Journal of Marine Science and Engineering

MDPI

Article

Online estimation of ship dimensions by combining images with AIS reports

Zishuo Huang [1], Qinyou Hu [1,*], Lan Lu [2], Qiang Mei [3] and Chun Yang [1]

[1] Merchant Marine College, Shanghai Maritime University, Shanghai, China; 202040110005@stu.shmtu.edu.cn (Z. H.); chunyang@shmtu.edu.cn (C. Y.)
[2] Hainan Research Institute, Nankai University, Tianjin, China; nklulan@163.com (L. L.)
[3] Navigation College, Jimei University, Xiamen, China; meiqiang@jmu.edu.cn (Q. M.)
* Correspondence: qyhu@shmtu.edu.cn; Tel.: +86-21-38282915

Abstract: Ship dimensions are an important component of static AIS information and are a key factor in identifying the risks of ship collision. We describe a method of extracting and correcting ship contours using inland waterway surveillance video combined with AIS information that does not depend on ship dimension data. A lightweight object detection model was used to determine the ship's position in an image. Dynamic AIS information was included to produce multigroup control points, solve the optimal homography matrix, and create a transformation model to map image coordinates onto water surface coordinates. A semantic segmentation DeepLabV3+ model was used to determine ship contours from images, and the actual dimensions of the ship contours were calculated by homography matrix transformation. The mAP of the proposed object detection model and the MIoU of the semantic segmentation model were 86.73% and 91.07% respectively. The calculation error of ship length and width were 5.8% and 7.4% respectively. These statistics indicate that the proposed method rapidly and accurately detected target ships in images and that the model estimated ship dimensions within a reasonable range.

Keywords: contour extraction; object detection; semantic segmentation; coordinate mapping

1. Introduction

In water traffic scenarios, ship collision avoidance algorithms are often unable to avoid collisions between ship and pier or other ships because they only use the position of AIS (automatic identification system) report without considering the shape of the ship. The AIS position is determined by the position of the VHF antenna, and the distance between the AIS antenna and the ship periphery can range from tens to hundreds of meters. For ship-pier collision, it is necessary to consider the transverse distribution of ships in the river, and use the ship width data to calculate the collision probability. In the process of calculating the collision risk index (CRI) between two ships, an AIS-equipped ship is often shown as a triangle or rectangle with the transponder at the center [1]. There are dimension attributes in AIS static information represent the length and width of the ship, but in practice, these AIS data are often unavailable. Table 1 shows the dimensions and numbers of ships passing through two inland waterway channels in China within a given time period. It is clear from the table that ship dimension information is unavailable for more than 60% of the vessels. Given the possible consequences of ship collisions, the problem of unavailable ship size data needs urgent resolution.

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Last-name

Received: date
Revised: date
Accepted: date
Published: date

Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Table 1. Available information for ship dimensions.

	Bulk carrier			Tanker		
	No. observed	No. with available information	Ratio	No. observed	No. with available information	Ratio
Jingzhou waterway	100	38	0.38	50	19	0.38
Yichang waterway	100	33	0.33	50	26	0.52

Images have become more widely used for information extraction by computer vision technologies. When coupled with AIS and advanced image technologies, the situational awareness of an autonomous surface vehicle (ASV) can be increased to ensure safe navigation [2]. Such use will enable vessels to conform to navigation regulations in key navigable waters [3]. Combining visual data with AIS information enables the estimation of the size of specific ships in the image. Remote sensing or visible light images have frequently been used to extract the contours of target ships, create matching external rectangles or ellipses based on the contour shapes, and derive longitudinal and transverse ship dimension information [4]. Therefore, contour extraction is the basis of ship size estimation, which has been widely used in the transportation sector.

There have been many studies of ship contour extraction. In conventional contour extraction methods, edge detection based on image characteristics has been used to determine contours. Yan et al. [5] improved the Canny edge detection algorithm by using a two-dimensional wavelet Gaussian function to calculate the partial derivative of structural filter gradient amplitude and adopting maximum inhibition and threshold filters for edge detection and connection for ships. Gu et al. [6] used binary image gradient calculation for edge detection and determined the minimum enclosing rectangle for ship contours. Zhu et al. [7] demonstrated a ship recognition method that used a predicted shape template to determine ship contours using the Otsu method with peak density detection and column scanning as well as a conventional area averaging algorithm. Nie et al. [8] used a binarized normed gradient (BING) algorithm to predict the location of a ship in SAR images and used an active contour algorithm to predict ship contours iteratively. Standard ship contour extraction methods are simple but are often unable to extract deep image information and are only suitable for simple scenes.

Convolutional neural networks (CNN) are widely used for image feature extraction in deep learning applications in networks such as VGGNet [9], GoogleNet [10], Inception [11] and ResNet [12]. CNNs have important applications in semantic segmentation; multicategory target contours can be accurately segmented by pixel-level classification of images, and they have been used in many ship contour extraction applications. FCN was commonly used to extract ship contours by categorizing each pixel in a remote sensing image into bow, hull, land, and sea [13]. Bovcon et al. [14] developed a deep encoder-decoder framework (the water obstacle separation and refinement network) for autonomous crewless ship navigation that extracted the contours of several ship targets. Ust et al. [15] introduced a scaffolding learning regime (SLR) that trained an obstacle detection segmentation network under weak supervision for individual ship contour extraction. Kelm et al. [16] trained a CNN to identify central pixels; the network recognized a part of an input image and calculated a rotation angle, and used the central pixel to describe the upcoming directional change in the contour. Deep learning methods that rely on training data are more accurate than other established methods and are highly adaptable to different scenarios.

Remote sensing data of a particular area are not frequently updated, and vessels are densely distributed on inland waterways, making it difficult to accurately extract contours from remote sensing images at any given point in time. Visible light images are generally made from a horizontal perspective, and this perspective is not particularly suitable for accurate ship size estimation. In this study, we propose a method for ship contour extraction and coordinate mapping from an overhead perspective using video waterway surveillance data from bridges. The research scenario is shown in Figure 1. The bridges used were on important stretches of the upper reaches of the Yangtze River that have high marine traffic flow and density. The main methods used were the following.

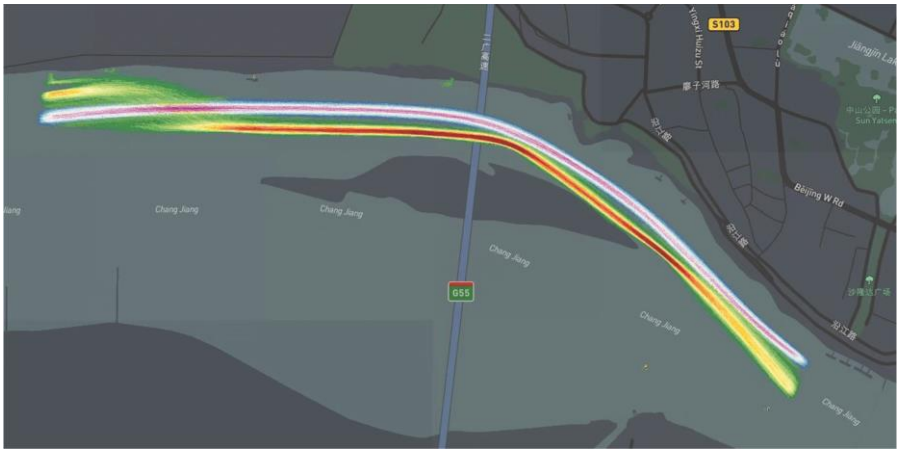


Figure 1. The traffic flow of Jingzhou Bridge in one month.

- A deep learning ship object detection model was developed, based on a lightweight object detection model and using the SENet attention mechanism to improve the network structure and increase the effectiveness of detection;
- An optimal homography matrix solution algorithm using several AIS control points was developed to determine the mapping relationship between image coordinates and water surface coordinates;
- Ship contours were extracted using the deep learning DeepLabV3+ semantic segmentation model in conjunction with the homology matrix transformations to determine the real size of the vessel.

2. Related Studies

2.1. Ship object detection

Ship object detection technology has a research history of more than twenty years and forms the basis for combining video and AIS information in our study [17]. Established methods include those based on the water–sky boundary, saliency detection, and moving object detection. Kim et al. [18] used a background algorithm to detect ships and combined it with AIS to match ships with ship-related information. Fefilatyev et al. [19] developed a method using optimal water–sky boundary extraction combined with Gaussian distribution and the Hough transform. Yang et al. [20] designed a ship motion tracking system based on FPGA that differed from traditional inter-frame difference methods, which had fixed frame intervals. Its inter-frame difference method was based on the adaptive extraction of key frames and was used to adaptively detect ships moving at different speeds.

CNN is used mainly to solve object detection problems, either in two-stage algorithms such as Faster RCNN [21] and Mask RCNN [22] or single-stage algorithms such as YOLO [23] and SSD [24]. Object detection algorithms that use deep learning overcome the shortcomings of target detection algorithms, such as lack of targeted region selection,

sliding window redundancy, and time complexity. Using a deep learning algorithm for ship object detection significantly improves detection. The rotational CNN algorithm [25] used for text detection because of its excellent rotation detection capability was introduced to ship target detection and produced good results. He et al. [26] combined the Gabor filter with Faster RCNN to increase ship object detection accuracy from satellite images. Zhang et al. [27] preprocessed images with a support vector machine and then processed the RoI images with a ship detection algorithm that used a regional CNN. This technique improved the recall and precision of small ship detection and the overall performance of the algorithm. Guo et al. [28] added rotation angle information to feature extraction, which increased the detection rate of ship objects at different scales and greatly reduced the quantity of redundant information in the detection frame.

2.2. Video ranging technology

Video ranging technology is important in determining the true location of imaged objects. The two types of video ranging are monocular ranging and binocular ranging. Monocular ranging has the benefits of a simple structure, rapid operation, and low cost; it is the main field of research at present, and a commonly used method is the Kalman filter (KF). Einhorn et al. [29] devised a feature-based extended Kalman filter (EKF) monocular visual ranging measurement algorithm that captured images with a single camera and used a depth estimation method to calculate a reliable initial estimate; the 3D positions were later reconstructed by EKF. Chen et al. [30] introduced a monocular vision ranging measurement method based on pixel area and aspect ratio that predicted and optimized the pixel position in the subsequent frame using KF processing.

Another widely used technique was to calculate the distance to the object using object detection and camera projection. Raza et al. [31] used marker points to establish a line in the image and used a linear equation to calculate the real-world distance between pixels based on the length of the line. Huang et al. [32] developed a monocular vision distance measurement method using object detection and segmentation; they developed a two-dimensional geometric vector model and used camera projection to calculate distance. Zhe et al. [33] developed a monocular vision distance measurement method based on 3D detection and created a regional distance geometric model to calculate the distance based on 3D detection and camera projection that produced good results when image detail was obscured.

The geometric principle of camera projection is shown in Figure 2, where C indicates the fixed position of the camera; A and B are points on the target; A' and B' are the respective projections of points A and B on the plane of the camera sensor, which are recorded in the image; The projection line AA' and camera optical axis plane belongs to the surface method; h is the height of the center of the camera sensor above the surface; d and d' are the surface distances from the target to the vertical axis of the center of the sensor; θ is the angle between the lines OA and OB on the horizontal surface; f is the focal length of the camera. When the conditions $x=\frac{X}{2}$、 $y<\frac{Y}{2}$、 $y=y'$ are satisfied, the distances between target points and the camera are calculated by:

$$d=\frac{f\cdot h}{Y/2-y} \tag{1}$$

$$d'=\frac{d}{\cos\theta} \tag{2}$$

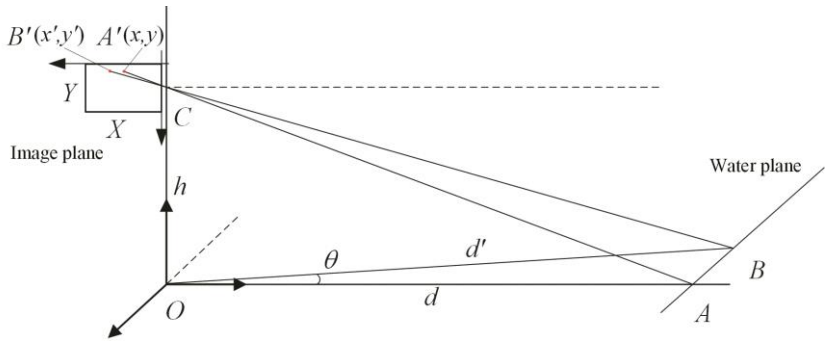


Figure 2. Geometric principle of camera projection.

3. Methods

There are four distinct stages of ship contour extraction: object detection, coordinate mapping, semantic segmentation, and image correction. Figure 3 is the technical roadmap of this research and shows how data is transmitted between the various stages. In this section, we describe the key steps in detail.

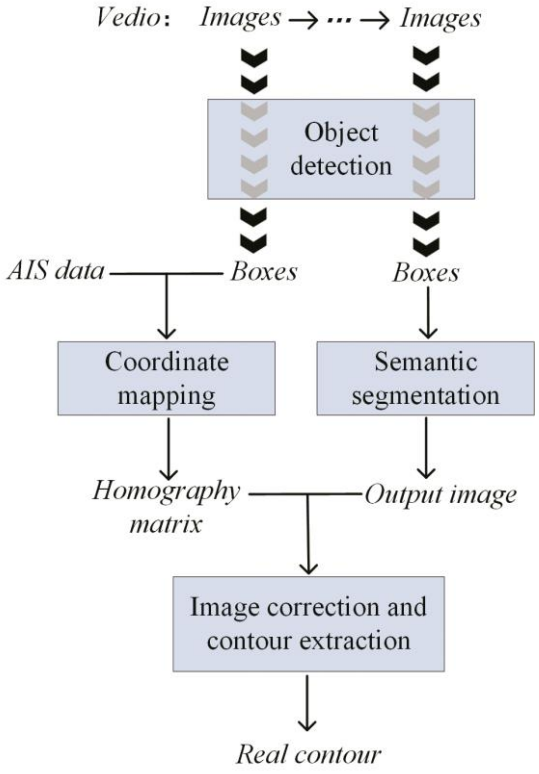


Figure 3. Technology roadmap of ship contour extraction and correction: vi[illegible]nd AIS are input data, and the optimal homography matrix firstly be calculated, which is [illegible] to map the segmented image, after image correction and contour extraction, the final actual contour is output.

3.1. Object detection model

In general, object detection models often have a deep network structure and require a large number of convolution layers with many parameters. Real-time inferences can only be performed if the devices they run on have adequate computing power. Nanodet is an excellent lightweight object detection model introduced in 2020 that dramatically reduces the number of parameters using a series of optimization methods. Nanodet can be quickly trained and ported to most embedded modules.

Nanodet uses several lightweight methods in the backbone, neck, and head, which enables it to balance accuracy, speed, and processing volume. The backbone is ShuffleNet V2, which removes the last layer of the convolution from the network and extracts 8, 16, and 32 downsampled features as the next inputs. ShuffleNet V2 is a CNN architecture that uses pointwise group convolutions to simplify the calculation of 1×1 convolutions and uses channel shuffle to resist negative influence. The network greatly reduces computation but maintains accuracy. The neck, which is an optimized PAN, deletes all convolutions in the PAN and only uses 1×1 convolutions extracted by the backbone for channel dimension alignment. An interpolation algorithm is used for upsampling and downsampling, and the multiscale feature map is added for feature fusion, which enables the network to learn the characteristics of multiscale targets. The optimized FCOS model was used as the detection head with abandoned weight sharing. FCOS uses different convolutions to extract features at each layer and uses batch normalization, which uses deep separable convolution instead of group normalization. The number of convolution kernels and convolution channels also decreases, and the generalized focal loss function is used to resolve problems of convergence in training. In all, these methods greatly reduce redundant convolution and the number of parameters in the model, thus decreasing computation time.

SENet [34] is a spatial attention mechanism that increases the depth of a CNN and improves feature extraction. It consists of squeeze, excitation, and reweight functions. In the squeeze stage, the feature space with dimensions $c \times h \times w$ is compressed to $c \times 1 \times 1$ by global pooling, and the feature maps of a single channel are compressed into a weight factor. Two fully connected devices are used in the excitation stage. The first compresses the global information obtained from global pooling; the feature dimension $c \times 1 \times 1$ is reduced to $c/r \times 1 \times 1$. The second fully connected device is used to map the feature back to $c \times 1 \times 1$ after ReLU activation. A sigmoid function is used to determine the normalized weight, in the range 0–1, of each channel to multiply the original feature map.

The network structure is shown in Figure 4. SENet was added between ShuffleNetV2 and the optimized PAN with 40×40, 20×20, and 10×10 feature maps as the input to emphasize useful features and suppress irrelevant features. Figure 4 shows the two feature maps before and after the SENet mechanism for comparison and displays the SENet-enhanced features. After the calculations for PAN feature fusion and the FCOS detection head, the final output represents the location of candidate boxes and their scores for different categories.

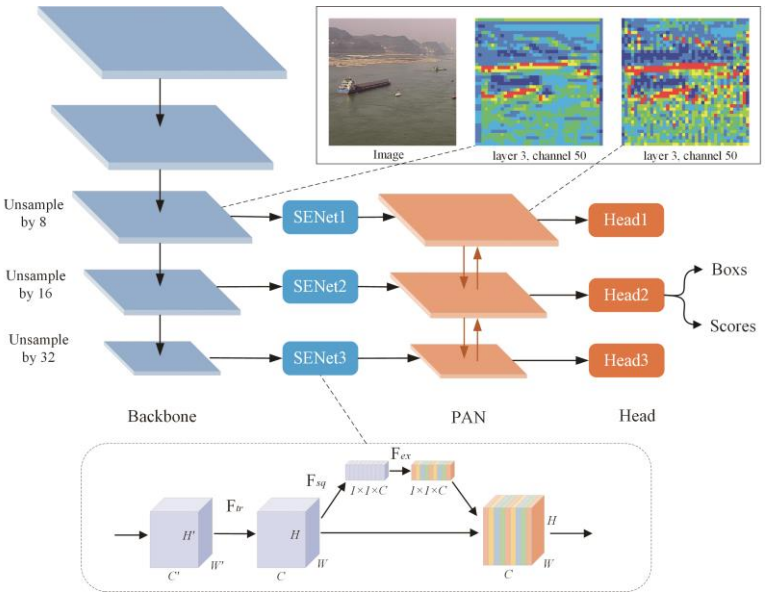


Figure 4. Nanodet-SENet network structure.

3.2. Coordinate mapping

Mapping sensor image coordinates onto the water surface is key to matching image information with real-world information. When the camera attitude is constantly shifting, it is often necessary to combine optical ranging methods to ensure the camera view is parallel to the water surface. When parameters for camera height above the water surface, focal length, and pitch angle are combined with the projection equations, the depth map of the image can be calculated to predict the distance to each pixel on the water surface in the image. However, deploying surveillance cameras in inland river navigation areas is complex, and parameters such as height, focal length, and pitch angle are difficult to obtain timelily. The use of data from fixed monitoring locations often requires using a homography transformation matrix to convert between the sensor image coordinate system and the water surface coordinate system, depending on the control points, and then mapping the pixel coordinates of the ship contour image to the water surface coordinates. The equation for the homography matrix transformation is:

$$[x' \;\; y' \;\; w'] = [u \;\; v \;\; 1] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix} = [u \;\; v \;\; 1] H \tag{3}$$

where u and v are the pixel coordinates of the control points, and $\boldsymbol{H}$ is the homography transformation matrix. At least four control points are required for the eight independent parameters in the solution of $\boldsymbol{H}$.

We used the real-time AIS position data and the corresponding observed ship positions in the image to create several control point coordinates, as shown in Figure 5. The specific calculation steps of the optimal homographic transformation matrix are as follows.

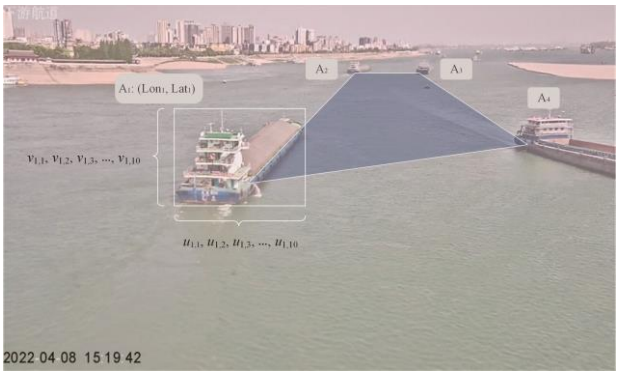


Figure 5. Multi-group control points constructed based on AIS information.

1. Obtain the video and AIS data for two ships on different courses to form four groups of matching coordinates and AIS positions. One is $A_i = \{(Lon_i, Lat_i), Box_i\}$, where Lon_i and Lat_i are the latitude and longitude coordinates of the control point i, and $\text{Box}_i = (u_{i1},v_{i1}),(u_{i2},v_{i2}) \ldots, (u_{in}, v_{in})$ is the set of pixel coordinates of control points in group i, which consists of coordinates of n equally spaced pixels. The control points of two ships traveling in the forward and backward directions are combined in pairs to produce several homography transformation matrices;
2. Detect the key point $\boldsymbol{p}_i$ of Box_i, where $\boldsymbol{p}_i$ is the intersection of the extended side ged in the lateral and inferior directions from the saliency detection image produced by the LC model. The two waterlines are determined by linear fitting;
3. Select the two sets of coordinates for one ship, Box_1 and Box_2. Two key point pixels $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are calculated as in Step2, and Eq. (3) is used to calculate the corresponding n^2 groups of water surface coordinates for $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. The error calculation of the homography matrix $\boldsymbol{H}_j$ is:

$$\overrightarrow{P_j} = p'_{2j} - p'_{1j}, \quad j=1,2,3,\ldots,n^2 \tag{4}$$

$$lon'_2 = lon_2 \times \cos\theta - lat_2 \times \sin\theta \tag{5}$$

$$lat'_2 = lat_2 \times \cos\theta + lon_2 \times \sin\theta \tag{6}$$

$$\overrightarrow{K} = (lon'_2, lat'_2) - (lon_1, lat_1) \tag{7}$$

$$\beta_j = \cos^{-1}\left(\frac{\overrightarrow{P_j} \cdot \overrightarrow{K}}{|\overrightarrow{P_j}| \times |\overrightarrow{K}|}\right), \quad j=1,2,3,\ldots,n^2 \tag{8}$$

where $\boldsymbol{p}'_{1j}$ and $\boldsymbol{p}'_{2j}$ are respectively the mapped coordinates of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and θ the steering angle of the ship as determined from the AIS information. When the smallest β_α has been obtained, the matrix $\boldsymbol{H}_\alpha$ calculated by β_α is considered to be the optimal homography matrix.

3.3. Semantic segmentation model

Ship contour extraction requires segmentation of the area covered by the ship surface when viewed from above, but the area is often obscured by the superstructure of the vessel. Commonly used image segmentation algorithms are often greatly affected by noise and lack robustness, and it is difficult to determine the target area when it is obscured. However, the deep learning DeepLabV3+ semantic segmentation model [35] is highly accurate, robust, and not very susceptible to noise. Therefore, it is suitable for use in the segmentation of specific targets in a complex environment.

The network structure of DeepLabV3+ is shown in Figure 6. It consists of an encoder and a decoder. The main body of the encoder is a deep CNN with dilated convolution that controls the size of the receptive field by a rate (r) without changing the size of the feature graph. A greater value of r produces a larger receptive field. The dilated convolution in the encoder is combined with a spatial pyramidal pooling module to produce multiscale information. The main constituents of the encoder are the following:

a. One 1×1 convolution layer and three 3×3 empty convolution layers. The rate r is (6,12,18) when the output step size is 16 and is doubled when the output step size is 8.

b. A global average pooling layer is used to produce image-level features that are then input into the 1×1 convolution layer and bilinearly interpolated to the original size.

c. Five features of different scales produced by steps a and b are combined in the dimension channel and then input into the 1×1 convolution layer to be combined to produce 256 channels of new features.

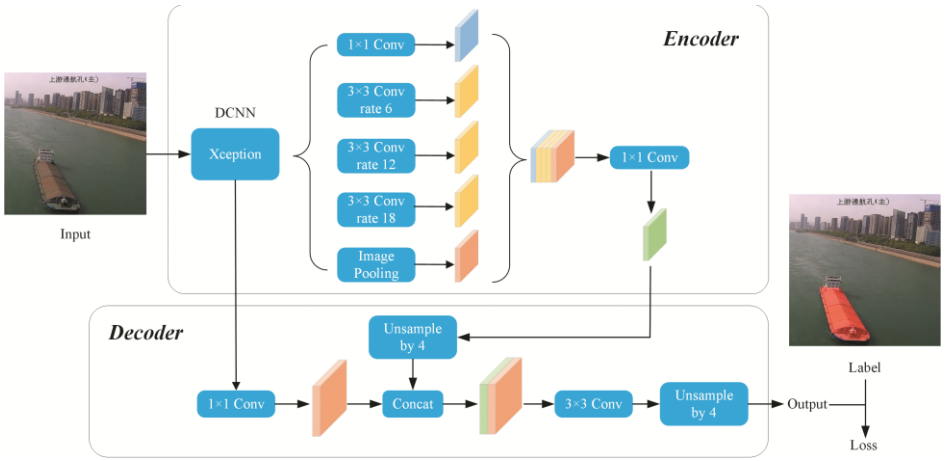


Figure 6. DeepLabV3+ network structure.

The decoder can also combine low-level features with high-level features to increase segmentation accuracy. The main steps of feature fusion are as follows. The multiscale feature information is bilinearly interpolated and upsampled. The encoder then combines it with the original features that were extracted by the CNN. The combined feature information is then convoluted for simple feature combination. Finally, the combined features are bilinearly interpolated and upsampled to produce the segmentation results.

Xception [36] was used as the backbone network for feature extraction (Figure 6). Xception, which is an improved version of Inception V3, introduces depthwise convolution derived from Inception V3 to reduce model complexity and improve segmentation. The label in Figure 6 is from the annotation of the input images. The spatial pyramid module combined with dilated convolution combines multiscale image information. A larger value of *r* will extract features from different regions of the image into a larger receptive field, thus reducing obscuration by the superstructure. The marked area is the area where the ship is vertically mapped onto the water surface.

4. Results

4.1. Ship object detection

The experimental platform was a desktop computer with Windows 10, a GTX1050Ti GPU, and the PyTorch 1.8.0 framework. We created a coco data set for training with 2535 images collected from surveillance videos from waterways upstream and downstream of the Yichang Yangtze River Bridge and the Jingzhou Yangtze River Bridge in China. The images included ships from different angles, of different sizes, and in various lighting conditions. The image count was increased to 5070 using data enhancement methods such as noise processing, random angle rotation, random brightness adjustment, and simulated rain and fog weather conditions. Labelme software was used to annotate the images, which contained 12,376 ship objects altogether. The dataset was divided for training, validation, and testing in the ratio 8:2:1. The input size for detection was 320×320. SGD was used to optimize with the initial learning rate 0.01.

The object detection models were trained and tested before and after validation, and their precision and recall were calculated. Precision represents the proportion of correctly predicted targets in total predictions, and recall represents the proportion of all target predictions that were correct. In general, precision decreases as recall increases.

To assess detection improvement attributed to SENet, P–R curves were plotted using precision and recall values for the two models before and after validation. In addition, the mainstream object detection algorithms Faster RCNN and YOLOv4 and the lightweight algorithm YOLOv4-tiny using the test set were selected for comparison to further assess the accuracy and efficiency of the Nanodet–SENet model detection. Table 2 shows the number and size of parameters for these three models. It can be seen from the table that Nanodet–Senet is an excellent lightweight model because its network complexity is much less than that of other models.

Table 2. Params number and size of different models.

	Total params	**Params size**
Faster RCNN	137.08M	522.91MB
YOLOv4	64.36M	245.53MB
YOLOv4-tiny	6.06M	23.10MB
Nanodet-Senet	**0.95**M	**3.62**MB

The P–R curves for the experiment are shown in Figure 7. Detection by several algorithms was assessed using four indicators: mean Average Precision (mAP), frames/s

(FPS), precision, and recall, which are shown in Table 3. It can be seen from the table that the Nanodet–Senet model outperformed Nanodet, YOLOv4-Tiny and Faster RCNN in terms of precision and mAP but did not perform as well as YOLOv4. Recall was greater than for Nanodet and YOLOv4-Tiny, but less than for Faster RCNN and YOLOv4. FPS was significantly greater than for all the other models except Nanodet. These results indicate that the SENet attention mechanism significantly influenced the detection effectiveness of the model and that Nanodet–Senet detected objects almost as well as YOLOv4 although it is a simpler model.

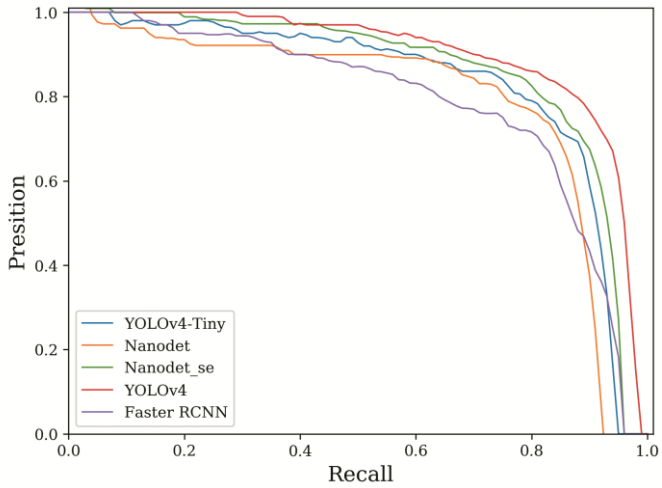


Figure 7. P-R curve.

Table 3. Comparison test results of target detection.

	Precision (%)	**Recall** (%)	**mAP** (%)	**FPS**
Faster RCNN	69.68	**81.51**	78.84	6.35
YOLOv4	86.23	79.64	**90.20**	9.10
YOLOv4-tiny	84.20	74.55	83.68	30.71
Nanodet	78.96	74.72	76.55	38.62
Nanodet-Senet	**85.09**	76.83	86.73	**37.47**

4.2. Establishment of the homography transformation model

We combined AIS information with video data to create the optimal homography transformation matrix between the sensor pixel coordinate system and the water surface coordinate system. The critical aspect of this algorithm is the synchronization of the time of the image sequence with the time of AIS information acquisition to ensure the accuracy of key point detection.

A Hikvision zoom network camera remotely captured video stream was transmitted using a real-time streaming protocol (RTSP), and a message queue telemetry transmission (MQTT) server was used to create an AIS information transmission platform. The video transmission rate was 3.2 Mbps with a 3.5 s delay, and the AIS signal delay was 6 s. As described in Section 3.2, the delays were eliminated, and the video and AIS information was used to obtain the coordinates of 250 sets of control points.

The key points were then tested. The accuracy of key point detection decreases for small-target ships, so we needed to ensure the ship detection boxes we selected as the control points were of adequate size and had well-defined contours, as shown in Figure 8. Figures 8a and 8b show ship objects detected at two distinct time points. The ship moving away from the sensor on the left side of the images was used to validate the op-

timal homography matrix. The red pixels in Figures 8c and 8d show the LC significant image pixels of change points, and the green pixels show key points after linear fitting. Figure 8d shows that the model predicted the exact position of the key point when the ship was further away from the sensor.

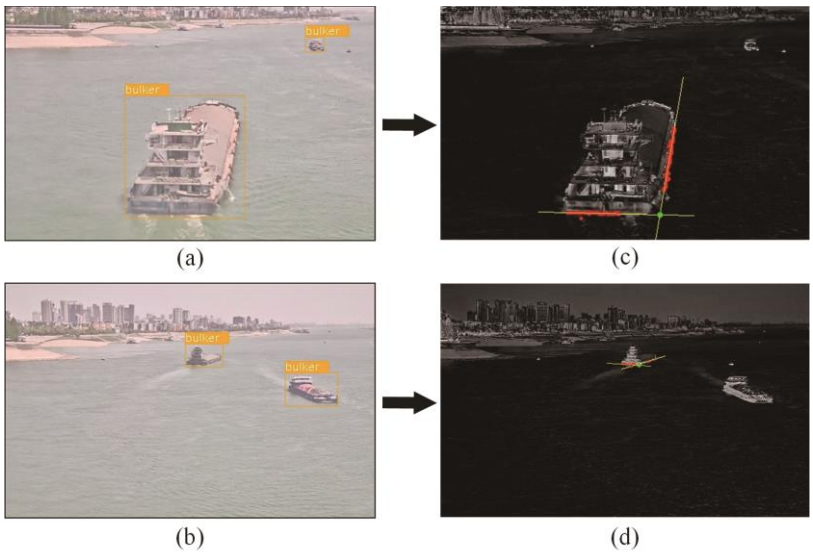


Figure 8. Key point detection progress: (a) ship object detection at first time, (b) ship object detection at second time, (c) key point detection at first time, and (d) key point detection at second time.

After we derived the optimal homography matrix, as described in Section 3.2, we used sensor data for the upstream and downstream directions to test the homography transformation model. Consecutive AIS coordinates were obtained, and the pixel coordinates of key points in the corresponding images were obtained and mapped onto the water coordinate system to predict the trajectory. Figure 9 shows that the trajectory of key points after coordinate mapping was very close to the AIS trajectory. The mean values of the distances between the corresponding points in the two sets of trajectories, 11.5 meters and 77.8 meters, were consistent with relative positions and distances between the real-world AIS antenna and the ship waterline inflection point. This result shows the accuracy of coordinate mapping using the homography transformation model.

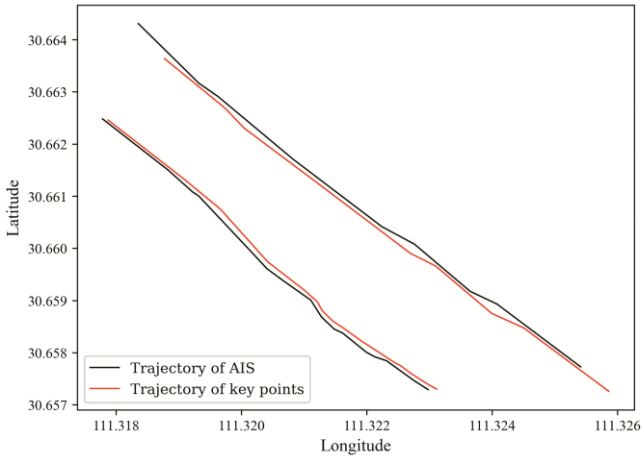


Figure 9. Test results of homography transformation model.

4.3. Ship contour extraction

Only images of bulk carriers and oil tankers were used because the irregular shapes of passenger and container ships add unnecessary complexity to the development of the

prototype model. We used surveillance cameras to get overhead views of vessels on the waterway to ensure that the deck surface matched the ship hull as much as possible. The object detection method was used to take automatic photos which were stored to create a semantic segmentation data set. The model training platform was a desktop computer with Windows 10 and an RTX3060Ti GPU using the Keras 2.2.5 framework. Target ship contours were labeled using Labelme software, and pixels were classified as either foreground or background. The ratio of the training set, validation set, and test set was 8:2:1. Online data enhancement was used to randomly amplify the image and label data in each batch in the training stage. The input data size for the model was 512×512.

Mean intersection over union (MIoU), which is a standard metric of the accuracy of a semantic segmentation algorithm, was used to assess the performance of the algorithm. The equation is:

$$\mathrm{MIoU}=\frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \tag{9}$$

where k is the number of pixel categories, p_{ij} is the number of pixels that originally belonged to category i but are predicted to be in category j, and MIoU is the average number of times the predicted value coincides with the actual value in each category. A greater value of MIoU indicates more accurate network prediction.

FCNs (fully convolutional networks) are commonly used for ship image semantic segmentation [37]. An FCN classifies images at a pixel level by selecting a sliding window for each pixel. Unet (unity networking), an improvement on FCN [38], has been widely used in the field of transportation. We therefore compared these two models with DeepLabV3+. The IoU results for background and foreground after training for 50 epochs are shown in Table 4. It can be seen that DeepLabV3+ has clear advantages over FCN and Unet.

Table 4. Comparison test results of semantic segmentation.

	Background_IoU (%)	**Boat_IoU** (%)	**Mean_IoU** (%)
FCN	99.42	79.80	89.61
Unet	99.28	67.35	83.32
DeeplabV3+	**99.54**	**82.61**	**91.07**

The output images of DeepLabV3+ needed to be trimmed. Trimming was performed by calculating the number of pixels in the stern area along the transverse axis of the ship and setting a threshold to eliminate scattered pixels to avoid individual misclassified pixels having undue impact on the corrected image. The segmentation results are shown in the first three columns of Figure 10. It can be seen from Figure 10 that the DeepLabV3+ model segmented the target area well without obstruction from the superstructure. The fourth column in Figure 10 shows the trimmed images that represent the actual ship contour region after coordinate mapping. The pixel size of the image corresponds to a real-world distance of 115 meters. It is clear from these results that the size and heading of ships at different distances in the image were approximately estimated.

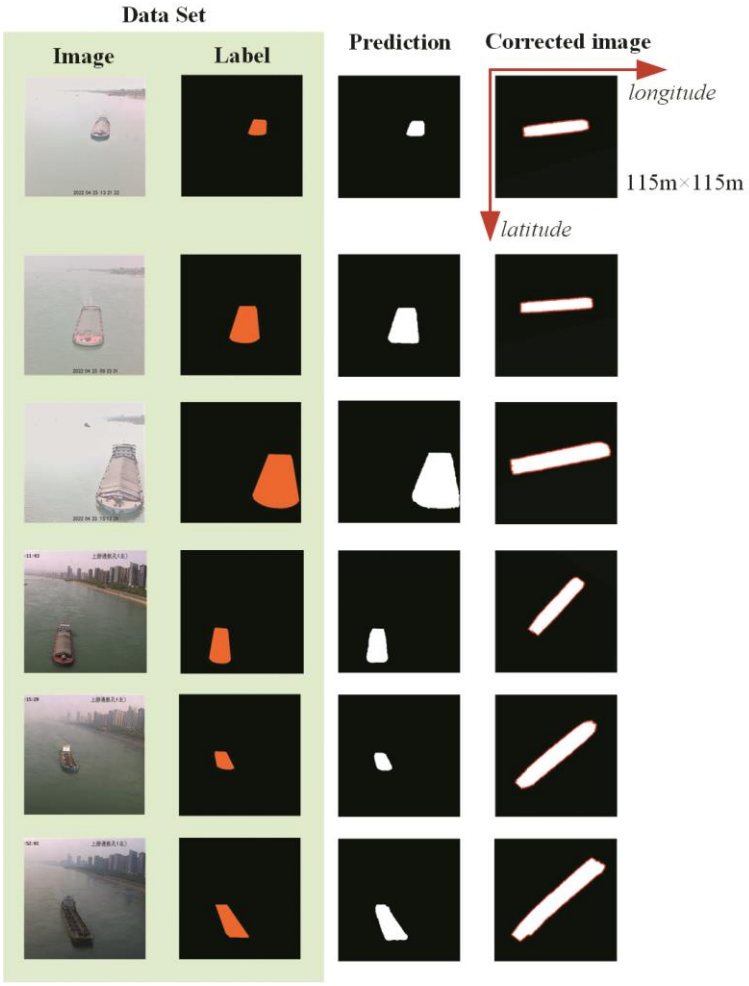


Figure 10. Ship contour extraction and size restoration results.

A comprehensive review of all the experimental results shows that the DeepLabV3+ semantic segmentation model can be successfully used on a high-performance server that receives automatic photos and uploading based on lightweight object detection and accomplishing contour extraction tasks on the cloud server, thus avoiding bandwidth usage for transmitting large amounts of remote video data. The object detection model will determine the rectangular region surrounding a vessel, increase the proportion of ship features in DeepLabV3+ input and improve the pertinence of the semantic segmentation model.

Based on the restored image, edge detection was combined with the Hough transform to further detect the minimum enclosing rectangle of the ship contour. We tested for several ships by estimating their dimensions which we then compared with the actual dimensions; the results are shown in Table 5. We found that the average length error was 5.84% and the average width error was 7.53%; these values are in a reasonable range. This final result is valuable for creating and improving AIS data and provides a foundation for calculating ship collision risk.

Table 5. Ship dimension data calculation results and errors.

MMSI	Length (m)			Width (m)		
	Actual	Calculated	Error (%)	Actual	Calculated	Error (%)
413773165	87	90.3	3.8	14	14.6	4.3
413774959	87	84.2	3.2	15	16.3	8.7
413779378	90	96.7	7.4	15	17.0	13.3
413781326	106	110.6	4.3	17	18.2	7.1
413783151	107	115.2	7.7	16	17.1	6.9
413803847	90	96.8	7.6	15	15.4	2.7
413811188	100	97.5	2.5	16	16.9	5.6
413819165	100	114.2	14.2	17	19.8	16.5
413831856	110	113.6	3.3	19	20.0	5.3
413801536	107	103.1	3.6	16	16.6	3.8
Mean	-	-	5.8	-	-	7.4

5. Conclusion

Ship dimension information is important at the micro level in ship collision risk calculation. To avoid having to work with missing or incorrect ship dimension data, deep learning algorithms were used to extract ship contours from inland waterway surveillance video and real-time AIS information. According to the experimental results, the proposed object detection model and semantic segmentation model were very accurate in our experimental trials and thus successfully resolved the missing AIS data issue.

The easily used lightweight ship object detection model that we developed for edge computing processed at a high frame rate without GPU acceleration and facilitated the automatic acquisition and uploading of ship images. It therefore provides crucial support for shipping control on inland waterways.

In this study, AIS real-time position information was taken as a virtual control point to create a coordinate mapping model which rapidly and accurately mapped sensor image coordinates onto water surface coordinates without the need to rely on various projection parameters. This method therefore merits future development to further promote the combination of video, AIS, and radar information.

The combination of the object detection model and the coordinate mapping model, together with the use of the semantic segmentation algorithm, allowed us to extract the ship contour from the image and predict its actual size. The experimental trial results suggest the method is effective and demonstrate the capability to extract ship dimension information automatically. This capability will benefit inland waterway regulatory authorities by enabling them to improve management of ship navigation.

Errors in ship contour extraction mainly come from the coordinate mapping model and semantic segmentation model. In our future research, we will add more ship images to the semantic segmentation dataset to improve the generalizability of the dataset so that the model can more exactly segment images of different types of ships. We will also endeavor to constantly use ships to simulate the control points during the experiment in order to continuously update the optimal homography matrix.

Author Contributions: methodology, validation, writing, original draft—Z. H. and Q. M.; data curation—L. L. and C. Y.; review and editing, funding acquisition—Q. H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 71804059), Social Development Major Project of Shanghai Municipal Science and Technology Commission (Grant No. 18DZ1206300).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

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

References

1. Bakdi, A.; Glad, I.K.; Vanem, E.; Engelhardtsen, Y. Ais-based multiple vessel collision and grounding risk identification based on adaptive safety domain. *Materials*. **2019**, 8, 5-25.
2. Liu, W.R.; Guo, Y.; Nie, J.; Hu, Q.; Xiong, Z.; Yu, H.; Guizani, M. Intelligent Edge-Enabled Efficient Multi-Source Data Fusion for Autonomous Surface Vehicles in Maritime Internet of Things, *IEEE Transactions on Green Communications and Networking*. **2022**, 6, 1574-1587.
3. Huang, Z.S.; Hu, Q.Y.; Mei, Q.; Yang, C.; Wu, Z. Identity recognition on waterways: A novel ship information tracking method based on multimodal data. *Journal of Navigation*. **2021**, 74, 1336-1352.
4. Park, J.J.; Park, K.A.; Foucher, P.Y.; Lee, M.; Oh, S. Estimation of ship size from satellite optical image using elliptic characteristics of ship periphery. *International Journal of Remote Sensing*. **2020**, 41, 1-23.
5. Yan, Z.; Yan, X.; Xie, L. Inland Ship Edge Detection Algorithm based on Improved Canny Operator. *Journal of Convergence Information Technology*. **2012**, 7, 567-575.
6. Gu, D.; Xu, X. Multi-feature extraction of ships from SAR images. 2013 6th International Congress on Image and Signal Processing (CISP), Hangzhou, China, 16-18 December 2013; pp. 454-458.
7. Zhu, J.; Qiu, X.; Pan, Z.; Zhang, Y.; Lei, B. Projection shape template-based ship target recognition in TerraSAR-X images. *IEEE Geoscience and Remote Sensing Letters*. **2017**, 14, 222-226.
8. Nie, Y.Y.; Fan, S.C.; Shui, P.L. Fast ship contour extraction in SAR images. *The Journal of Engineering*. **2019**, 19, 5885-5888.
9. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
10. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Rabinovich, A. Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 07-12 June 2015; p. 15523970
11. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International conference on machine learning, Lile, France, 6-11 July 2015; pp. 448-456.
12. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016; P. 16541111.
13. Lin, H.; Shi, Z.; Zou, Z. Fully Convolutional Network with Task Partitioning for Inshore Ship Detection in Optical Remote Sensing Images. *IEEE Geoscience and Remote Sensing Letters*. **2017**, 14, 1665-1669.
14. Bovcon, B.; Kristan, M. A water-obstacle separation and refinement network for unmanned surface vehicles. *arXiv* **2020**, arXiv:2001.01921.
15. Ust, L.; Kristan, M. Learning maritime obstacle detection from weak annotations by scaffolding. *arXiv* **2022**, arXiv:2108.00564.
16. Kelm, A.P.; Zolzer, U. Walk the Lines: Object Contour Tracing CNN for Contour Completion of Ships. 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 13-18 September 2020; pp. 3993-4000.
17. Howard, D.; Roberts, S.C.; Brankin, R. Evolution of Ship Detectors for Satellite SAR Imagery. *Lecture Notes in Computer Science*. **1999**, 1598, 135-148.
18. Kim, H.T.; Park, J.S.; Yu, Y.S. Ship detection using background estimation of video and AIS informations. *Journal of the Korea Institute of Information and Communication Engineering*. **2010**, 14, 2636–2641.
19. Fefilatyev, S.; Goldgof, D.; Shreve, M.; Lembke, C. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*. **2012**, 54, 1-12.
20. Yang, T.J.; Zhang, S.; Zhou, G.Q.; Jiang, C.X. Design of a real-time system of moving ship tracking on-board based on FPGA in remote sensing images. International Conference on Intelligent Earth Observing and Applications 2015, Guilin, China, 9 December 2015; p. 980804.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2017**, 39, 1137-1149.
22. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22-29 October 2017; pp. 2980-2988.
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016; pp. 779-788.
24. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 10-16 October 2016; pp.21-37.

25. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*. **2018**, 20, 3111-3122.
26. He, L.; Yi, S.; Mu, X.; Zhang, L. Ship Detection Method Based on Gabor Filter and Fast RCNN Model in Satellite Images of Sea. the 3rd International Conference on Computer Science and Application Engineering (CSAE), Sanya, China, 22-24 October 2019; pp. 1–7.
27. Zhang, S.; Wu, R.; Xu, K.; Wang, J.; Sun, W. R-cnn-based ship detection from high resolution remote sensing imagery. *Remote Sensing*. **2019**, 11, 631-646.
28. Guo, H.; Yang, X.; Wang, N.; Song, B.; Gao, X. A rotational libra r-cnn method for ship detection. *IEEE Transactions on Geoscience and Remote Sensing*. **2020**, 58, 5772-5781.
29. Einhorn, E.; Schroter, C. Attention-driven monocular scene reconstruction for obstacle detection, robot navigation and map building. *Robotics and Autonomous Systems*. **2011**, 59, 296-309.
30. Chen, S.; Mei, S.; Jia, G. KFPA Monocular Ranging Algorithm Design and Application in Mobile edge Computing. *Journal of Internet Technology*. **2021**, 22, 1131-1142.
31. Raza, M.; Chen, Z.; Rehman, S.U. Framework for estimating distance and dimension attributes of pedestrians in real-time environments using monocular camera. *Neurocomputing*. **2018**, 275, 533-545.
32. Huang, L.; Zhe, T.; Wu, J.; Wu, Q.; Pei, C.; Chen, D. Robust inter-vehicle distance estimation method based on monocular vision. *IEEE Access*. **2019**, 7, 46059-46070.
33. Zhe, T.; Huang, L.; Wu, Q.; Zhang, J.; Pei, C.; Li, L. Inter-vehicle distance estimation method based on monocular vision using 3d detection. *IEEE Transactions on Vehicular Technology*. **2020**, 69, 4907-4919.
34. Jie, H.; Li, S.; Gang, S.; Albanie, S. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2017**, 8, 2011-2023.
35. Chen, L.C.; Papandreou, G.; Kokkinos L. Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE transactions on Pattern Analysis and Machine Intelligence*. **2018**, 40, 834-848.
36. Chollet, F. Xception: deep learning with depthwise separable convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, 21-26 July 2017; pp. 1800-1807.
37. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **2015**, 39, 640-651.
38. Ronneberger, O.; Fischer, P.; Brox, T. U-net: convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5-9 October 2015; pp. 234-241.

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.