Electronic supplementary materials

For https://doi.org/10.1631/jzus.A2500277

Digital twin-assisted automatic ship size measurement for ship–bridge collision early warning systems

Ruixuan LIAO[1], Yiming ZHANG[1], Hao WANG[1], Jianxiao MAO[1], Aoyang LI[2], Zhengyi CHEN[1,3]

[1]*Key Laboratory of Concrete & Prestressed Concrete Structures of Ministry of Education, Southeast University, Nanjing 211189, China*
[2]*Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana IL 61801, USA*
[3]*Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China*

Section S1. Virtual environment modelling

Given the imported geometric and semantic resources, the physical properties should be created for the virtual camera, as illustrated in Fig. S1.

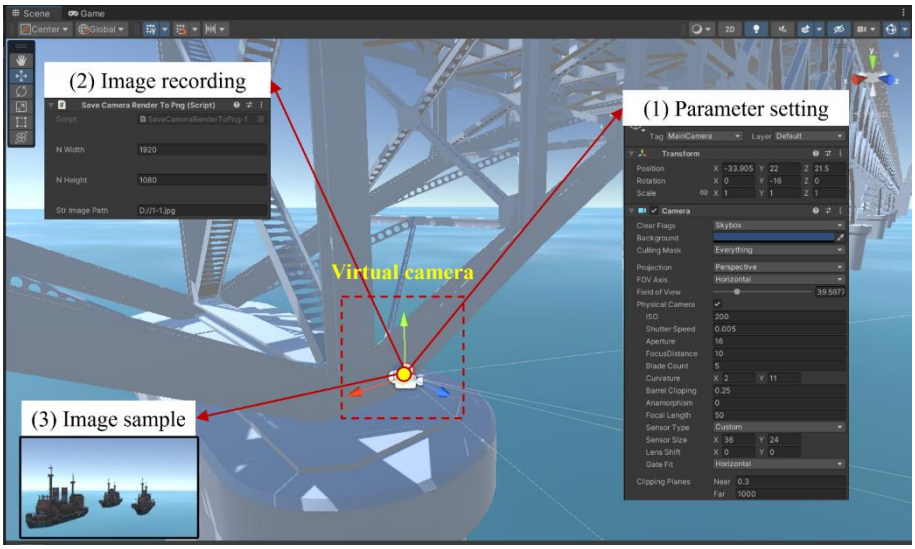


Fig. S1. Virtual camera modelling.

(1) Parameter setting: The spatial position and rotation angle of the virtual camera model should be first set to enable it to capture images from a specific orientation. Other physical properties of the camera, such as focal length, field of view, and sensor size, can also be

configured.

(2) Image recording: scripting application programming interfaces (in C#) are created to capture and save images, allowing users to adjust the image resolution.

(3) Image sample: An interface enabling for viewing of images captured by the virtual camera in real time. When observing a 2D screen image of the 3D world, the virtual camera is used to capture a view for display.

The navigational environment should be modelled after deploying the virtual camera, mainly including ship and water area modelling (Fig. S2).

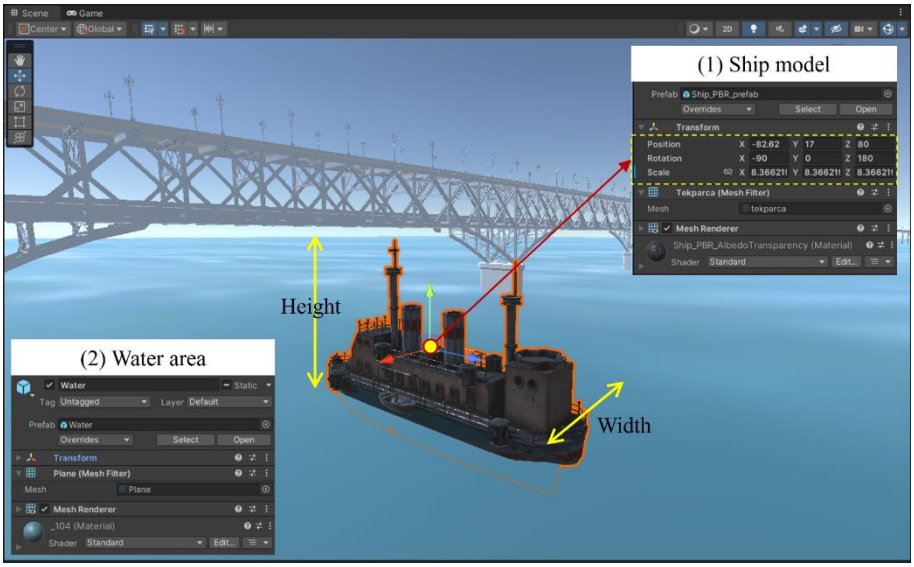


Fig. S2. Ship navigation modelling.

(1) Ship model: The position of the ship model varies within the field of view of the virtual camera. The dimensions of the ship model are determined by the scale parameter; a larger scale factor results in greater height and width of the ship model.

(2) Water area: A range of variables, including the mesh, the angle of direct sunlight, and the intensity of the light, are used to characterize the water area.

Section S2. You Only Look Once Version 8 (YOLOv8) model

The structure of the YOLOv8 network is displayed in Fig. S3. In YOLOv8, the significant improvement is reflected by the Spatial Pyramid Pooling-Fast (SPPF) and Cross Stage Partial Bottleneck with Two Convolutions (C2f) structures. The SPPF module enables adaptive output sizes, enhancing sensitivity and capturing feature information at various levels within the image, contributing to improved feature extraction at the end of the backbone. The C2f module, inspired by the efficient layer aggregation network, is a lightweight convolutional structure designed to enhance gradient propagation efficiency and enable faster network convergence [S1].

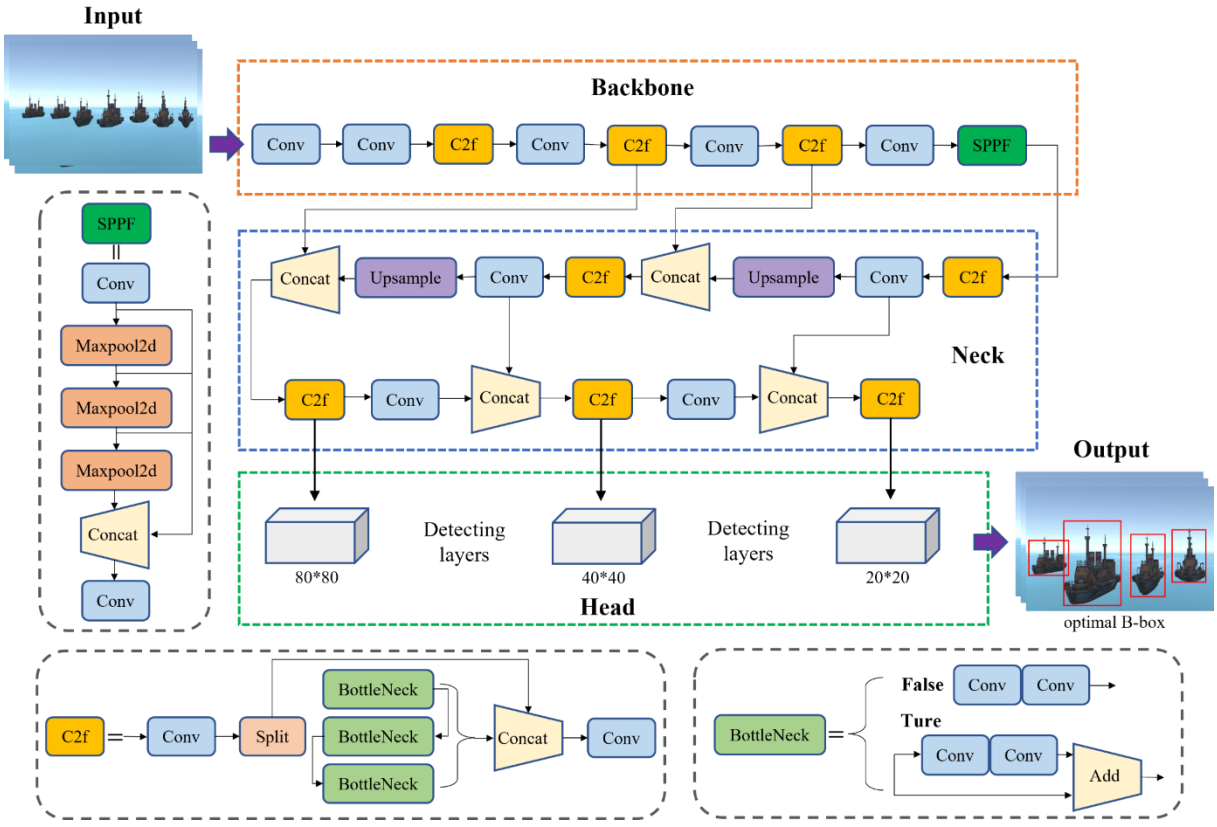


Fig. S3. The structure of the YOLOv8 network.

Section S3. Calculation formulas for K, R, and t

The alignment between a ship's 3D spatial and 2D pixel information can be established

by mapping the relationship between the local and pixel coordinate systems (Yoon et al., 2018), as illustrated in Fig. S4.

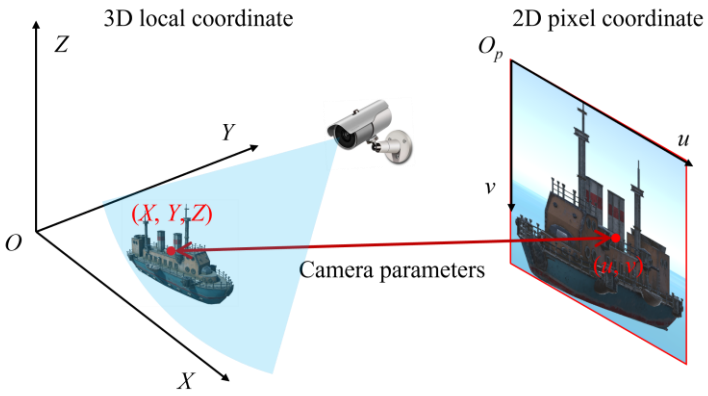


Fig. S4. Mapping between the two coordinate systems.

The calculation formulas for **K**, **R**, and **t** are given by [S2]

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{S1}$$

$$\mathbf{R} = \begin{bmatrix} -\cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{S2}$$

$$\mathbf{t} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{S3}$$

where f_x and f_y are the focal lengths of the camera in the horizontal and vertical directions, respectively, (c_x, c_y) is the principal point, which is the point where the optical axis intersects the image plane, θ represents the yaw angle of the virtual camera (in this study, only yaw rotation is required for the virtual camera), and (x_c, y_c, z_c) denote the local coordinates of the installation position for the virtual camera. The above formulas are used to convert the 3D spatial coordinates of all ships into 2D-pixel coordinates.

Section S4. Mapping real-world ship coordinates to the virtual environment

The latitude-longitude coordinates of ships can be converted to Earth-Cantered, Earth-Fixed (ECEF) coordinates by [S3]

$$\begin{cases} X_e = \left(N_0 + h\right)\cos\left(\dfrac{lat \times \pi}{180}\right)\cos\left(\dfrac{lon \times \pi}{180}\right) \\ Y_e = \left(N_0 + h\right)\cos\left(\dfrac{lat \times \pi}{180}\right)\sin\left(\dfrac{lon \times \pi}{180}\right) \\ Z_e = [N_0(1-e^2) + h]\sin\left(\dfrac{lon \times \pi}{180}\right) \end{cases} \tag{S4}$$

where (X_e, Y_e, Z_e) are points in the ECEF coordinate system, (*lat*, *lon*) refers to geographic coordinates in latitude and longitude, N_0 denotes the radius of curvature of the Earth corresponding to different latitudes, typically expressed by the formula $N_0 = \dfrac{a}{\sqrt{1-e^2\sin^2(\text{lat})}}$, a is the equatorial radius, e represents the first eccentricity of the Earth's ellipsoid, with a value approximately equal to 0.081819, and h denotes the apparent height of the observation point relative to the Earth's surface.

The latitude-longitude coordinates of the real-world camera are converted into the ECEF coordinates (x_r, y_r, z_r) through Eq. (S4), from which the translation vector (t_x, t_y, t_z) between the ECEF and local coordinate systems can be derived as

$$\begin{cases} t_x = x_c - x_r \\ t_y = y_c - y_r \\ t_z = z_c - z_r \end{cases} \tag{S5}$$

where (x_c, y_c, z_c) is the coordinates of the virtual camera in the local coordinate system of the simulated space.

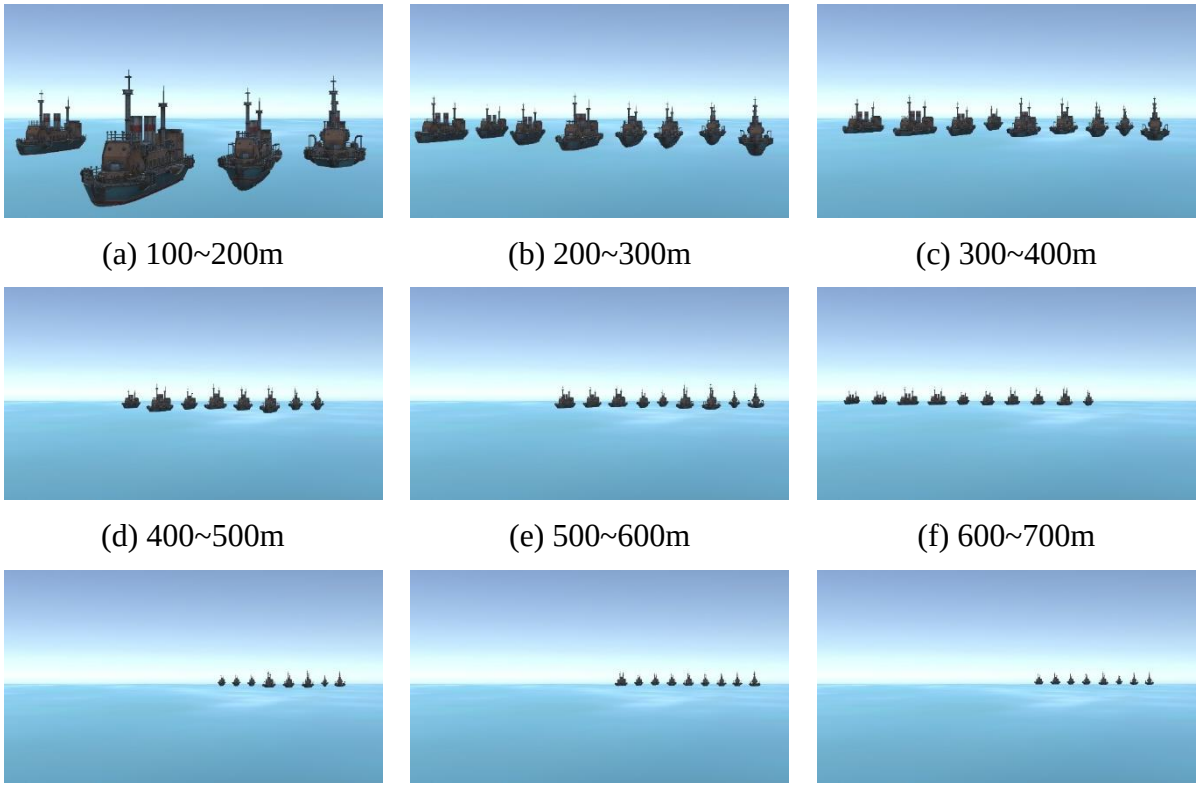
All latitude-longitude coordinates of real-world ships can be further transformed into the local coordinate system using

$$\begin{cases} X_s = X_e + t_x \\ Y_s = Y_e + t_y \\ Z_s = Z_e + t_z \end{cases} \tag{S6}$$

where (X_s, Y_s, Z_s) are the transformed points in the local coordinate system of the virtual world.

Section S5. Synthetic ship images and detection results

Fig. S5 provides image samples of synthetic ships at varying camera-to-ship distances.



(a) 100~200m (b) 200~300m (c) 300~400m

(d) 400~500m (e) 500~600m (f) 600~700m

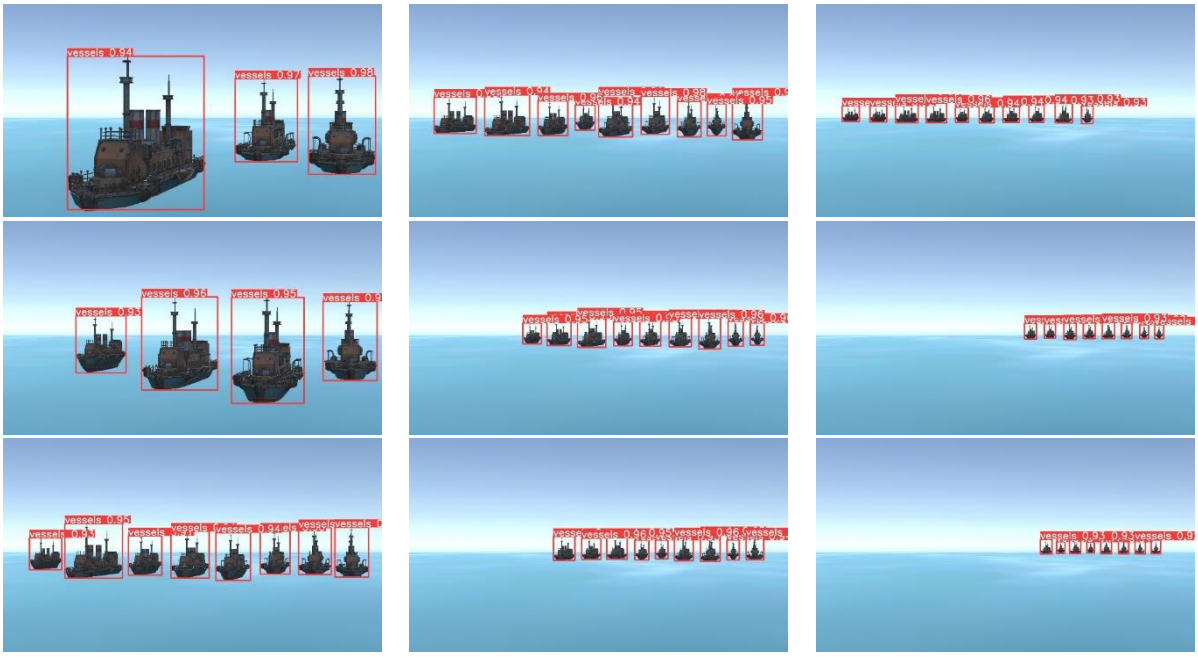
(g) 700~800m (h) 800~900m (i) 900~1000m

Fig. S5. Image samples of synthetic ships.

The image dataset comprised 210 training images, with 45 images designated for

validation and another 45 for testing. The adaptive moment estimation optimiser is employed with an initial learning rate of 0.001, a momentum of 0.937, and a weight decay of 0.0005. YOLOv8 is trained using a batch size of 32 for 300 epochs. Mean Average Precision (mAP) is used to evaluate the performance of the detection model, with higher mAP values indicating better detection accuracy. Specifically, mAP@0.5 refers to the mAP calculated at Intersection over Union (IoU) = 0.5, and mAP@0.5:0.95 denotes the mAP within the IoU range of (0.5, 0.95) with a step size of 0.05 [S4].

Under the aforementioned experimental conditions, YOLOv8 achieved mAP@0.5 and mAP@0.5:0.95 scores of 97.8% and 91.7%, respectively, on the synthetic image dataset. The detection results in Fig. S6 showcase that YOLOv8 provides complete and accurate bounding boxes for ships located within 1000 meters of the bridge in the virtual environment.

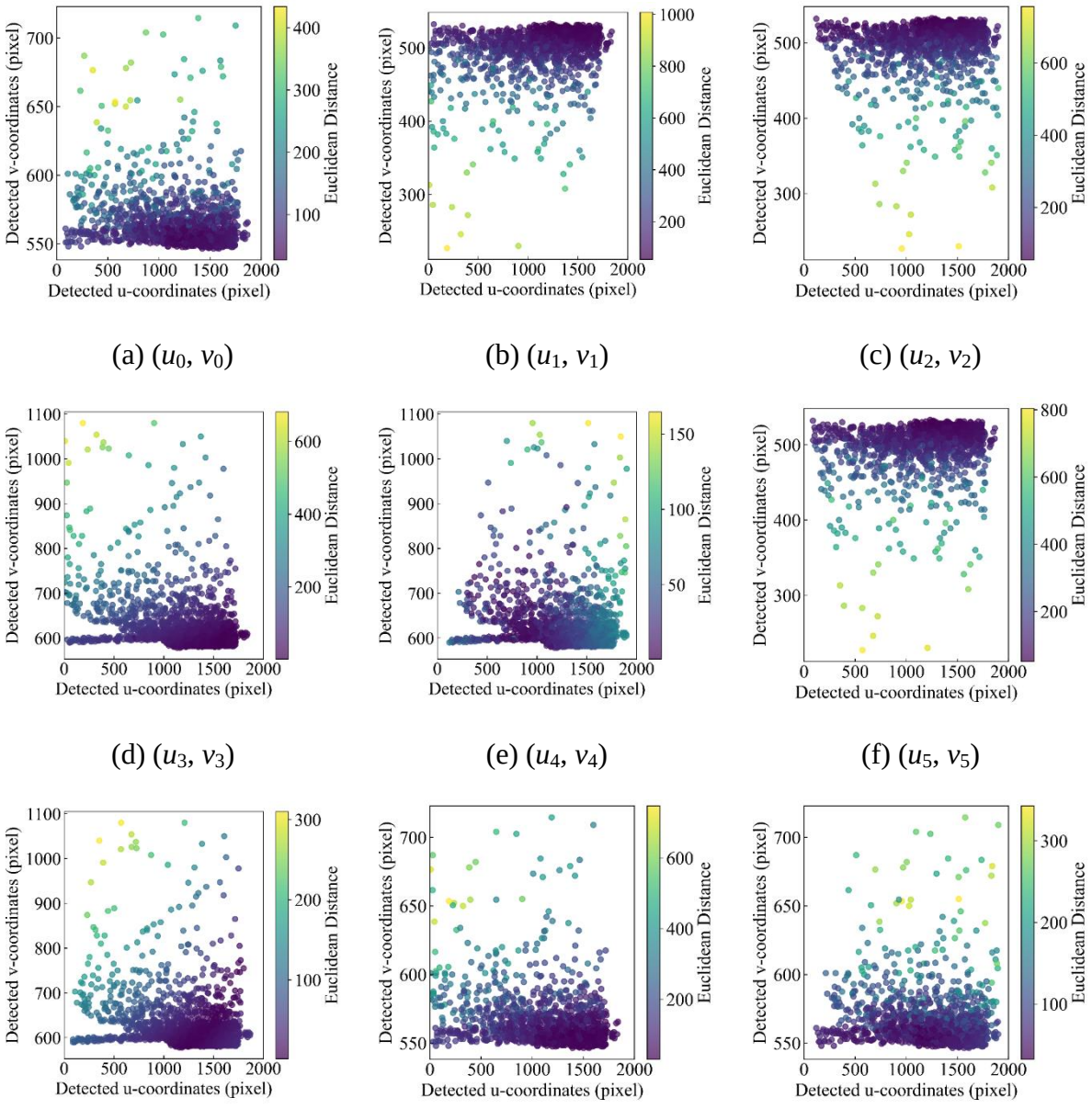


(a) 100~400m (b) 400~700m (c) 700~1000m

Fig. S6. Detection results from the YOLOv8.

Section S6. Euclidean distance distribution

Fig. S7 presents the scatter plot of the Euclidean distance distribution for the nine pixels following target matching.



(a) (u_0, v_0) (b) (u_1, v_1) (c) (u_2, v_2)

(d) (u_3, v_3) (e) (u_4, v_4) (f) (u_5, v_5)

(g) (u_6, v_6) (h) (u_7, v_7) (i) (u_8, v_8)

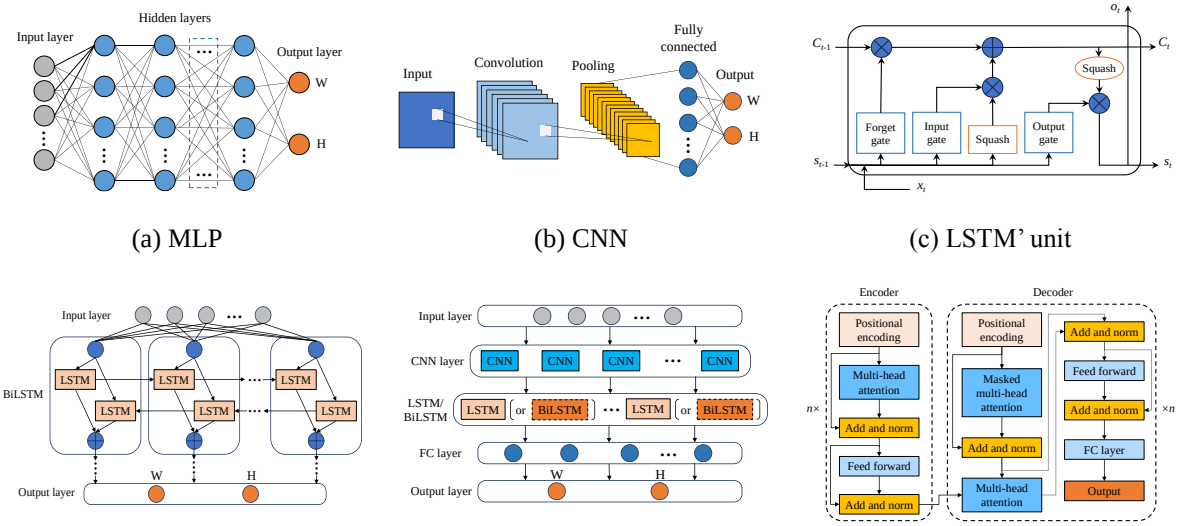
Fig. S7. Euclidean distance distribution for the nine pixels after object matching.

According to Fig. S7, for the extracted point (u_4, v_4), the Euclidean distance between each ship's corresponding P^* and P^o in the simulated environment is generally within 150

pixels, representing the smallest distance among all extracted points.

Section S7. Model configuration parameters and comparison

Seven benchmark models, including Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), CNN-LSTM, CNN-BiLSTM, and Transformer are selected for comparison [S5]. Their model architectures are illustrated in Fig. S8, and the corresponding hyperparameters are provided in Table S1.



(a) MLP (b) CNN (c) LSTM' unit

(d) BiLSTM (e) CNN-LSTM or CNN-BiLSTM (f) Transformer

Fig. S8. Seven benchmark DL models.

Table S1. Hyperparameters of typical benchmark models.

Models	Hidden layers	CNN layers	LSTM layers	BiLSTM layers	Encoders	Decoders	Multi-head attention	Neurons	Number of Parameters
MLP	5	/	/	/	/	/	/	640	87946
CNN	3	2	/	/	/	/	/	384	132160
LSTM	3	/	3	/	/	/	/	496	190630
BiLSTM	3	/	/	2	/	/	/	512	605218
CNN-LSTM	2	2	2	/	/	/	/	224	535306
CNN-BiLSTM	2	2	/	1	/	/	/	128	2316482

Transformer	2	/	/	/	6	6	8	164	9702274

The batch size for all models is set to 32, and each model is trained for 200 epochs. The solver used is Stochastic Gradient Descent (SGD) optimiser with an initial learning rate of 0.01 [S6]. 1080 ship samples are allocated for model training, while the remaining 270 ship samples are reserved for evaluating model performance.

Formulas for Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percent Error (MAPE), and coefficient of determination (R^2) are given by

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{S7}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{N}} \tag{S8}$$

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{S9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y_i}\right)^2} \tag{S10}$$

where y_i and $\hat{y}_i$ are the observed and predicted value of the ship size, respectively, $\overline{y_i}$ is the mean of y_i values, and N is the number of samples.

Fig. S9 describes the distribution of actual values, predicted values, and their relative errors for ship sizes across the test dataset for the seven models. It is evident that MLP exhibits the lowest measurement accuracy, mainly due to its simpler network structure. For the other models, the majority of points cluster near the 1:1 reference line, indicating high predictive accuracy.

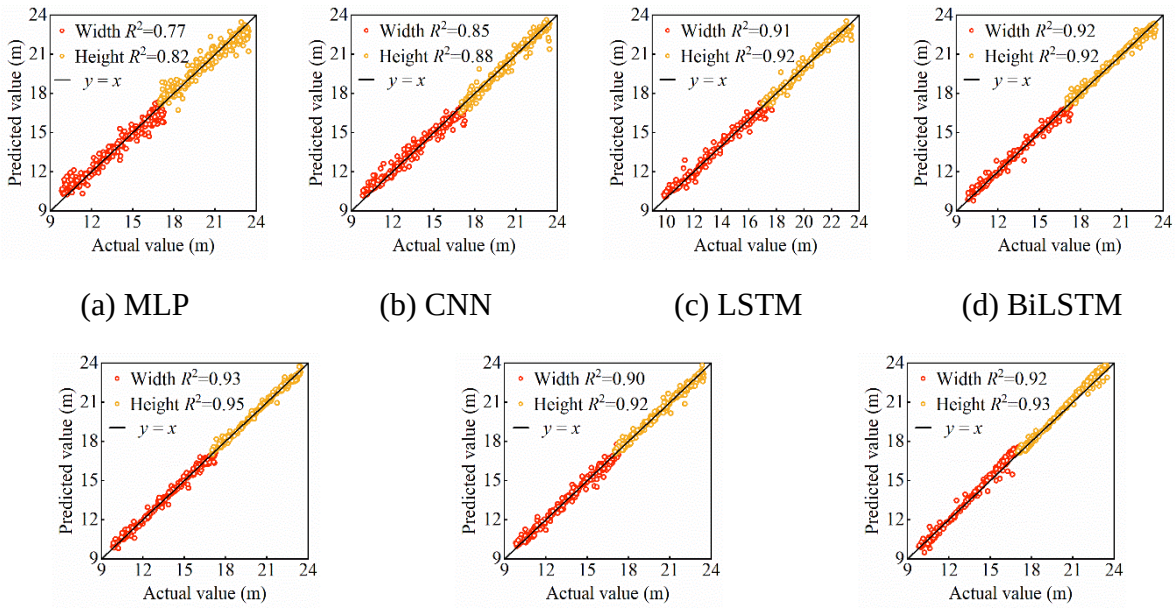
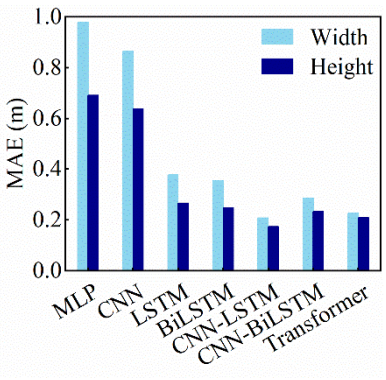
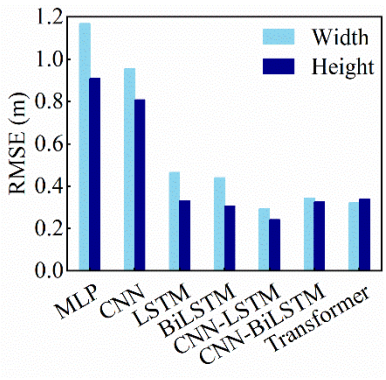


Fig. S9. The distribution of actual values, predicted values, and their relative errors for the different models

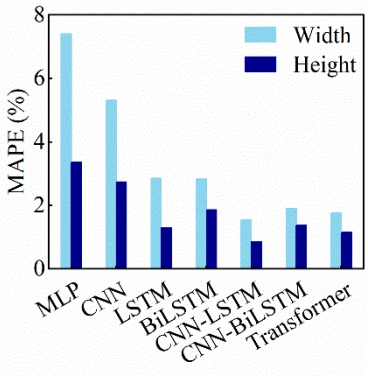
The MAE, RMSE, MAPE, and R^2 of these models are compared in Fig. S10. The values of MAE, RMSE, and MAPE for the ship height measurements are generally smaller than for the width measurements. This disparity arises from considering the height and width of the bounding box as the pixel dimensions of the ship in the image. Many ships appear in side views in images, leading to an overestimation of ship widths. However, with the increase in data volume and optimization of the model, this issue is expected to resolve. Additionally, whether measuring ship height or width, the CNN-LSTM model exhibits the lowest MAE, RMSE, and MAPE, and shows the highest R^2 value. Fig. S10 demonstrates that the CNN-LSTM model exhibits the best performance, which is why it is selected as the predictive model for virtual-to-real-world transfer learning.



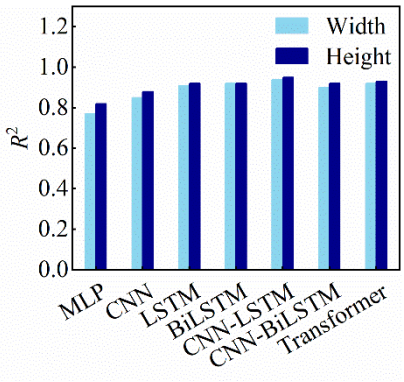
(a) MAE



(b) RMSE



(c) MAPE

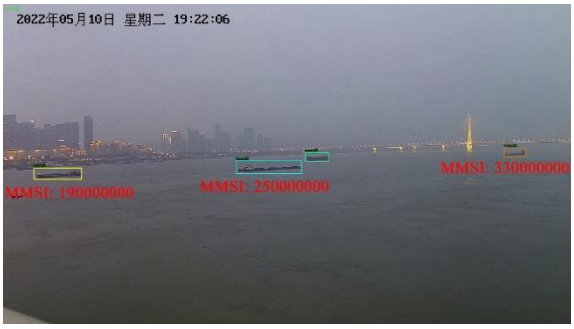


(d) R^2

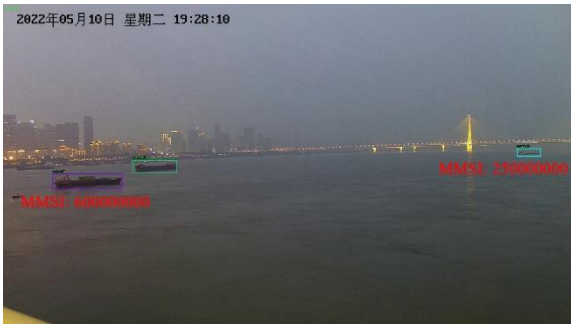
Fig. S10. The MAE, RMSE, MAPE, and R^2 of the seven models

Section S8. Description of the real-world dataset

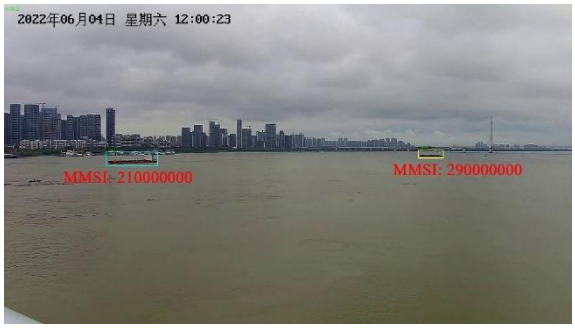
Ten ships with AIS appeared in the three sets of videos. The ship image samples are shown in Fig. S11. If the MMSI label is not shown, AIS information for that ship is unavailable.



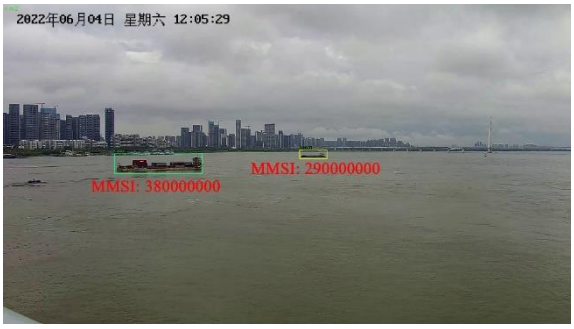
(a) The 130th frame of Video1



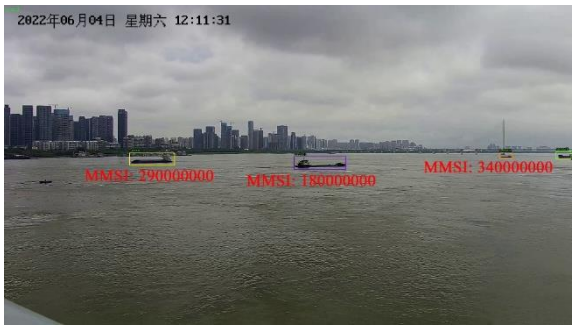
(b) The 455th frame of Video1



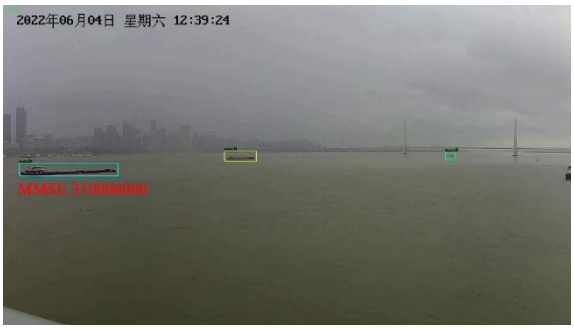
(c) The 65th frame of Video2



(d) The 390th frame of Video2



(e) The 780th frame of Video2



(f) The 65th frame of Video3

Fig. S11. Image samples extracted from the videos.

A total of 90 samples, each with MMSI, ship sizes, positions and bounding box information, are obtained due to the 1~2 minutes interval required for AIS data updates [S7]. These data cover a camera-to-ship distance range of 300m to 1000m [S8], with distances greater than 200m generally regarded as far-range perception [S9, S10]. Therefore, it is well-suited to validate the effectiveness of the proposed size measurement framework in overcoming the challenges of long-range monitoring. For each ship target, the bounding box information and latitude-longitude coordinates corresponding to the time when the AIS data first changed are presented in Table S2.

Table S2. Representative sample data used for transfer learning.

MMSI	Video	Time (s)	Bounding boxes		Ship positions		Ship sizes
			u_1 (pixel)	v_1 (pixel)	latitude (°E)	longitude (°N)	W(m)
250000000	Video1	62	1050	705	114.3262	30.6111	8

190000000	Video1	122	6	746	114.3211	30.6083	8
330000000	Video1	122	2373	640	114.3355	30.6191	11
600000000	Video1	482	782	732	114.3260	30.6105	8
210000000	Video2	62	460	666	114.3229	30.6108	10
290000000	Video2	62	1867	657	114.3306	30.6182	10
380000000	Video2	542	1075	673	114.3253	30.6096	13
340000000	Video2	722	1100	684	114.3241	30.6084	13
180000000	Video2	842	2341	656	114.3338	30.6204	8
310000000	Video3	62	791	696	114.3234	30.6083	9

Note: (u_1, v_1) and W have the same meanings as defined earlier. Only (u_1, v_1) is listed here; additional information of bounding boxes can be found at [S8].

To reduce dataset bias between simulated and real worlds, the pixel coordinates of all real-world ships are transformed into the virtual pixel coordinate system using Eq. (5), and the latitude-longitude coordinates of ships in the videos are converted into local coordinates in the virtual space using Eq. (S6). The latitude and longitude of the real-world camera are 114.3311 ° E and 30.6183 ° N, respectively. Other parameters used for the calculations can be found at [S8]. Table S3 lists the results of the transformed data from Table S2.

Table S3. The results of the transformed real-world data.

MMSI	Video	Time (s)	Bounding boxes		Ship positions		Ship sizes
			u_1 (pixel)	v_1 (pixel)	X_s (m)	Y_s (m)	W (m)
250000000	Video1	62	1725	1099	-95	1077	8
190000000	Video1	122	1606	1112	-54	1047	8
330000000	Video1	122	1474	1113	60	967	11
600000000	Video1	482	1344	1118	111	933	8

210000000	Video2	62	1234	1115	208	843	10
290000000	Video2	62	1153	1114	231	810	10
380000000	Video2	542	1074	1110	252	778	13
340000000	Video2	722	986	1104	311	689	13
180000000	Video2	842	893	1096	346	627	8
310000000	Video3	62	774	1093	383	567	9

Note: (u_1, v_1) and W have the same meanings as defined earlier. (X_s, Y_s) represents the position of the ships in the local coordinate system of the virtual space, as derived from the videos.

Section S9. Measurement results of the transfer learning model

The MAE, RMSE, MAPE, and R^2 of the fine-tuned model and the original CNN-LSTM are compared in Fig. S12. The transfer learning model exhibits lower MAE, RMSE, and MAPE values than the original CNN-LSTM. Additionally, the transfer learning model achieves an R^2 value of 0.90, significantly surpassing that of its original counterpart. This demonstrates its superior accuracy in measuring ship widths on the real-world data.

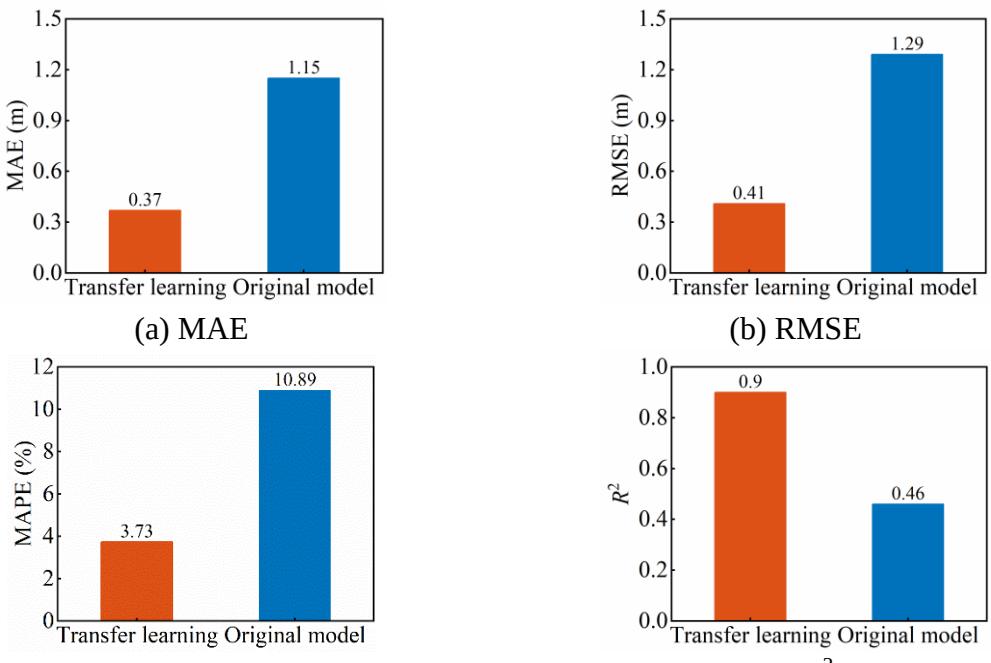


Fig. S12 The MAE, RMSE, MAPE, and R^2 of the two models

Table S4. Comparison between real ship widths and predicted ship widths.

Sample no.	Camera-to-ship distance (m)	W^* (m)	W (m)	Error (m)	Relative error (%)
1	762.83	11.18	11	0.18	1.64
2	562.91	8.24	8	0.24	3.00
3	988.97	9.48	10	-0.52	-5.2
4	741.93	13.50	13	0.50	3.85
5	904.91	9.60	10	-0.40	-4.00
6	426.56	13.43	13	0.43	3.31
7	573.67	10.55	10	0.55	5.5
8	879.72	11.36	11	0.36	3.27
9	480.24	11.50	11	0.50	4.55
10	696.31	8.66	9	-0.34	-3.78
11	379.18	8.41	9	-0.59	-6.56
12	935.12	8.49	8	0.49	6.13
13	816.62	12.81	13	-0.19	-1.46
14	696.54	11.46	11	0.46	4.18
15	556.76	9.71	10	-0.29	-2.90
16	563.98	7.82	8	-0.18	-2.25
17	656.07	9.54	10	-0.46	-4.60
18	765.53	8.89	9	-0.11	-1.22

Note: W^* and W represent the predicted width and the actual width, respectively.

References

[S1] Ma, H., Zhang, Y., Sun, S., et al., 2024. Weighted multi-error information entropy based you only look once network for underwater object detection. *Engineering Applications of Artificial Intelligence*, **130**: 107766. https://doi.org/10.1016/j.engappai.2023.107766

[S2] Xu, H.R., Yin, J.N., Zhang, N., 2025. Transformer-based deformation measurement of underground structures from a single-camera video. *Automation in Construction*, **172**: 106070. https://doi.org/10.1016/j.autcon.2025.106070

[S3] Zhou, Y., Leung, H., Blanchette, M., 1999. Sensor alignment with earth-centered earth-fixed (ECEF) coordinate system. *IEEE Transactions on Aerospace and Electronic systems*, **35**(2): 410-418. https://doi.org/10.1109/7.766925

[S4] Wang, J., Ma, X., Zhu, X., et al., 2025. Kinematic modeling and stability analysis for a wind turbine blade inspection robot. *Journal of Zhejiang University-SCIENCE A*, **26**(2): 121-137.
https://doi.org/10.1631/jzus.A2300619

[S5] Zhang, Y.M., Wang, H., 2023. Multi-head attention-based probabilistic CNN-BiLSTM for day-ahead wind speed forecasting. *Energy*, **278**: 127865.
https://doi.org/10.1016/j.energy.2023.127865

[S6] Ye, X.W., Zhang, X.L., Zhang, H.Q., et al., 2023. Prediction of lining upward movement during shield tunneling using machine learning algorithms and field monitoring data. *Transportation Geotechnics*, **41**: 101002.
https://doi.org/10.1016/j.trgeo.2023.101002

[S7] Pallotta, G., Vespe, M., Bryan, K., 2013. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy*, **15**(6): 2218-2245.
https://doi.org/10.3390/e15062218

[S8] Guo, Y., Liu, R.W., Qu, J., et al., 2023. Asynchronous trajectory matching-based multimodal maritime data fusion for vessel traffic surveillance in inland waterways. *IEEE Transactions on Intelligent Transportation Systems*, **24**(11): 12779-12792.
https://doi.org/10.1109/TITS.2023.3285415

[S9] Zhang, L., Chen, P., Li, M., et al., 2022. A data-driven approach for ship-bridge collision candidate detection in bridge waterway. *Ocean Engineering*, **266**: 113137.
https://doi.org/10.1016/j.oceaneng.2022.113137

[S10] Gargoum, S.A., Karsten, L., El-Basyouny, K., et al., 2018. Automated assessment of vertical clearance on highways scanned using mobile LiDAR technology. *Automation in Construction*, **95**: 260-274.
https://doi.org/10.1016/j.autcon.2018.08.015