Search and classification of objects in the zone of reservoirs and coastal zones

Viktorija M. Smolij[1], Natan V. Smolij[2] and Sergii P. Sayapin[1]

[1]National University of Life and Environmental Sciences of Ukraine, 15 Heroyiv Oborony Str., Kyiv, 03041, Ukraine
[2]Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Beresteiskyi Ave., Kyiv, 03056, Ukraine

Abstract

A minimal working version of the computer vision subsystem has been developed specifically for deployment on a research unmanned aerial vehicle (UAV). This subsystem focuses on detecting specific objects present on the surfaces of water bodies and subsequently classifying them. The effectiveness of this subsystem was evaluated by comparing two state-of-the-art models, YOLOv5 and YOLOv8, to determine their suitability for addressing the target problem. To evaluate performance of the resulted model's series of test was performed. It resulted in achieving desired output of object detections but with low accuracy of classification, however such systems can be used as wider-area object detector. According to the obtained results, it can be seen that the system detects objects on the water surface, but the classification of these objects is not good. There are several reasons for this: errors in the labeling of the dataset and the small size of the dataset. A possible scenario of using the built model is the general collection of information about the reservoir without regard to the classification output. In the process of such exploitation, it can be considered as expedient to collect a dataset that will correspond to the data from the drone (the data of the current dataset is data from surveillance cameras and video recordings from boats). In the future, form the dataset according to the developer's requirements, applying the necessary data augmentation steps.

Keywords

dataset, model, image distribution, confusion matrix, training metrics, augmentation, mosaic placement of images

1. Introduction

In the modern world of robotics, many tasks require the intervention of artificial intelligence to increase the number of tasks to be solved [1, 2, 3, 4], increase productivity, reduce execution time, scale processing, exclude a person from the process of performing routine tasks, and ensure online information collection and processing processes [5, 6, 7, 8]. So, for example, creating maps and patrolling water bodies using traditional methods is a time-consuming and time-consuming process that can be improved and accelerated with the help of artificial intelligence [9, 10, 11].

In the modern period of development of unmanned aerial vehicles comes the realization that many tasks of research and observation can be transferred to automated drones, which will perform them faster and better due to the possibility of installing additional computing power as a payload [12, 13, 14, 15]. This approach is also supported by the fact that flight controllers available on the market, such as Betaflight, Pixhawk, etc. [16, 17], have a wide range of interfaces for communicating with external devices, exchanging telemetry information, camera data and other interesting data sets that can be grouped into datasets for automation management and debugging processes [18, 19].

Computer vision systems are technologies that give computers the ability to recognize and analyze visual data [20]. The structure of a computer vision system is usually complex and depends on the specific task and the technologies used [21, 22]. However, generally speaking, a computer vision system can be divided into several key components: data collection, data pre-processing, feature summarization, recognition and classification, decision-making process, presentation of results, etc.

doors-2024: 4th Edge Computing Workshop, April 5, 2024, Zhytomyr, Ukraine
vmsmolij@nubip.edu.ua (V. M. Smolij); hoibbitizukrainy@gmail.com (N. V. Smolij); sayapin@nubip.edu.ua (S. P. Sayapin)
https://nubip.edu.ua/node/3062 (V. M. Smolij); https://www.facebook.com/profile.php?id=61553465353363 (N. V. Smolij); https://nubip.edu.ua/node/3062 (S. P. Sayapin)
0000-0002-1268-7837 (V. M. Smolij); 0009-0002-3763-6726 (N. V. Smolij); 0000-0003-1565-4034 (S. P. Sayapin)

Unmanned aerial vehicles (UAVs) have various structural elements that determine their functionality and characteristics. The main structural elements of a UAV include: fuselage (body), wings, tail unit, engines, connecting elements, equipment for filming and observation (cameras, sensors). The variety of tasks for the use of UAVs in water and coastal zones illustrate the importance and relevance of the conducted research for various fields of application, including full-scale war on the territory of Ukraine, customs, security, monitoring, ecology and natural science. The purpose of the research is to create a minimum working version of the computer vision subsystem for use on a research UAV and to provide instructions for further improvement of the system and its development.

The feasibility of using AI is due to the fact that there is no clear algorithm for detecting objects on the water surface using image processing methods other than AI. Also, the use of UAVs in combination with AI will allow processing data from large areas of the earth and water surface, which will improve the response to emergency situations with the use of a limited number of human resources [23].

For task of object detection in the image, there are a large number of software solutions that allow you to construct and train a neural network. Examples of such solutions are tensorflow, pytorch, theano, ultralytics, chainer libraries. Since the task of creating a dataset is part of the usual functionality of the libraries, the range of possible options is narrowed to the ultralytics API, which is less flexible in terms of model selection, but provides a wide functionality for working with data. To perform the given task, it is most appropriate to use the Ultralytics API, as they provide the necessary functionality for dataset synthesis and provide interfaces for programming the training of a wide range of models for object detection. The software is written in the Python programming language due to its dynamic typing and automatic garbage collection, as well as a port of the above API for this language.

2. A model of an artificial intelligence system

2.1. Creating a dataset

The subsystem will control the drone, which must move along the route at the points specified by the user, and be able to detect such objects as boats, ships, buoys, garbage islands, swimmers and drowning people from the image from the camera.

The dataset is under development and is a compilation from several data sources: https://universe.roboflow.com/hamdi-ali/plastic-pollution-ugslg, https://universe.roboflow.com/double-o-co-ltd/marine-object-detection-yjybm/dataset/4, https://universe.roboflow.com/pwnface4-gmail-com/drowning-people. The general principles, application, versatility and features of use are given in [24, 25, 26, 27, 28]

Model definition: since it is planned to implement the model on a Raspberry microcomputer, 2 possible models for training can be distinguished, YOLOv5 due to its small size and YOLOv8 due to the fact that with approximately the same number of parameters as YOLOv5, the model gives a better result, as shown in figure 1.

In the figure 1, the number of parameters is shown on the abscissa axis, and the effectiveness is shown on the ordinate axis.

The following are examples of the considered images. The “Boat” object class is shown in figure 3.

The object class “Ship” is shown in figure 4.

The object class “Buoy” is given on figure 5.

The class of objects “Swimmer” is shown in figure 6.

The object class “Drowning man” is shown in figure 7.

The distribution of images by classes is shown in figure 8.

The largest number of markings belongs to the boat class (1629 units). Along with this class, the following classes are well represented (in descending order): swimmer, boat and buoy, respectively 1498, 1270 and 1186. The garbage and drowning man classes contain the least number of images, which can lead to training anomalies. The distribution of images between datasets is shown in figure 9.

This distribution is due to the fact that for the available 6,000 images, the test data set will consist of three hundred images, which is more than enough to test the performance of the model.

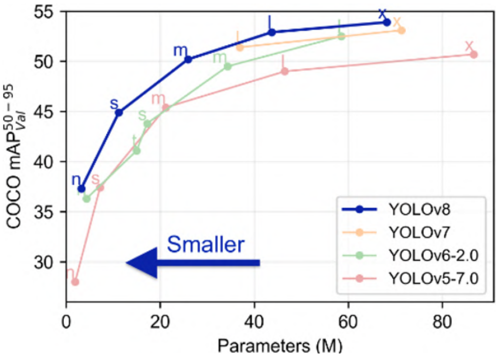


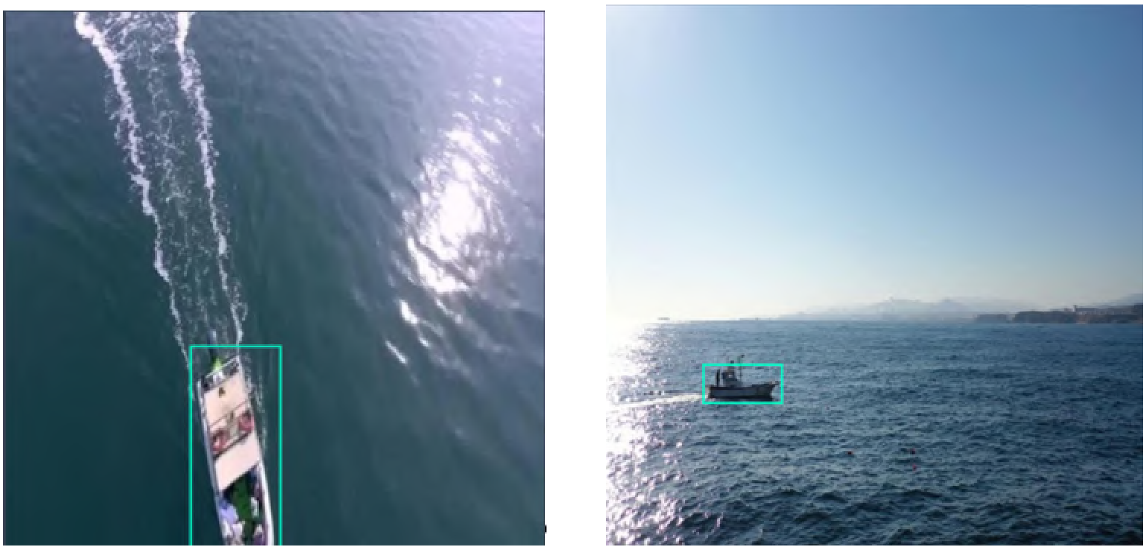
Figure 1: Average accuracy on the dataset common objects in context depending on the model used and the number of trained parameters.

Performance Comparison of YOLOv8 vs YOLOv5

Model Size	Detection*	Segmentation*	Classification*
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

*Image Size = 640 *Image Size = 224

Figure 2: Comparison of performance of YOLO (you only look once) v8 vs YOLOv5 models in detection, segmentation and classification tasks.



(a) (b)

Figure 3: Examples of images of the “Boat” object class, (a) – top view and (b) – front view.

The selection of these classes for the dataset was due to the fact that it allows covering a large part of the objects of maritime navigation and interaction. The addition of human images to the dataset is also due to the fact that the deployable drone can be used as a rescue drone and add the functionality of calling rescuers or providing assistance: lifebuoys or vests can be attached to the drone.

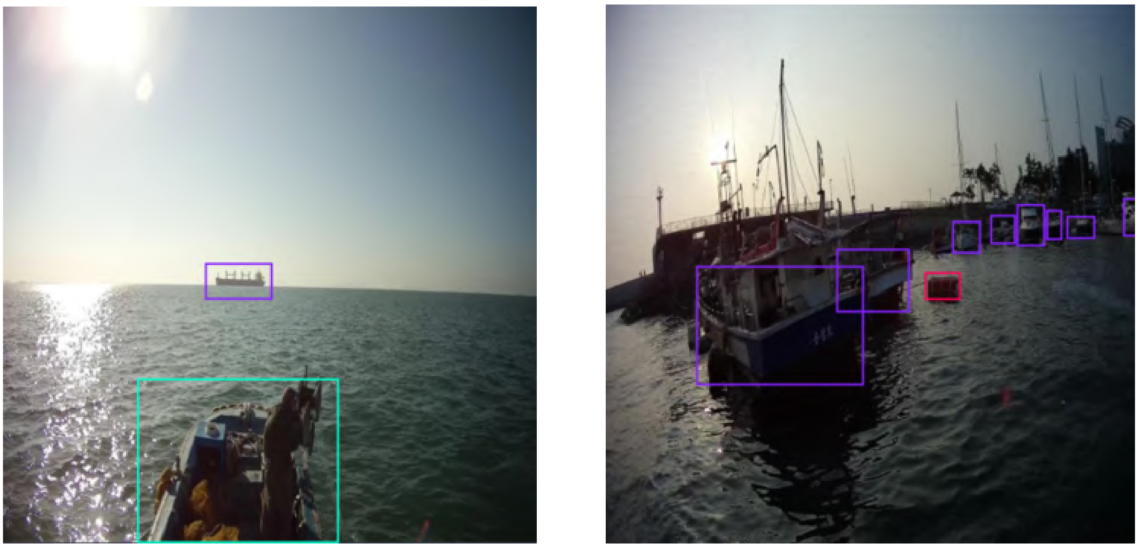
2.2. Primary testing

For the YOLOv5 model, the confusion matrix is shown in figure 10.

The history of the metrics of the training model is given in figure 11.

We can see that the model classifies swimmers and debris islands well, although this may be the result of insufficient images for these classes. The most successful class for recognition was “boat” with a probability of correct recognition of 52 percent, which is the expected result for this model. The classes “Buoy” and “Drowning man” are recognized worse and usually the model classifies them as background.

This may be due to both their similarity in the image and their small size and few special features. Ships are also poorly recognized, possibly due to the large number of images in the dataset, in which the ship is a tanker on the horizon and, accordingly, has small dimensions in the image.



(a) (b)

Figure 4: Examples of images of the “Ship” object class, (a) – single images of ships at a distance from each other and (b) – boats with a background and a certain number of the same and different types with and without accompanying objects.

(a) (b)

Figure 5: Examples of images of the object class “Buoy”, a), b) contain concentrated and distributed aggregates of buoys in a frontal view with background and extraneous objects. A distinctive feature of all markings is their small size.

Figure 12 shows the results of model verification.

The analysis shows that the model recognizes the object correctly, but gives a small percentage of confidence in its predictions. It is because of this fact that some objects remain unrecognizable.

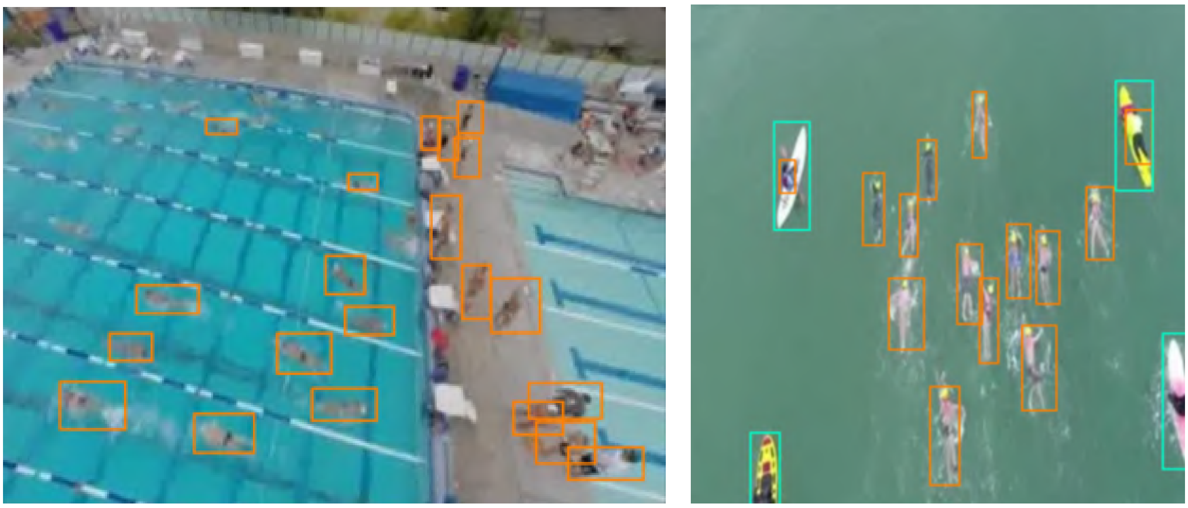
For the YOLOv8 model, figure 13 shows the confusion matrix.

The history of the metrics of the training model is given in figure 14.

Results of model verification is given in figure 15.

The analysis shows that the model recognizes the object correctly, but gives a small percentage of confidence in its predictions. It is because of this fact that some objects remain unrecognizable.

The obtained results give grounds for the conclusion that problems in training the model arise



(a) (b)

Figure 6: Examples of images of the “Swimmer” object class, all images are shallow and a large cluster is noted for them, Figure 6 b) contains extraneous objects.

(a) (b)

Figure 7: Examples of images of the object class “Drowning Man”, all images in Figure 7 are similar and almost indistinguishable from images of swimmers.

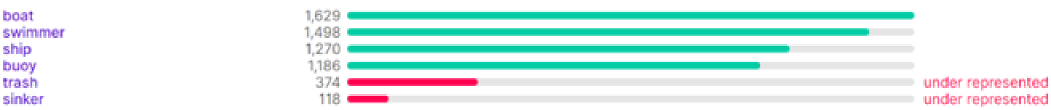


Figure 8: Distribution of images by classes.

precisely because of the dataset. Accordingly, for further development and obtaining a higher-quality model, an increase in the number and quality of marked images is required.

At the current stage, YOLO8 has better class recognition performance, although both models correctly locate objects, but have low confidence in the obtained results. This could be due to poor annotation, namely the similarity of the classes “Boat” and “Ship”, “Drowning man” and “Swimmer”, so I think it is

Figure 9: Distribution of images between datasets.

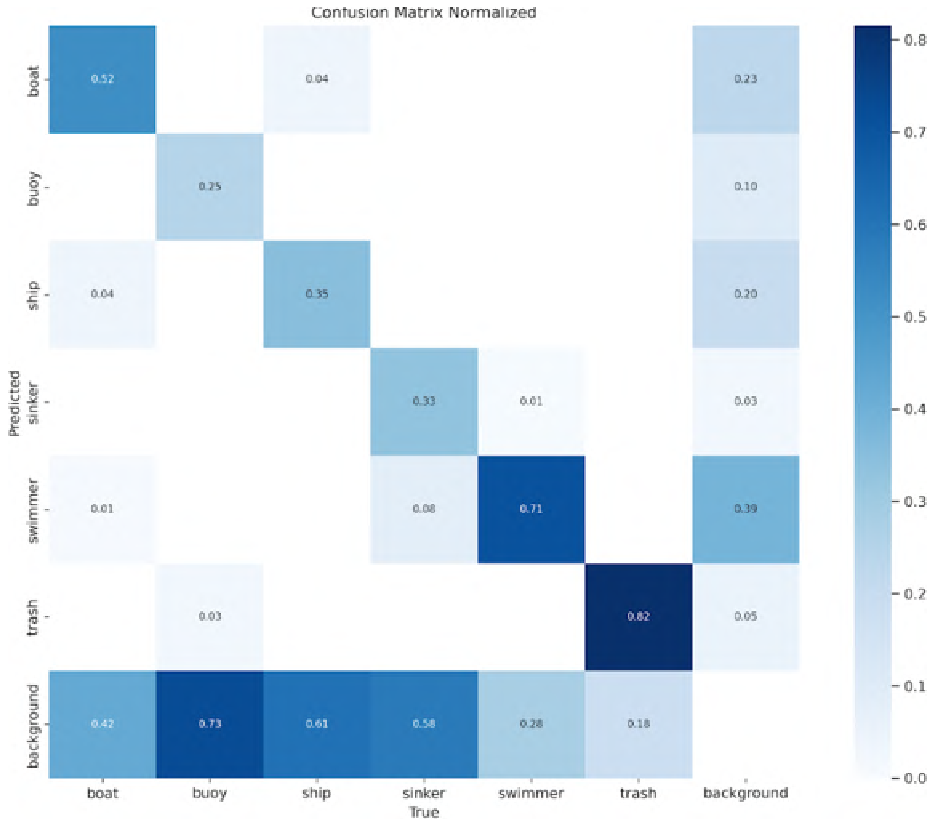


Figure 10: The confusion matrix.

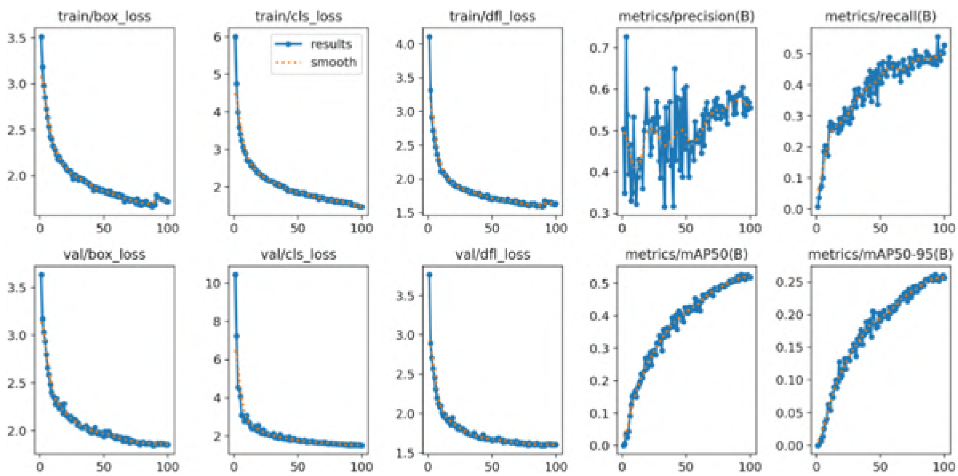
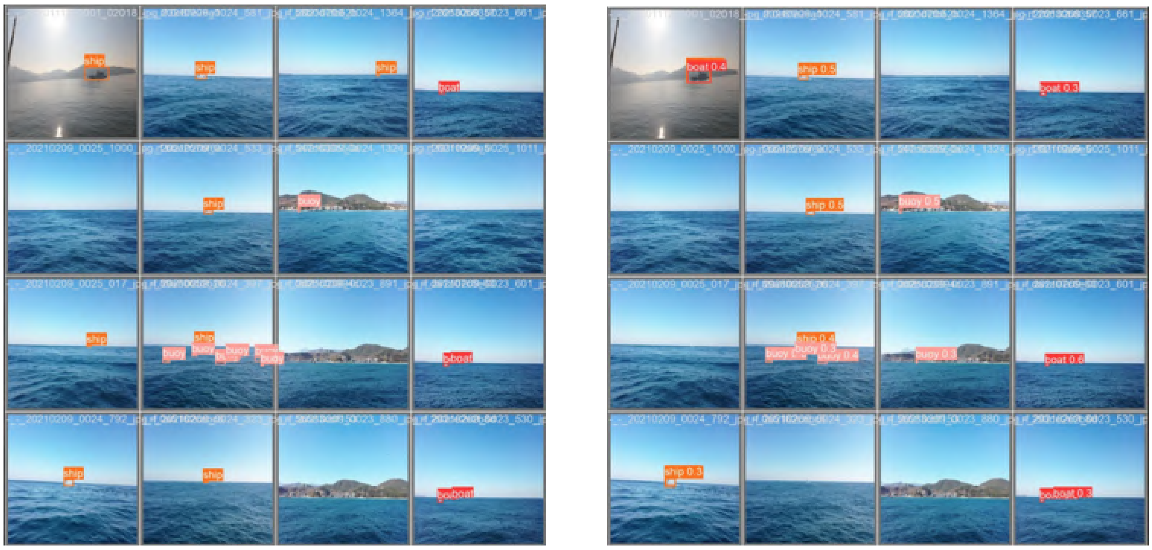


Figure 11: History of training model metrics.

necessary to add more images, re-evaluate the old ones and add data augmentation to diversify the training data.



(a) (b)

Figure 12: Results of model verification.

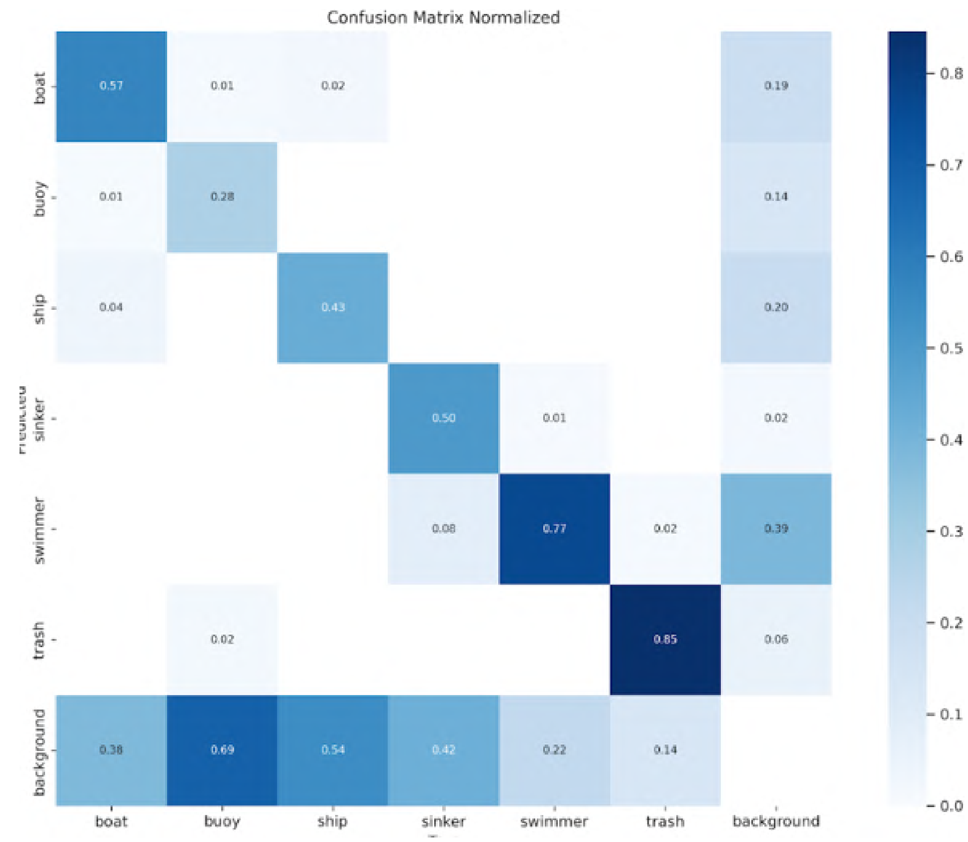


Figure 13: Confusion matrix for the YOLOv8 model.

3. Dataset editing

After marking another two thousand images, it was decided to move on to training a new model based on the YOLOv8 model. Images with the following types of augmentations were introduced into the dataset: cropping images for better behavior on images with small objects, generating a new image by

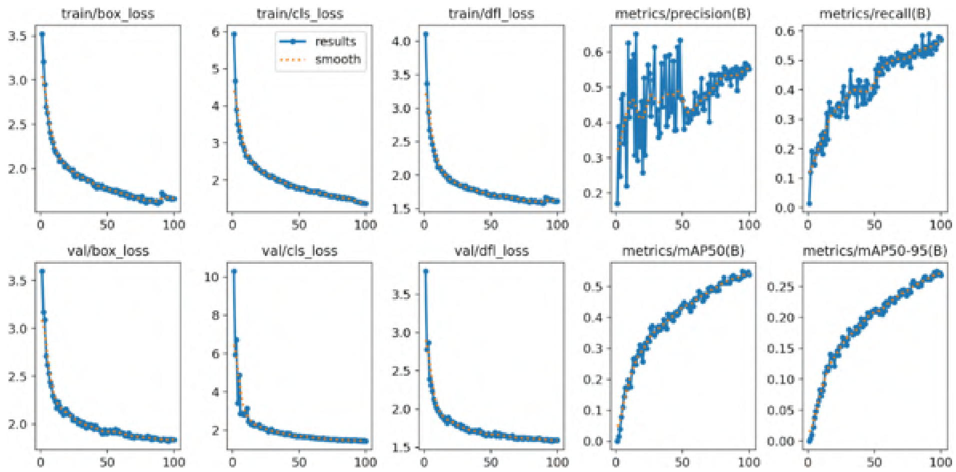
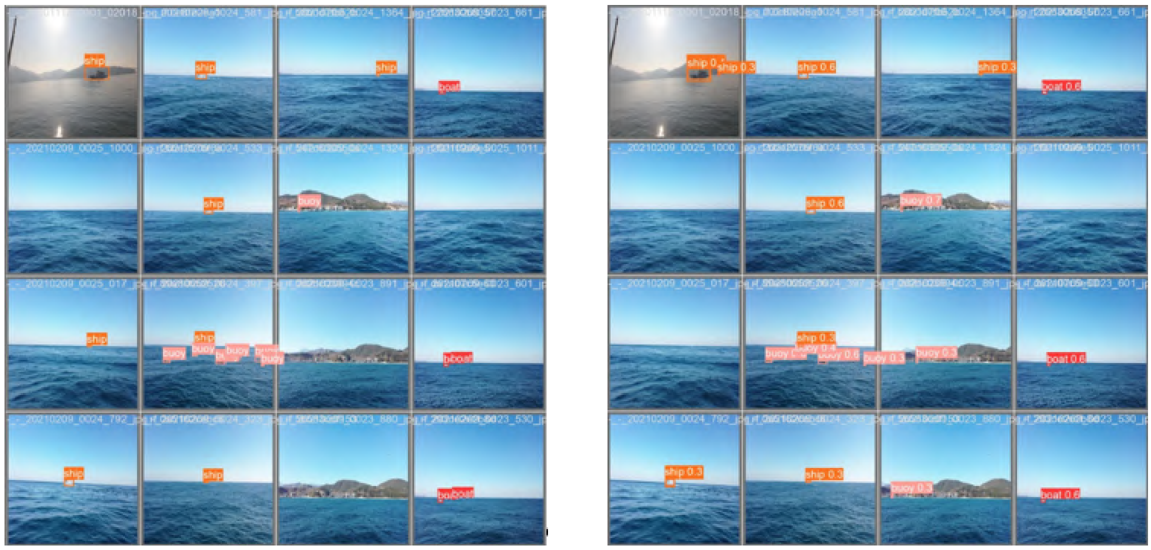


Figure 14: History of training model metrics.



(a) (b)

Figure 15: Results of model verification. a) – from the dataset, b) – network prediction.

placing images in a mosaic, which allows the model to better process images with lower resolution and smaller objects, vertical mirroring in order to exclude the possibility of an uneven distribution of boats between the classes with the nose to the left and to the right.

The parameters of the resulting dataset are given in figure 16.

Examples of augmented images of objects are shown in figure 17.

Training was performed over one hundred epochs with a pre-trained model provided by the ultralytics API. In figure 18 shows the history of training metrics of the YOLOv8 model.

Figure 19 shows the confusion matrix for the YOLOv8 model.

The results of detection of the newly trained model are shown in figure 20.

We can see that the results differ to a certain extent, which confirms the correctness of the chosen direction of improving the dataset.

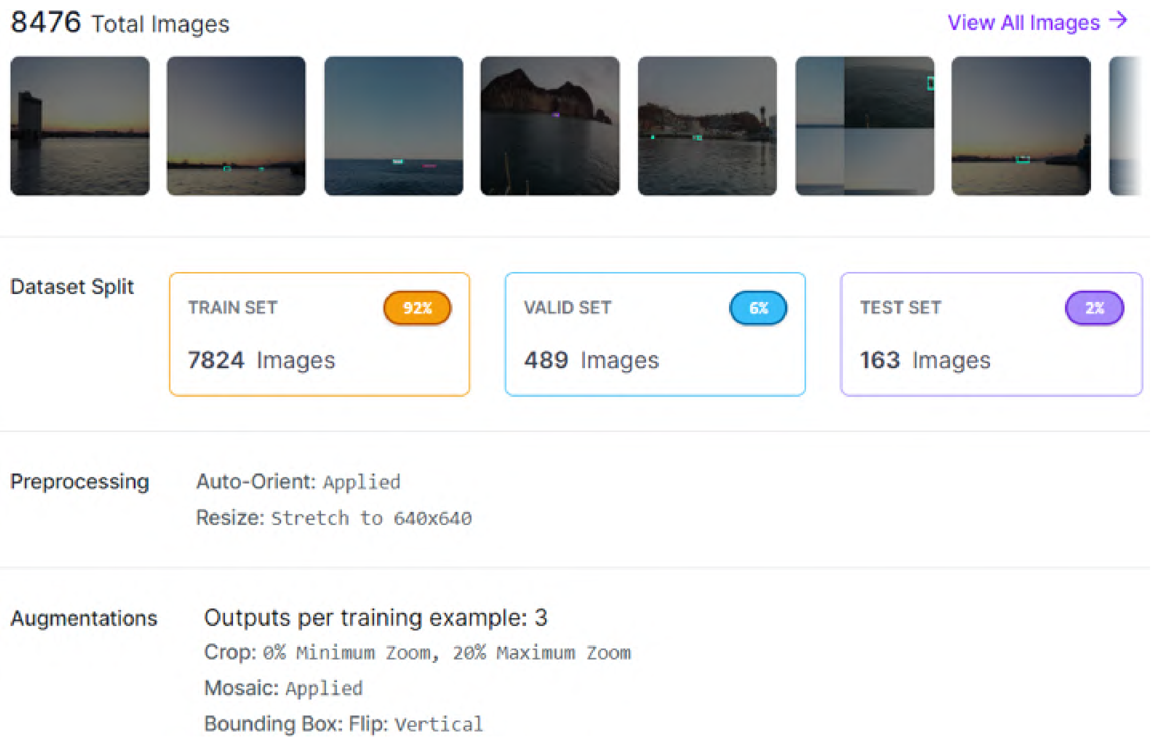
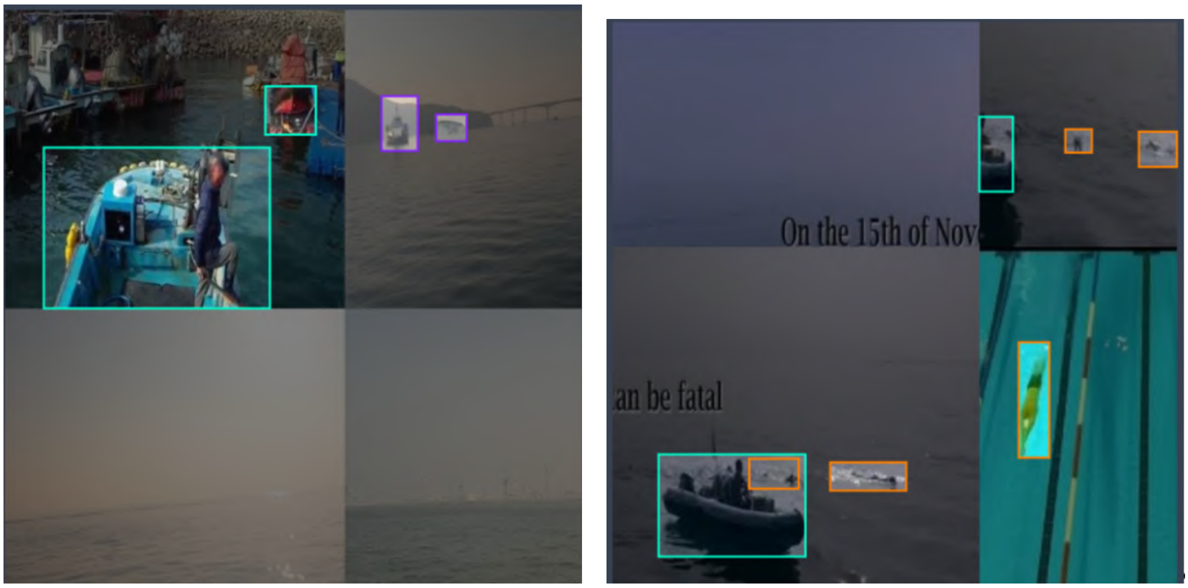


Figure 16: Parameters of the resulting dataset.



(a) (b)

Figure 17: A mosaic-augmented image of the object.

4. Testing

4.1. Testing on third-party images with classification enabled

In figure 21a shows the image after processing by the YOLOv8 model with an accuracy threshold of 0.4, which was able to detect the object “Swimmer” at the location of one of the many objects “Boat” with a probability of 0.56.

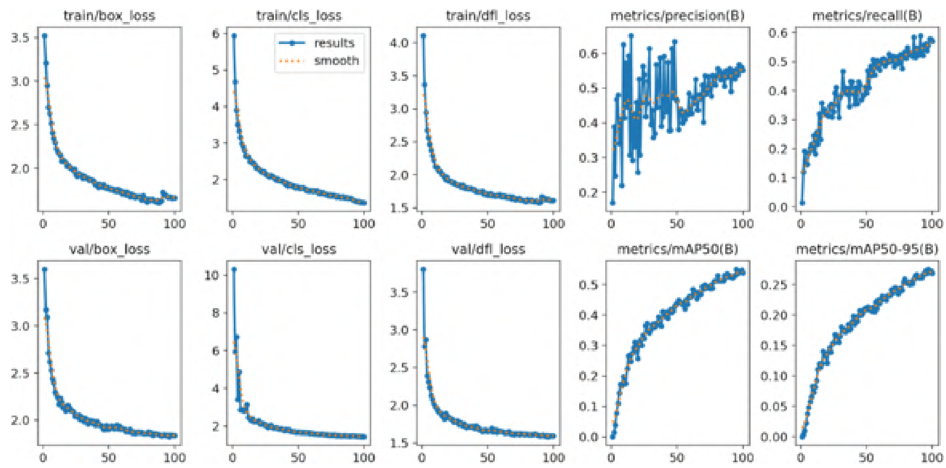


Figure 18: YOLOv8 model training metric history after dataset change.

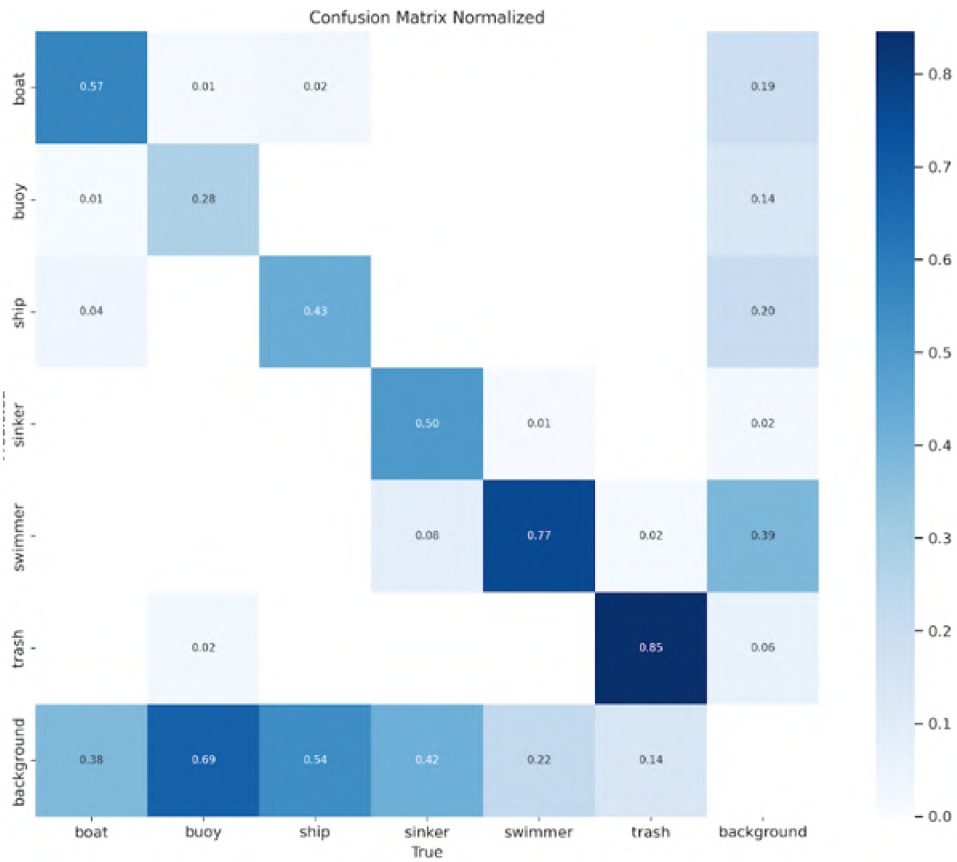


Figure 19: Confusion matrix for the YOLOv8 model.

The image shown in figure 21b has a high probability of recognizing the objects "Boat" and "Ship", but assigns the recognized objects to the wrong class (misclassification).

A fairly large volume of research was conducted, as a result of which a significant number of results were obtained and summarized, in particular. A single simple object in the frontal image is correctly recognized and classified. The small Buoy object in the background is completely ignored by the model. A small number of relatively large mountain-view objects were classified with high confidence and correctly. The swimmers closest to the camera were most likely detected, all other objects, including the "Boat", were not detected. When testing the model, the Garbage object was not recognized, and the Boat object was completely ignored. The example of a swimming frame illustrates the correct recognition

(a) (b)

Figure 20: Results of model verification: a) – marked, b) – predicted

(a) (b)

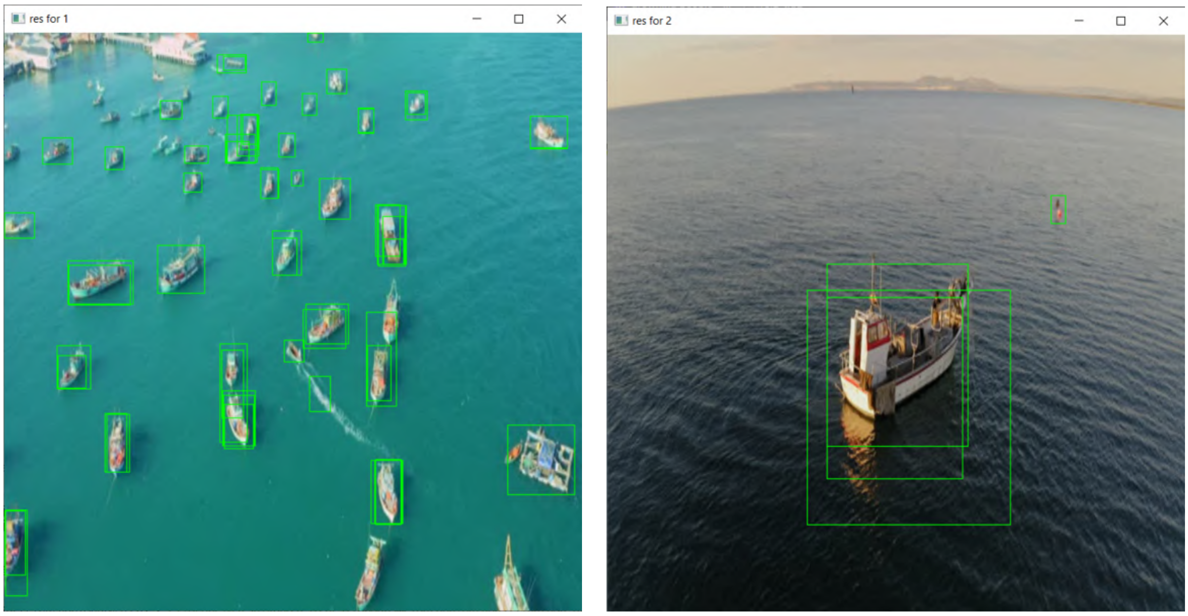
Figure 21: Image after processing with YOLOv8 model

of several Swimmer objects from a large number of available ones and the complete ignoring of the recognition of Boat objects. For the three Boat objects, the YOLOv8 image model sees two Swimmer objects instead of the expected Boat objects. Non-existent objects are not found, the class of existing objects is confused, possibly due to the small size of the latter.

4.2. Test results for reduced accuracy threshold with object detection classification function disabled, YOLOv8

For the following objects, the accuracy threshold was reduced to 0.1 and the classification function was disabled. Almost all recognized objects were correctly located. The model does not define objects where they do not exist.

The figure 22 shows examples of images with the accuracy threshold reduced to 0.1 and the classification function disabled.



(a) (b)

Figure 22: Image with reduced accuracy threshold to 0.1 and disabled classification function.

For experimental objects, 100 percent of objects were marked regardless of scale. Marking occurred several times, which can be corrected by filtering the resulting bounding boxes.

In the presence of a large cluster of diverse and different objects, large-scale foreground objects were marked. The rest of the objects are consolidated into one large object.

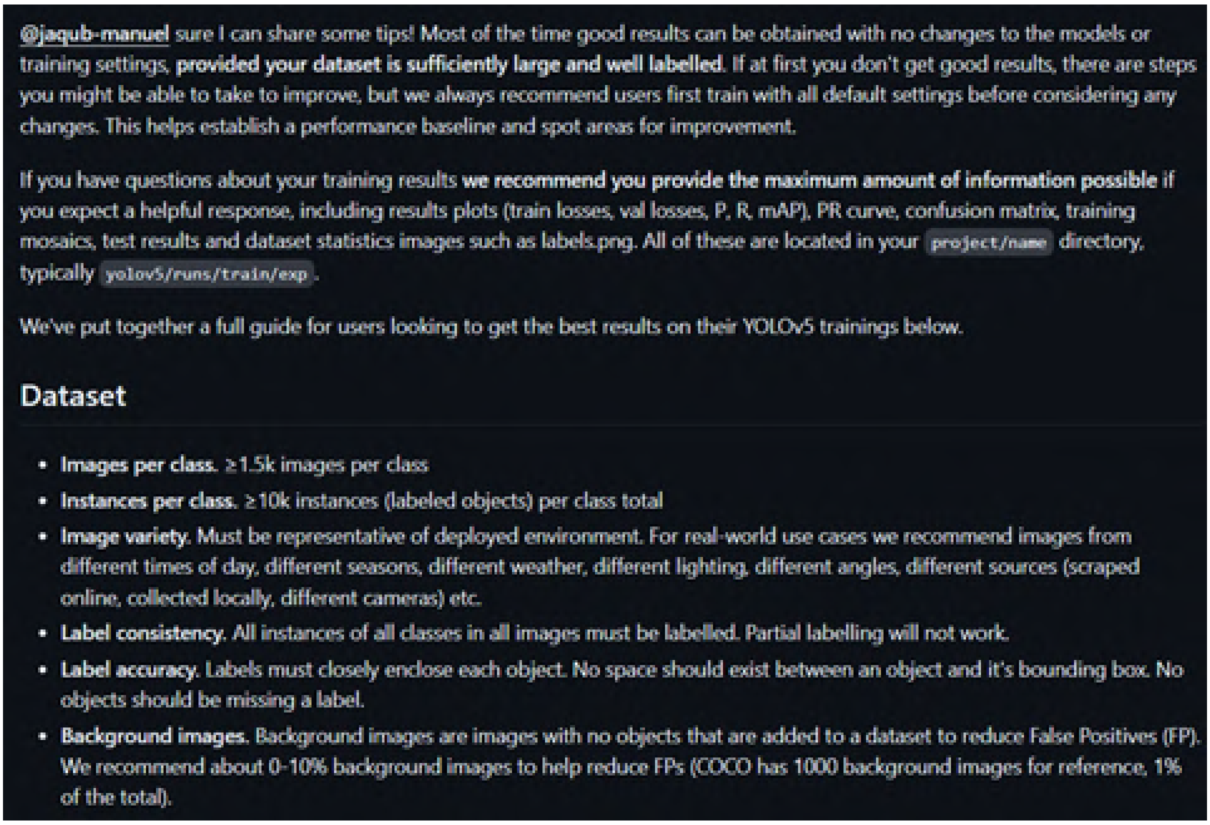
There were options where objects in the foreground only were recognized that were given a large scale, 100 percent of objects were marked regardless of scale, most of the various objects were marked regardless of scale, and non-existent objects were not marked. All obtained results are processed and summarized.

5. Conclusions

According to the obtained results, it can be seen that the system detects objects on the water surface, but the classification of these objects is not good. There are several reasons for this: errors in the labeling of the dataset and the small size of the dataset.

The comments shown in the figure 23 have been received from the API developers for working with artificial intelligence.

A possible scenario of using the built model is the general collection of information about the reservoir without regard to the classification output. In the process of such exploitation, it can be considered as expedient to collect a dataset that will correspond to the data from the drone (the data of the current dataset is data from surveillance cameras and video recordings from boats). In the future, form the dataset according to the developer's requirements, applying the necessary data augmentation steps.



@jaqub-manuel sure I can share some tips! Most of the time good results can be obtained with no changes to the models or training settings, **provided your dataset is sufficiently large and well labelled.** If at first you don't get good results, there are steps you might be able to take to improve, but we always recommend users first train with all default settings before considering any changes. This helps establish a performance baseline and spot areas for improvement.

If you have questions about your training results **we recommend you provide the maximum amount of information possible** if you expect a helpful response, including results plots (train losses, val losses, P, R, mAP), PR curve, confusion matrix, training mosaics, test results and dataset statistics images such as labels.png. All of these are located in your `project/name` directory, typically `yolov5/runs/train/exp`.

We've put together a full guide for users looking to get the best results on their YOLOv5 trainings below.

Dataset

- **Images per class.** ≥1.5k images per class
- **Instances per class.** ≥10k instances (labeled objects) per class total
- **Image variety.** Must be representative of deployed environment. For real-world use cases we recommend images from different times of day, different seasons, different weather, different lighting, different angles, different sources (scraped online, collected locally, different cameras) etc.
- **Label consistency.** All instances of all classes in all images must be labelled. Partial labelling will not work.
- **Label accuracy.** Labels must closely enclose each object. No space should exist between an object and it's bounding box. No objects should be missing a label.
- **Background images.** Background images are images with no objects that are added to a dataset to reduce False Positives (FP). We recommend about 0-10% background images to help reduce FPs (COCO has 1000 background images for reference, 1% of the total).

Figure 23: Developer comments.

6. Author contributions

The idea of writing the article belongs to all authors. Viktorija Smolij built and trained the model, Natan Smolij performed testing and analyzed the training results, Sergii Sayapin collected and marked up the dataset, performed the design of the materials.

Acknowledgments

When writing the article, special thanks should be expressed to the head of the department Oleksandr Rolik, Volodymyr Oliynyk, Yury Berdnyk and Mykola Shynkevych. Great gratitude for the explanation and organization of work, support and kindness.

References

[1] P. Schönfelder, F. Stebel, N. Andreou, M. König, Deep learning-based text detection and recognition on architectural floor plans, Automation in Construction 157 (2024) 105156. doi:10.1016/j.autcon.2023.105156.

[2] N. Giakoumoglou, E. M. Pechlivani, D. Tzovaras, Generate-Paste-Blend-Detect: Synthetic dataset for object detection in the agriculture domain, Smart Agricultural Technology 5 (2023) 100258. doi:10.1016/j.atech.2023.100258.

[3] M. M. Mintii, Exploring the landscape of STEM education and personnel training: a comprehensive systematic review, Educational Dimension 9 (2023) 149–172. doi:10.31812/ed.583.

[4] K. S. Tarisayi, Strategic leadership for responsible artificial intelligence adoption in higher education, CTE Workshop Proceedings 11 (2024) 4–14. doi:10.55056/cte.616.

[5] M. Ashourpour, G. Azizpour, K. Johansen, Real-Time Defect and Object Detection in Assembly Line: A Case for In-Line Quality Inspection, in: F. J. G. Silva, A. B. Pereira, R. D. S. G. Campilho (Eds.), Flexible Automation and Intelligent Manufacturing: Establishing Bridges for More Sustainable Manufacturing Systems, Springer Nature Switzerland, Cham, 2024, pp. 99–106. doi:10.1007/978-3-031-38241-3_12.
[6] P. Azevedo, V. Santos, Comparative analysis of multiple YOLO-based target detectors and trackers for ADAS in edge devices, Robotics and Autonomous Systems 171 (2024) 104558. doi:10.1016/j.robot.2023.104558.
[7] M. Sanjai Siddharthan, S. Aravind, S. Sountharrajan, Real-Time Road Hazard Classification Using Object Detection with Deep Learning, in: P. P. Joby, M. S. Alencar, P. Falkowski-Gilski (Eds.), IoT Based Control Networks and Intelligent Systems, Springer Nature, Singapore, 2024, pp. 479–492. doi:10.1007/978-981-99-6586-1_33.
[8] N. V. Morze, O. V. Strutynska, Advancing educational robotics: competence development for pre-service computer science teachers, CTE Workshop Proceedings 10 (2023) 107–123. doi:10.55056/cte.549.
[9] Z. H. Wei, Y. J. Zhang, X. J. Wang, J. T. Zhou, F. Q. Dou, Y. H. Xia, A YOLOv8-based approach for steel plate surface defect detection, Metalurgija 63 (2024) 28–30.
[10] F. Wu, Y. Zhang, L. Wang, Q. Hu, S. Fan, W. Cai, A Deep Learning-Based Lightweight Model for the Detection of Marine Fishes, Journal of Marine Science and Engineering 11 (2023) 2156. doi:10.3390/jmse11112156.
[11] G. Zhang, Y. Tang, H. Tang, W. Li, L. Wang, A global lightweight deep learning model for express package detection, Journal of Intelligent and Fuzzy Systems 45 (2023) 12013–12025. doi:10.3233/JIFS-232874.
[12] J. Wang, H. Dai, T. Chen, H. Liu, X. Zhang, Q. Zhong, R. Lu, Toward surface defect detection in electronics manufacturing by an accurate and lightweight YOLO-style object detector, Scientific Reports 13 (2023) 7062. doi:10.1038/s41598-023-33804-w.
[13] A. Li, Z. Zhang, S. Sun, M. Feng, C. Wu, MultiNet-GS: Structured Road Perception Model Based on Multi-Task Convolutional Neural Network, Electronics 12 (2023) 3994. doi:10.3390/electronics12193994.
[14] L. Han, C. Ma, Y. Liu, J. Jia, J. Sun, SC-YOLOv8: A Security Check Model for the Inspection of Prohibited Items in X-ray Images, Electronics 12 (2023) 4208. doi:10.3390/electronics12204208.
[15] A. R. Petrosian, R. V. Petrosyan, I. A. Pilkevych, M. S. Graf, Efficient model of PID controller of unmanned aerial vehicle, Journal of Edge Computing 2 (2023) 104–124. doi:10.55056/jec.593.
[16] J. Mao, L. Wang, N. Wang, Y. Hu, W. Sheng, A novel method of human identification based on dental impression image, Pattern Recognition 144 (2023) 109864. doi:10.1016/j.patcog.2023.109864.
[17] E. Kara, G. Zhang, J. J. Williams, G. Ferrandez-Quinto, L. J. Rhoden, M. Kim, J. N. Kutz, A. Rahman, Deep learning based object tracking in walking droplet and granular intruder experiments, Journal of Real-Time Image Processing 20 (2023) 86. doi:10.1007/s11554-023-01341-4.
[18] S. Zhou, M. Zhong, X. Chai, N. Zhang, Y. Zhang, Q. Sun, T. Sun, Framework of rod-like crops sorting based on multi-object oriented detection and analysis, Computers and Electronics in Agriculture 216 (2024) 108516. doi:10.1016/j.compag.2023.108516.
[19] P. Shan, R. Yang, H. Xiao, L. Zhang, Y. Liu, Q. Fu, Y. Zhao, UAVPNet: A balanced and enhanced UAV object detection and pose recognition network, Measurement: Journal of the International Measurement Confederation 222 (2023) 113654. doi:10.1016/j.measurement.2023.113654.
[20] S. O. Semerikov, T. A. Vakaliuk, I. S. Mintii, V. A. Hamaniuk, V. N. Soloviev, O. V. Bondarenko, P. P. Nechypurenko, S. V. Shokaliuk, N. V. Moiseienko, V. R. Ruban, Development of the computer vision system based on machine learning for educational purposes, Educational Dimension 5 (2021) 8–60. doi:10.31812/educdim.4717.
[21] F. M. Talaat, H. ZainEldin, An improved fire detection approach based on YOLO-v8 for smart cities, Neural Computing and Applications 35 (2023) 20939–20954. doi:10.1007/s00521-023-08809-1.

[22] S. Liu, Q. Fan, C. Zhao, S. Li, RTAD: A Real-Time Animal Object Detection Model Based on a Large Selective Kernel and Channel Pruning, Information 14 (2023) 535. doi:10.3390/info14100535.

[23] V. Smolij, About features of management preproduction of electronic vehicles, Problems of Modeling and Design Automatization 11 (2019). doi:10.31474/2074-7888-2019-1-33-42.

[24] Y. Su, W. Tan, Y. Dong, W. Xu, P. Huang, J. Zhang, D. Zhang, Enhancing concealed object detection in Active Millimeter Wave Images using wavelet transform, Signal Processing 216 (2024) 109303. doi:10.1016/j.sigpro.2023.109303.

[25] C. Liu, K. Wang, Q. Li, F. Zhao, K. Zhao, H. Ma, Powerful-IoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism, Neural Networks 170 (2024) 276–284. doi:10.1016/j.neunet.2023.11.041.

[26] W. Xu, C. Liu, G. Wang, Y. Zhao, J. Yu, A. Muhammad, D. Li, Behavioral response of fish under ammonia nitrogen stress based on machine vision, Engineering Applications of Artificial Intelligence 128 (2024) 107442. doi:10.1016/j.engappai.2023.107442.

[27] G. Dimauro, N. Barbaro, M. G. Camporeale, V. Fiore, M. Gelardi, M. Scalera, DeepCilia: Automated, deep-learning based engine for precise ciliary beat frequency estimation, Biomedical Signal Processing and Control 90 (2024) 105808. doi:10.1016/j.bspc.2023.105808.

[28] X. Zhao, Y. Song, Improved Ship Detection with YOLOv8 Enhanced with MobileViT and GSConv, Electronics 12 (2023) 4666. doi:10.3390/electronics12224666.