TRANSNAV the International Journal on Marine Navigation and Safety of Sea Transportation
http://www.transnav.eu
Volume 19
Number 4
December 2025
DOI: 10.12716/1001.19.04.18

Automatic Detection of Navigational Signs on Inland Waterways Using YOLO Neural Networks

P. Adamski & J. Łubczonek
Maritime University of Szczecin, Szczecin, Poland

ABSTRACT: The study analysed the detection of navigation signs for inland navigation using YOLO neural networks. All major versions of the network available at the time of the study were analysed, i.e. from YOLO 1 to YOLO 12. The study considered two criteria: detection efficiency and detection accuracy. The first case is related to applications requiring the highest possible number of object detections, while the second is related to mapping tasks, where the accuracy of determining the location of a sign plays an important role. The results of the study showed that different efficiencies can be expected for the neural models studied. The latest models do not always prove to be the best. In terms of detection efficiency, the YOLO 4 network proved to be the best model, while in terms of sign detection precision, YOLO v7 had the highest horizontal accuracy and YOLO v10 had the highest vertical accuracy. The results of the study indicated that in some cases, attention should be paid to the oldest versions of YOLO, such as YOLO V1 and V2, which significantly reduce false detections.

1 INTRODUCTION

The rapid development of machine learning technology finds increasing application in waterborne transport, particularly in vessel navigation. Potential fields of applicability of AI methods can be seen in, among other things, vessel traffic prediction, detection or tracking [1-4]. Sensors acquiring image data are a special case in performing the above tasks. Digital images are ideal for automated acquisition of information about navigation objects, as they can be combined with deep learning methods to automate navigation processes. Autonomous vehicles play an important role in terms of target applicability [5,6], shaping the direction of development of intelligent solutions in shipping and often based on knowledge obtained through image processing. Appropriate implementation of artificial intelligence can significantly improve the safety and efficiency of water transport and prepare the ground for its autonomisation. It should be noted in this connection that the development of intelligent systems for IWT is now firmly on the agenda, with plans for their implementation extending to 2050 [7,8].

A special case of image processing applications based on deep learning is inland navigation, where navigation signs play a key role in the navigation process. Their main role is to ensure safety and regulate traffic rules on waterways. Misinterpretation of this type of information can lead to serious incidents, such as collisions with bridges - important pieces of infrastructure [9]. Navigation signs are used in the navigation process when operating a vessel and are one of the elements of IENC (Inland Electronic Navigational Chart). Hence, another potential application of neural networks could be systems for automatic mapping of navigation objects, where, in addition to the detection efficiency itself, its accuracy will be important. Automating this process could also

significantly speed up database updates [10] and streamline the process of creating the maps themselves [11], which often requires time-consuming indoor work. As can be seen, in general navigation applications, the most common use of neural networks will be object detection. For navigation object mapping systems, the best result achieved by a neural model will be both high detection and accuracy. This will make it possible to complete the largest number of objects as well as to determine their exact position later on.

This paper focuses attention on the use of YOLO convolutional networks, enabling real-time detection [12]. This type of network is suitable both for applications requiring ongoing sign detection (e.g. for automated situation assessment based on detected objects) and for mapping tasks. Due to the wide range of applicability of this type of network, a comparative analysis of all available major versions of the YOLO network was carried out. The research focused on identifying the most effective version, considering two aspects: detection efficiency and detection accuracy. The objects to be detected were standardised navigation signs for inland navigation [13].

As can be seen, the problem of detecting fairway markings has different requirements for the neural network depending on the use case of the vision system. A detection system for automated navigational analyses, in which the priority action will be to detect all the marks that are present in the fairway, requires that the TP (True Positive) score is as high as possible, even at the expense of a poorer FP (False Positive) score. In such cases, the need to verify data is often less of a problem for the user than omitting signs. However, the system responsible for mapping signs has different requirements. Such a system should be characterised by the highest possible indication of IoU (Insertion Over Union), which will allow the exact position of the mark to be determined. In this case, the user knows that the mark is already in the image, so the task of the neural network is to indicate the position of the object being searched for as accurately as possible. With such a system, it is also desirable to achieve as low an FP as possible, to reduce the need for manual filtering of data after measurements have been taken.

2 METHODOLOGY

The research was conducted in two stages, related to the assessment of the effectiveness of navigation marking detection. In the first stage, the effect of YOLO network type on object detection performance was analysed. The following measures were used: TP (True Positives), FP (False Positives), FN (False Negatives), Precision, Recall, F1-score and average IoU (Intersection over Union). The changes in Precision, TP and FP parameters were then analysed as a function of the IoU threshold. These values were examined in a range from 0%, meaning that the detection has at least one point in common with the label, to a value of 99%, where this value means that the common area is almost identical to the sum of the detection and label areas. The aim of this step was to generally evaluate YOLO-type neural networks in the context of detecting inland waterway markings. In this case, the desired effect of the neural network model is to maximise the number of detected signs (TP) and the quality of detection, i.e. the highest possible IoU.

The second stage of the research involved analysing the precision of sign detection. The deviation of the centre point of detection in the vertical and horizontal planes from the position of the centre of the object marked in the image in proportion to the length or width of the object marked in the image, respectively, was used as a measure of the precision of the detection position. The calculation of the precision of the object's position is shown in formulas 1 and 2.

$$\delta_w = \frac{\Delta x_c}{w_l} \cdot 100\%, \quad \delta_h = \frac{\Delta y_c}{h_l} \cdot 100\%$$

where:
δ_w - horizontal precision of the detected object.
δ_h - vertical precision of the detected object.
Δx_c - horizontal distance of the centre point of the detected object from the centre point of the label.
Δy_c - vertical distance of the centre point of the detected object from the centre point of the label.
w_l – label width
h_l - label height

2.1 *Data*

A total of 3269 images were prepared, with resolutions ranging from 1920x1080 to 1920x1440. The photographs will show inland navigation signs, observed both from the quay and from the water. The images show signs against a variety of backgrounds, including bridges, buildings, coastal infrastructure features and vegetation. Each photo contains between 1 and 15 manually labelled signs. The training collection contains 2692 images, the validation collection 194 images and the test collection 384 images. The photos show the photographed beacons between 2008 and 2022 on the Odra River. Exemplary photo is illustrated in fig. 1.

Figure 1. Example of a sign photo taken in 2009, West Oder River, city of Szczecin

2.2 *Models*

The models were trained using the Darknet [14] and Ultralytics [15] frameworks. The models: YOLOv1 [16], YOLOv2 [17], YOLOv3 [18], YOLOv4 [19] and YOLOv7 [20] were trained using configuration files and software in the latest version of the Darknet framework developed by Bochkovskii A. [21]. The models were trained in the Ultralytics framework,

YOLOv5 [22], YOLOv6 [23], YOLOv8 [24], YOLOv9 [25], YOLOv10 [26], YOLOv11 [27], YOLOv12 [28], used the medium profile, labelled M, whose input size is 640x640. All models were trained for 300 epochs, using a default set of hyperparameters provided by the framework and model configuration file. The models used in the study, together with the input size, are summarised in table 1.

Table 1. Models used in the study

Model name	Framework	Input size
YOLOv1	Darknet	416 x 416
YOLOv2	Darknet	416 x 416
YOLOv3	Darknet	416 x 416
YOLOv4	Darknet	608 x 608
YOLOv5 M	Ultralytics	640 x 640
YOLOv6 M	Ultralytics	640 x 640
YOLOv7	Darknet	640 x 640
YOLOv8 M	Ultralytics	640 x 640
YOLOv9 M	Ultralytics	640 x 640
YOLOv10 M	Ultralytics	640 x 640
YOLO 11 M	Ultralytics	640 x 640
YOLO 12 M	Ultralytics	640 x 640

2.3 *Hardware and software*

Model training was carried out on a desktop computer with the configuration: Intel Core i9 14900K processor, RAM: 128 GB DDR5 5400 MHz, graphics card: Nvidia RTX 4090 24GB. Ubuntu 24.04 LTS was used as the operating system and the drivers for the graphics card - version 550.107.02 - were installed, along with the CUDA framework [29] version 12.4. The Darknet framework has been compiled from source code with support for GPU computing in the CUDA framework. The Ultralytics framework based on PyTorch [30] has the ability to train using Nvidia GPUs as standard.

3 RESULTS

The first stage of the study analysed what the results of standard neural network metrics such as the number of TPs, FPs, FNs, Presicion, Recall, F1-score and mean IoU were for the YOLO models at the settings: confidence threshold: 0.5, NMS threshold: 0.4, IoU threshold 0.5. The results are presented in Table 2.

Table 2. Results obtained for the different versions of YOLO.

Model	TP	FP	FN	Precision	Recall	F1-score	Average IoU
(darknet) YOLO v1	456	8	405	0,98	0,53	0,69	0,80
(darknet) YOLO v2	461	20	400	0,96	0,54	0,69	0,80
(darknet) YOLO v3	750	42	111	0,95	0,87	0,91	0,83
(darknet) YOLO v4	810	17	51	0,98	0,94	0,96	0,89
(darknet) YOLO v7	776	14	85	0,98	0,90	0,94	0,88
(ultralytics) YOLO v5	768	22	93	0,97	0,89	0,93	0,89
(ultralytics) YOLO v6	757	29	104	0,96	0,88	0,92	0,89
(ultralytics) YOLO v8	763	22	98	0,97	0,89	0,93	0,89
(ultralytics) YOLO v9	771	29	90	0,96	0,90	0,93	0,89
(ultralytics) YOLO v10	768	19	93	0,98	0,89	0,93	0,89
(ultralytics) YOLO v11	776	31	85	0,96	0,90	0,93	0,89
(ultralytics) YOLO v12	772	35	89	0,96	0,90	0,93	0,89

The highest TP score was achieved by the YOLO v4 model, which correctly detected 810 objects, achieving a similar precision score to the YOLO v1, YOLO v7 and YOLO v10 models. The differences in the precision of the models are small, ranging from 0.95 to 0.98. Significantly greater differences can be observed in the Recall and F1-score indicators. Recall for all networks ranged from 0.53 to 0.94. Clearly worse Recall scores were obtained by the oldest models YOLOv1 - 0.53 and YOLOv2 - 0.54, while the scores of all the others were similar and ranged from 0.87 to 0.94. The F1-score results are similar, with the YOLOv1 and YOLOv2 models performing noticeably worse (at 0.69), while the results of the other models ranged from 0.91 to 0.96. The highest F1-score was achieved by the YOLOv4 model. The average IoU for all networks ranged from 0.8 to 0.89, with only the score of models YOLOv1 to YOLOv3 slightly underperforming compared to the others. The lowest FP score was achieved by model YOLOv1, which detected 8 unwanted objects.

It was also analysed how precision varies with the IoU Threshold set, with confidence threshold: 0.5 and NMS threshold: 0.4, the results are presented in the graph in Figure 2.

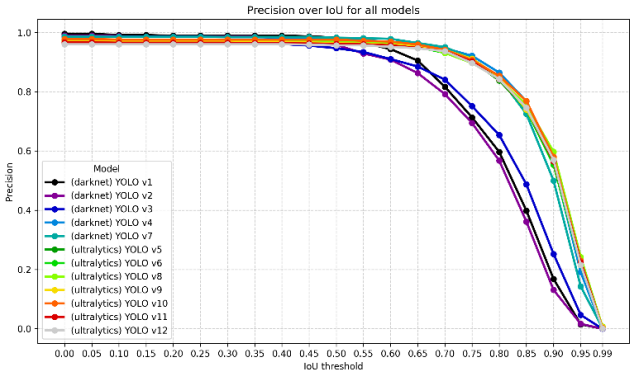

Figure 2. Graph of the dependence of the Precision indicator on the set IoU Threshold

For IoU threshold values between 0 and 0.5, all models achieved similar precision. Only above an IoU threshold of 0.5 do the models YOLOv1, YOLOv2 and YOLOv3 begin to show significantly lower precision.

Figure 3 shows the dependence of TP on IoU threshold. The model with the best characteristics is YOLOv4, which achieves a significantly higher TP than the other models in the range from 0.0 to 0.85 IoU Threshold. The YOLO v3 model in the IoU Threshold range from 0 to 0.45 achieves a similar result to the other models, but above this range it shows a significantly lower TP.

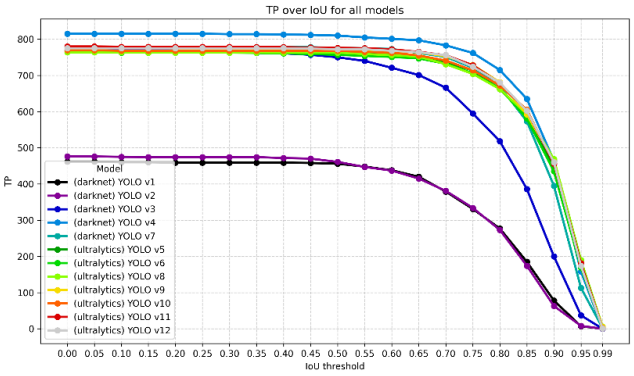

Figure 3. Diagram of the dependence of the TP indicator on the IoU threshold.

The dependence of FP on IoU threshold was also tested (Figure 4). The YOLO v1 and YOLO v2 models proved to be interesting cases. For IoU threshold values between 0.0 and 0.5, where they showed the

lowest amount of FP, while for IoU threshold values between 0.55 and 0.9 they showed a significant increase in FP, clearly greater than most of the other models.

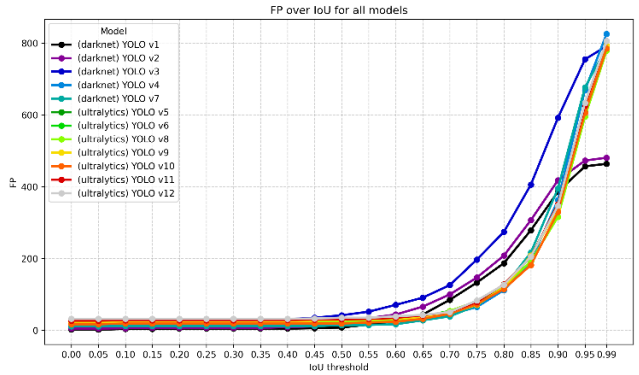

Figure 4. Graph of the dependence of the FP indicator on the IoU threshold.

It follows that these models detect the position of objects in the image with significantly lower accuracy than newer models such as YOLOv7, YOLOv4, and all models in the Ultralytics framework. Only the YOLO v3 model performs worse than YOLO v1 and YOLO v2 for a high (above 0,5) IoU threshold. An example of detection accuracy is shown in Figure 5.

Figure 5. Detection results for the YOLO v1 (left) and YOLOv7 (right) networks. Red shows the prediction made by the model, blue the ground truth.

The second stage of the research involved analysing the precision of the prediction centre point. The measurement of the precision of the prediction centre point indicates that the first three models – YOLOv1, YOLOv2 and YOLOv3 – are characterised by significantly lower prediction precision compared to newer versions. In the case of these models, lower precision in the horizontal axis than in the vertical axis is also noticeable. The remaining models achieved very similar results. These data are presented in Figure 6.

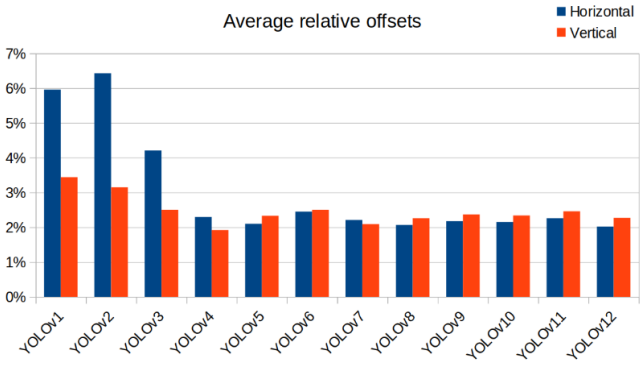

Figure 6. Average relative precision of prediction centre determination

Table 3 presents the values of the average and maximum relative prediction offset in relation to its size for all analysed models. The largest maximum shift in the horizontal axis was observed for the YOLO v6 - 108.4% and YOLO v11 – 105.9 % models. These models obtained a maximum error significantly higher than the others. The lowest maximum horizontal position error was achieved by the YOLO v7 model, which was 22.2%. The YOLO v1, YOLO v2 and YOLOv3 models achieved lower horizontal position precision than the other models, at 6%, 6.4% and 4.2% respectively. The other models achieved similar precision, ranging from 2% to 2.5%. The average vertical position measurement results for the YOLO v1, YOLO v2, and YOLO v3 models were less different from the other models, at 3.4%, 3.2%, and 2.5%, respectively, while all results ranged from 1.9% to 3.4%.

Table 3. Values of the mean and maximum relative offset of the prediction relative to its size for all models analysed

Model	Average Horizontal Offset	Maximum Horizontal Offset	Average Vertical Offset	Maximum Vertical Offset
YOLOv1	6,0%	60,7%	3,4%	61,3%
YOLOv2	6,4%	75,2%	3,2%	24,0%
YOLOv3	4,2%	52,6%	2,5%	70,3%
YOLOv4	2,3%	50,0%	1,9%	27,5%
YOLOv5	2,1%	30,5%	2,3%	32,4%
YOLOv6	2,5%	108,4%	2,5%	81,2%
YOLOv7	2,2%	22,2%	2,1%	68,5%
YOLOv8	2,1%	54,8%	2,3%	19,1%
YOLOv9	2,2%	83,3%	2,4%	38,2%
YOLOv10	2,2%	64,5%	2,3%	15,9%
YOLOv11	2,3%	105,9%	2,5%	120,7%
YOLOv12	2,0%	36,1%	2,3%	19,6%

In addition, the results of maximum shifts are presented in Figure 7. The lowest maximum errors in determining the vertical position were obtained by the YOLO v8 and YOLO v12 models, with offset values of 19.1% and 19.6%, respectively. The worst result was achieved by the YOLOv12 model – 120.7%.

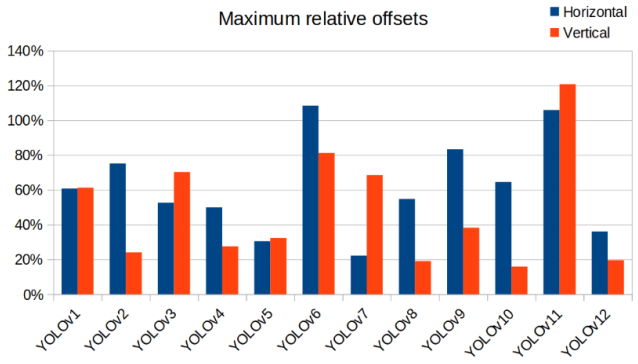

Figure 7. results of maximum shifts of the prediction centre point

A maximum prediction centre point shift of more than 100% means that the prediction centre is further away than the dimension of the label applied to the image. Such a case is presented in Figure 8, where a horizontal and vertical shift of the centre point of prediction by a value greater than the width and height of the label applied to the image was observed.

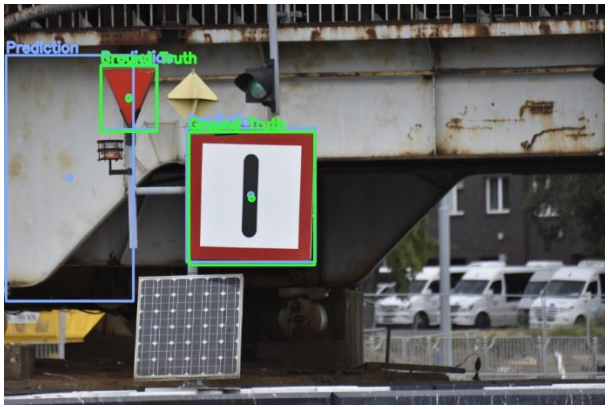

Figure 8. YOLO v11 prediction with vertical and horizontal shift of more than 100%

4 DISCUSSION

The choice of the best model may depend on the specific task. In the case where the main problem is to detect all objects, e.g. during automated navigation analyses, the best YOLO model with this training and test data set turned out to be YOLOv4, which had the highest TP score - a model that is already quite old, given how many newer models are currently available. For a task where the most important element is to reduce the number of erroneous detections required for subsequent manual filtering, YOLOv1, the first YOLO network model to achieve an FP of 8, proved to be the best. However, if the most important task performed by the network is to accurately indicate the position of the searched object in the image, then YOLOv10 proved to be the best model, achieving the highest precision at an IoU threshold of 0.95. This means that this model most accurately indicates the position of the searched objects. The difference between the models from YOLO v4 to YOLO v12 is not significant, and only the oldest models, YOLO v1, YOLO v2 and YOLO v3, deviate from these results.

Newer models are not always better in terms of the quality of the data they provide – this is particularly evident in the results of the YOLOv4 and YOLOv7 models trained in the Darknet framework. These models can still be useful in many applications, and upgrading to a newer model is not always justified. The oldest models, YOLOv1 and YOLOv2, lag significantly behind the others, especially in terms of the number of TP objects detected, but they can still be useful in certain specific cases. The latest available model, YOLOv12, achieved good results, although in many tasks it proved to be inferior to older models.

Determining the centre point of an object in an image may require the use of a model that is closely tailored to the specific task. For example, in systems whose sole purpose is to determine the horizontal position of objects, YOLOv5 or YOLOv7 models may be the appropriate choice. In the measurements carried out, they not only achieved a low average error in determining the centre point of prediction, but also the lowest values of maximum shifts in the horizontal axis.

In contrast, for applications requiring precise vertical positioning - such as measuring the height of signage location relative to the observer - the YOLOv8 model would be the best choice. It showed the lowest maximum error in the vertical position of the prediction centre point, while also having a low average error.

5 CONCLUSIONS

Older versions of the YOLO models, such as YOLOv4 and YOLOv7, despite the advent of newer architectures, still prove to be useful in the context of detection of fairway markings. The results obtained by these models are at a level comparable to the newer models.

The oldest models – YOLOv1 and YOLOv2 – differ in terms of quality from newer models. Their architectures are outdated compared to current standards, which means that they do not perform well in detecting objects in most cases. Their low precision and limited ability to locate signs make them unsuitable for most applications in navigation sign detection. In certain specific cases, the oldest models – YOLO v1 and YOLO v2 – can be useful when the essence of the problem is to limit FP.

The highest precision in determining horizontal position was achieved by the YOLO v7 model, while the highest precision in determining vertical position was achieved by the YOLO v10 model.

Models from YOLO v4 to YOLO v12 achieved similar results of average precision in determining the centre point of an object, the choice of the appropriate model in this case can be made based on other factors.

In terms of the number of objects detected in the image, the YOLOv4 model achieved the best results. This model is best suited for tasks where it is necessary to detect as many navigation signs as possible.

REFERENCES

[1] Azimi, S.; Salokannel, J.; Lafond, S.; Lilius, J.; Salokorpi, M.; Porres, I. A Survey of Machine Learning Approaches for Surface Maritime Navigation. 2020. Available online: http://hdl.handle.net/2117/329714 (accessed on 12 February 2025).

[2] Donandt, K.; Böttger, K.; Söffker, D. Short-Term Inland Vessel Trajectory Prediction with Encoder-Decoder Models. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 974–979. https://doi.org/10.1109/ITSC55140.2022.9922148.

[3] Agorku, G.; Hernandez, S.; Falquez, M.; Poddar, S.; Amankwah-Nkyi, K. Traffic Cameras to Detect Inland Waterway Barge Traffic: An Application of Machine Learning, Computer Vision and Pattern Recognition. arXiv 2024, arXiv:2401.03070.

[4] Hart, F.; Okhrin, O.; Treiber, M. Vessel-Following Model for Inland Waterways Based on Deep Reinforcement Learning. Ocean. Eng. 2023, 281, 114679. https://doi.org/10.1016/j.oceaneng.2023.114679.

[5] Vanneste, A.; Vanneste, S.; Vasseur, O.; Janssens, R.; Billast, M.; Anwar, A.; Mets, K.; De Schepper, T.; Mercelis, S.; Hellinckx, P. Safety Aware Autonomous Path Planning Using Model Predictive Reinforcement Learning for Inland Waterways. In Proceedings of the IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 17–20 October 2022; pp. 1–6. https://doi.org/10.1109/IECON49645.2022.9968678.

[6] Qiao, Y.; Yin, J.; Wang, W.; Duarte, F.; Yang, J.; Ratti, C. Survey of Deep Learning for Autonomous Surface Vehicles in Marine Environments. IEEE Trans. Intell. Transp. Syst. 2023, 24, 3678–3701. https://doi.org/10.1109/TITS.2023.3235911.
[7] Hao, G.; Xiao, W.; Huang, L.; Chen, J.; Zhang, K.; Chen, Y. The Analysis of Intelligent Functions Required for Inland Ships. J. Mar. Sci. Eng. 2024, 12, 836. https://doi.org/10.3390/jmse12050836.
[8] Li, Y.; Hu, Y.; Rigo, P.; Lefler, F.E.; Zhao, G.; Eds. Proceedings of PIANC Smart Rivers 2022: Green Waterways and Sustainable Navigations; Lecture Notes in Civil Engineering; Springer: Singapore, 2023; Volume 264. https://doi.org/10.1007/978-981-19-6138-0.
[9] Fan, W.; Zhong, Z.; Wang, J.; Xia, Y.; Wu, H.; Wu, Q.; Liu, B. Vessel-Bridge Collisions: Accidents, Analysis, and Protection. China Journal of Highway and Transport. 2024, 37(5), 38–66.
[10] Łubczonek, J., i M. Włodarczyk. Wykorzystanie geobazy danych w procesie tworzenia elektronicznych map nawigacyjnych dla żeglugi śródlądowe [Application of geodatabase in the process of creation electronic navigational charts for inland shipping]. Archiwum Fotogrametrii, Kartografii i Teledetekcji, t. 21, 2010, s. 221–34.
[11] Łubczonek, J. Opracowanie i implementacja elektronicznych map nawigacyjnych dla systemu RIS w Polsce [Elaboration and implementation of electronic navigational charts for RIS in Poland]. Roczniki Geomatyki 2015, 13, 359–368.
[12] Adamski, P.; Lubczonek, J. A Comparative Analysis of the Usability of Consumer Graphics Cards for Deep Learning in the Aspects of Inland Navigational Signs Detection for Vision Systems. Appl. Sci. 2025, 15, 5142. https://doi.org/10.3390/app15095142
[13] SIGNI. European Code for Signs and Signals on Inland Waterways: Resolution No. 90; United Nations: New York, NY, USA, 2018.
[14] Redmon J. Darknet: Open Source Neural Networks in C. Online source: https://pjreddie.com/darknet. Access date 13.07.2025
[15] Jocher G.; Qiu J.; Charasia A. Ultralyiics YOLO. Online source: https://github.com/ultralytics/ultralytics Access date:13.07.2025
[16] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You only look 1182 once: Unified, real-time object detection. In Proc. IEEE Conf. Comput (Vol. 1183, pp. 779-788).
[17] Redmon, J., Farhadi, A. YOLO9000: Better, Faster, Stronger. Proceedings of the IEEE conference on computer vision and pattern recognition
[18] Farhadi A., Redmon J. "Yolov3: An incremental improvement." In Computer vision and pattern recognition, vol. 1804, pp. 1-6. Berlin/Heidelberg, Germany: Springer, 2018.
[19] Bochkovskiy, Alexey & Wang, Chien-Yao & Liao, Hong-yuan. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. 10.48550/arXiv.2004.10934.
[20] Wang, C., Bochkovskiy, A., & Liao, H. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2207.02696
[21] Bochkovskiy, A. Open Source Neural Networks in C. Online source: https://github.com/AlexeyAB/darknet Access date:13.07.2025
[22] Khanam, R., & Hussain, M. (2024). What is YOLOv5: A deep look into the internal features of the popular object detector. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2407.20892
[23] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X. (2022). YOLOV6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2209.02976
[24] R. Varghese and S. M., "YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533619.
[25] Wang, C., Yeh, I., & Liao, H. M. (2024). YOLOV9: Learning what you want to learn using programmable gradient information. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2402.13616
[26] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). YOLOV10: Real-Time End-to-End Object Detection. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2405.14458
[27] Khanam, R., & Hussain, M. (2024b). YOLOV11: An overview of the key architectural enhancements. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2410.17725
[28] Tian, Y., Ye, Q., & Doermann, D. (2025). YOLOV12: Attention-Centric Real-Time Object Detectors. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2502.12524
[29] Nvidia Corporation. CUDA Toolkit Documentation 12.4 Available online: https://docs.nvidia.com/cuda/archive/12.4.0/ (accessed on 13 July 2025).
[30] Paszke, A.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv 2019, arXiv:1912.01703. Available online: http://arxiv.org/abs/1912.01703 (accessed on 12 February 2025)