distance is achieved, but the ship in the far distance is missing (the buoy tracking loss). Still, after testing with 150 images in poor visibility, the mAP@0.5 is 0.746.

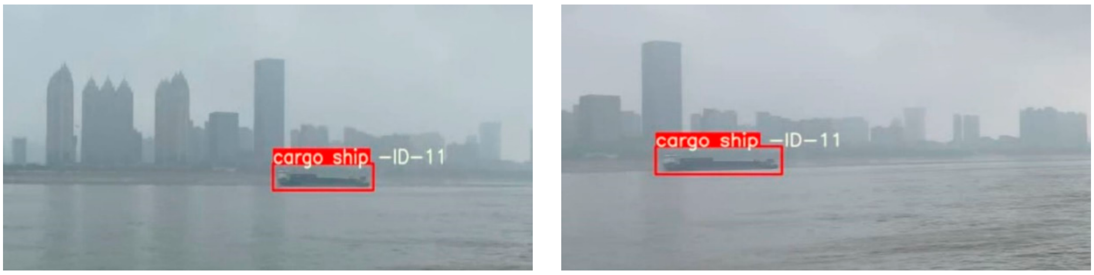


Figure 9. Cargo ship tracking in poor visibility.

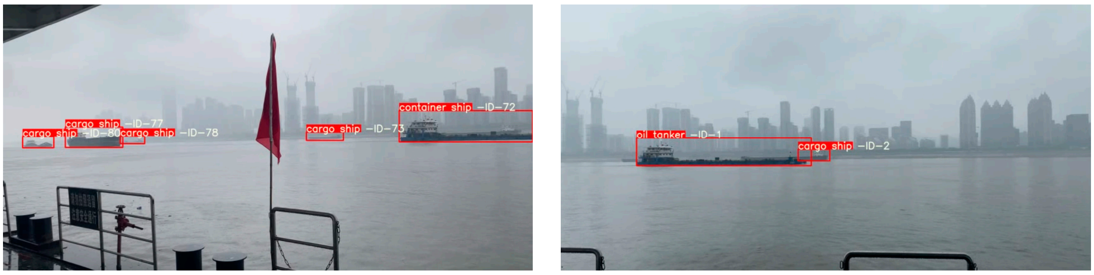


Figure 10. Cargo ship and oil tanker tracking in poor visibility.

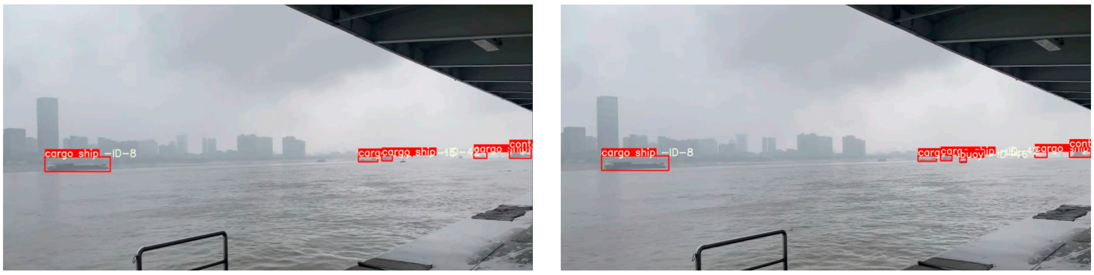


Figure 11. Cargo ship and buoy tracking in poor visibility.

7. Conclusions

In this paper, we proposed a transfer-learning-based object detection and tracking method for small sample datasets. The YOLOv5 network is improved by introducing the Shuffle Attention mechanism and smaller anchors. Using a transfer learning strategy, the UA-DETRAC high-speed dataset is then introduced to train a pre-trained model. A small amount of self-collected Yangtze River vessel dataset is used to fine-tune the model, combined with the tracking algorithm DeepSORT for vessel detection and tracking in the case of limited data. Finally, the feasibility of our method is demonstrated through extensive experiments. The main contributions of this article are as follows: (1) improvements and innovations have been made to the YOLOv5 network, specifically for ship detection; (2) implementing a tracking monitoring method that only requires a small amount of ship image data; and (3) proposing an improved YOLOv5 + DeepSORT object detection and tracking network and introducing a transfer learning strategy to enhance detection and tracking accuracy. The proposed method is characterized by fast training speed and high accuracy with small datasets. Compared with existing methods, the proposed method achieved 84.9% (mAP@0.5) with only 585 training images.

Future research directions are considered to improve the method. In terms of ship detection, the accuracy of the ship detection model may be significantly reduced due to extreme weather conditions. In the future, fusion with LiDAR can be added, as LiDAR can provide more geometric and visual information to improve detection accuracy. Regarding ship tracking, rapid ID switching may occur after ship crossing.

Author Contributions: Conceptualization, J.W.; methodology, J.W. and M.G.; investigation, J.W. and M.L.; writing—original draft preparation, J.W.; writing—review and editing, J.W. and M.L.; supervision, J.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the National Key R&D Program of China under grant No. 2023YFE0203700, the National Natural Science Foundation of China (NSFC) under grant No. 52271367, and the Fundamental Research Funds for the Central Universities under grant WUT: 2023IVA100.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions of privacy.

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

References

1. Nunes, D.; Fortuna, J.; Damas, B.; Ventura, R. Real-time Vision Based Obstacle Detection in Maritime Environments. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Santa Maria da Feira, Portugal, 29–30 April 2022; pp. 243–248. [CrossRef]
2. Hong, X.; Cui, B.; Chen, W.; Rao, Y.; Chen, Y. Research on Multi-Ship Target Detection and Tracking Method Based on Camera in Complex Scenes. *J. Mar. Sci. Eng.* **2022**, *10*, 978. [CrossRef]
3. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
4. Nie, S.; Jiang, Z.; Zhang, H.; Cai, B.; Yao, Y. Inshore ship detection based on mask R-CNN. In Proceedings of the IGARSS 2018—IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 693–696. [CrossRef]
5. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef] [PubMed]
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multibox detector. In *Computer Vision–ECCV 2016: Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]

8. Zhao, K.; Zhou, Y.; Chen, X. A dense connection based SAR ship detection network. In Proceedings of the IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; pp. 669–673. [CrossRef]
9. Zhou, Z.; Guan, R.; Cui, Z.; Cao, Z.; Pi, Y.; Yang, J. Scale expansion pyramid network for cross-scale object detection in sar images. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 5291–5294. [CrossRef]
10. Wang, B.; Han, B.; Yang, L. Accurate real-time ship target detection using Yolov4. In Proceedings of the 6th International Conference on Transportation Information and Safety (ICTIS), Wuhan, China, 22–24 October 2021; pp. 222–227. [CrossRef]
11. Zou, Y.; Zhao, L.; Qin, S.; Pan, M.; Li, Z. Ship target detection and identification based on SSD_MobilenetV2. In Proceedings of the IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1676–1680. [CrossRef]
12. Liu, M.; Zhu, C. Residual YOLOX-based Ship Object Detection Method. In Proceedings of the 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 14–16 January 2022; pp. 427–431. [CrossRef]
13. Hou, X.; Zhang, F. The Improved CenterNet for Ship Detection in Scale-Varying Images. In Proceedings of the 3rd International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 8–11 November 2021; pp. 1–5. [CrossRef]
14. Grabner, H.; Leistner, C.; Bischof, H. Semi-supervised On-Line Boosting for Robust Tracking. In *Computer Vision—ECCV 2008, Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 234–247.
15. Nair, V.; Clark, J.J. An Unsupervised, Online Learning Framework for Moving Object Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004; p. II-II. [CrossRef]
16. Babenko, B.; Yang, M.H.; Belongie, S. Robust Object Tracking with Online Multiple Instance Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1619–1632. [CrossRef] [PubMed]
17. Zhou, X.; Yang, C.; Yu, W. Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 597–610. [CrossRef]
18. Mittal, A.; Paragios, N. Motion-Based Background Subtraction using Adaptive Kernel Density Estimation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004; Volume 2, p. II-302.
19. Ablavsky, V. Background models for tracking objects in water. In Proceedings of the International Conference on Image Processing (Cat. No. 03CH37429), Barcelona, Spain, 14–17 September 2003; p. III-125. [CrossRef]
20. Zhang, S.; Qi, Z.; Zhang, D. Ship tracking using background subtraction and inter-frame correlation. In Proceedings of the 2nd International Congress on Image and Signal Processing, Tianjin, China, 17–19 October 2009; pp. 1–4. [CrossRef]
21. Deng, C.; Cao, Z.-G.; Zhiwen, F.; Yu, Z. Ship detection from optical satellite image using optical flow and saliency. In Proceedings of the 8th International Symposium on Multispectral Image Processing and Pattern Recognition, Wuhan, China, 26–27 October 2013; Volume 8921, p. 89210F.
22. Kaido, N.; Yamamoto, S.; Hashimoto, T. Examination of automatic detection and tracking of ships on camera image in marine environment. In Proceedings of the Techno-Ocean (Techno-Ocean), Kobe, Japan, 6–8 October 2016; pp. 58–63. [CrossRef]
23. Zechuang, C.; Bin, L.; Lian Fang, T.; Dong, C. Automatic detection and tracking of ship based on mean shift in corrected video sequences. In Proceedings of the 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, 2–4 June 2017; pp. 449–453. [CrossRef]
24. Fang, K.; Xiang, Y.; Li, X.; Savarese, S. Recurrent autoregressive networks for online multi-object tracking. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 466–475. [CrossRef]
25. Zhu, J.; Yang, H.; Liu, N.; Kim, M.; Zhang, W.; Yang, M.-H. Online multi-object tracking with dual matching attention networks. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; pp. 379–396.
26. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. MOTS: Multi-Object Tracking and Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7934–7943. [CrossRef]
27. Chu, Q.; Ouyang, W.; Li, H.; Wang, X.; Liu, B.; Yu, N. Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4846–4855. [CrossRef]
28. Milan, A.; Rezatofighi, H.; Dick, A.; Reid, I. Online Multi-Target Tracking Using Recurrent Neural Networks. *Proc. AAAI Conf. Artif. Intell.* **2016**, *31*, 4225–4232. [CrossRef]
29. Tang, S.; Andres, B.; Andriluka, M.; Schiele, B. Multi-person tracking by multicut and deep matching. In Proceeding of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 October and 15–16 October 2016; pp. 100–111.
30. Zhang, W.; Zhou, H.; Sun, S.; Wang, Z.; Shi, J.; Loy, C.C. Robust multi-modality multi-object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2365–2374.
31. Zhang, W.; He, X.; Li, W.; Zhang, Z.; Luo, Y.; Su, L.; Wang, P. A Robust Deep Affinity Network for Multiple Ship Tracking. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–20. [CrossRef]

32. Meng, Z.; Xia, X.; Xu, R.; Liu, W.; Ma, J. HYDRO-3D: Hybrid Object Detection and Tracking for Cooperative Perception Using 3D LiDAR. *IEEE Trans. Intell. Veh.* **2023**, 1–13. [CrossRef]
33. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *2*, 3320–3328.
34. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
35. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.-S. $\alpha $-IoU: A family of power intersection over union losses for bounding box regression. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20230–20242.
36. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
37. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
38. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
39. Jiang, Z.; Zhao, L.; Li, S.; Jia, Y. Real-time object detection method based on improved YOLOv4-tiny. *arXiv* **2020**, arXiv:2011.04244.
40. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
41. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
42. 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, 17–24 June 2023; pp. 7464–7475.
43. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. SeaShips: A Large-Scale Precisely Annotated Dataset for Ship Detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [CrossRef]
44. Iancu, B.; Soloviev, V.; Zelioli, L.; Lilius, J. ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sens.* **2021**, *13*, 988. [CrossRef]
45. Liu, R.W.; Yuan, W.; Chen, X.; Lu, Y. An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system. *Ocean Eng.* **2021**, *235*, 109435. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.