UDC 681.325

POST PROCESSING OF PREDICTIONS TO IMPROVE THE QUALITY OF RECOGNITION OF WATER SURFACE OBJECTS

Smolij V. M. – Dr. Sc., Professor, Professor of the Department of Information systems and technologies, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine.

Smolij N. V. –Student of the Department of Information systems and technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Mokriiev M. V. – PhD, Associate professor of the Department of Information systems and technologies, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine.

ABSTRACT

Context. The significance of this work stems from the growing need for UAV technologies integrated with artificial intelligence, aimed at detecting and identifying objects on the surface of water bodies. Modern needs in water body monitoring, especially in the context of environmental monitoring, protection and resource management, require accurate and reliable solutions. This work demonstrates methods for improving the performance of neural networks and offers approaches for processing NN predictions, even if they are trained on irrelevant data, which increases the versatility and efficiency of the technology.

Objective. The goal of the work is to solve the problem of false recognition of objects on the surface of water bodies, which is due to a decrease in the accuracy threshold for the neural network. This provides more accurate and reliable detection, reducing the number of false positive predictions and increasing the efficiency of the system in general.

Method. It is proposed to add a stage of post-processing of NN predictions, which inherits concepts of min-max suppression used by YOLO models. This algorithm suppresses the re-detection of the object by the network and relies on the cross-sectional area of the detected rectangles. It uses a threshold value of 0.8 for the two points of the rectangle, which can effectively reduce the number of re-predictions and improve the accuracy.

Results. As a result of the implementation of the proposed algorithm and the script created on its basis, a result was achieved in which groups from several predictions are combined and filtered. The received data is stored in the database as found and detected objects. The proposed post-processing algorithm effectively removes redundant predictions while maintaining forecast accuracy. This ensures the reliability of the system and increases its performance in real conditions.

Conclusions. Detected images of objects on the surface of water bodies are stored in the database in the form of records with unique file name identifiers. After tests with pre-taken images algorithm proved it`s persistence against data duplication scenarios. This increases the efficiency and reliability of the monitoring system, ensuring accurate and timely detection of objects on the surface of water bodies.

KEYWORDS: UAV, detection, recognized objects, water surface, neural network, dataset, model, image distribution, error matrix, training metrics, magnification, image mosaicking, image post-processing, implementation script, mission log, database.

ABBREVIATIONS

AI stands for artificial Intelligence;
FC is a flight controller;
UAV is a unmanned aerial vehicle;
YOLO is the You Only Look Once;
NN is a neural network;
RS is a remote sensing;
SPI is a serial peripheral interface;
IS is a information system;
ConvN is a deep convolution neural network;
DB is a database;
AP is a indicator of average accuracy – average accuracy;
IoU is a intersection over union;
NCU is a nonlinear model predictive control.

NOMENCLATURE

xi, yi is the coordinates of the corresponding corner of the rectangle;
TP is a true positive;
TN is a true negative;
FP is a false positive;
FN is a false negative;
GT is a ground truth;
P is a prediction;
r is a recall value for one sample;
$\tilde{r}$ is a average recall value.

INTRODUCTION

The importance of the project arises from the current need for artificial intelligence integration into UAV technologies. This article shows bright example in context of localization and identification problems for objects on the surface of water bodies.

The object of study may be defined as algorithm description for task of quantity reduction of false recognition cases and further implementation into information system.

The process of forming and downloading the algorithm and the created script based on it, a result was obtained where groups from several predictions are combined and filtered. As a result, these findings are saved in the database in the form they were obtained.

The subject of study is the implementation of the proposed algorithm and the created script based on it, a result was obtained where groups from several predictions are combined and filtered. As a result, detected objects are saved in the database in the form they were obtained.

DOI 10.15588/1607-3274-2024-4-13

The purpose of the work is to solve the problem of false recognition of objects caused by manipulations with train and process parameters for neural network.

1 PROBLEM STATEMENT

The current state of development of artificial intelligence technologies significantly expands the range of tasks that can be solved by robotics tools [1], [2]. Most needed areas of automatization for area of UAVs involve such tasks as mission completion time reduction, more efficient usage of computational powers and reduction of time of human intervention into UAV control processes [3], [4], [5]. Patrolling water bodies, searching, identifying and classifying objects on their surface using traditional methods is a complex process with high requirements for time and computational powers that needs to be enhanced with implementation of various modern techniques such as artificial intelligence and mode [6], [7], [8].

In the process of performing the mission, the UAV finds a certain object, determines its coordinates, with the help of a neural network, prediction rectangles are generated, characterized by *AP*, *Pinterp*, which must be filtered. The same object can be recognized by the neural network as several different objects, which causes redundancy and duplication of data in the DB.

It is necessary to develop an algorithm (procedure, technique, method) of filtering (elimination of redundancy) generated rectangles in order to avoid duplication of predictions. It is also necessary to test the proposed algorithm, check the efficiency of the work, determine the cut-off threshold value at which the object will be detected, but there will be no duplication. This coefficient is defined as the ratio of the cross-sectional area to the areas of two rectangles. It is necessary to expand the box with which the intersection is fixed to the borders of the intersecting box. This achieves preservation of detection quality while eliminating duplication.

It was found that the coordinates of the upper right and lower left corners of the rectangle are considered for each object. A set of rectangles at two points ($x1$, $y1$, $x2$, $y2$) associated with objects that can intersect with each other and thus create duplicate objects. The specified rectangles are the result of neural network predictions. The number of specified rectangles is inversely proportional to the prediction accuracy threshold.

The job task is to eliminate duplication while maintaining the accuracy of recognition. To do this, it is necessary to filter this set of rectangles using the proposed threshold filtering algorithm with an overlap threshold of 0.8, getting rid of duplicate markings and not reducing the number of detected objects. It is imperative to investigate the operability and efficiency of the proposed post-processing algorithm.

2 REVIEW OF THE LITERATURE

For modern unmanned aerial vehicles, great variety of tasks such as: classification, observation, research, and classification are getting redirected to certain automated subsystems, which may enhance performance thanks to the additional computational modules involved [9], [10], [11]. Such type of an approach is appropriate and widespread due to the availability of flight controllers on the market [12], [13]. Available options suggest great variety of sockets for external devices for variety of purposes: exchange of GPS data, OSD frames and other types of messages defined via MavLink. Such data can be clustered into data sets for the variety of purposes: prioritized task queuing and data processing routines [14], [15]. Computer vision systems implement various machine learning algorithms and complex data processing techniques for image processing and localization tasks The complexity of the structure of a computer vision system may vary depending on the type of task and the stack of technologies used [16], [17].

As well as any other information system, computer vision system involves such functionality as data collection processing and storage. Additional functions may vary from performing computations for neural network up to data generalization, clustering and image segmentation with further additions in the form of decision-making process. The design of UAV for computer vision tasks includes the following structural elements: reconnaissance equipment, UAV frame, propeller group and flight controller. It should be noted that the range of use of UAVs in water and coastal zones is very large and responsible, since the functions of object detection, observation, identification, classification, etc. directly related to the security, protection and territorial integrity of Ukraine. The purpose of the research is to determine the accuracy of object recognition, to find out the reasons for possible unsatisfactory recognition quality, to form means and strategies for improving the quality of image recognition of objects. It should be noted that effective algorithms for the detection and classification of objects on the water surface have not yet been created, apart from artificial intelligence and neural network tools based on it.

The technical system, which includes UAVs and AI tools, will allow relatively fast and high-quality processing of data frames representing any type of surface below UAV, thereby providing efficient observation functions performance, response and elimination of emergency situations with minimal human involvement [18]. The main task of the work is to detect and recognize the object in the image. Mentioned area suggests great variety of artificial intelligence scientific works and solutions to study that differ in the structure of the neural networks used, the dataset used, and are accordingly characterized by the quality of object recognition. These works mostly rely on such ML libraries as pytorch, theano, tensorflow, chainer, ultralytics. The main goal of the work is to create a data set regarding the recognized object, which in turn requires the use of the Ultralytics API library, which provides a wide functionality for dataset gathering and marking and an interface for variety of new and aout of date convolution neural networks from YOLO family. Due to the support of dynamic typing, automatic garbage collection and the availability of a library API, the software in

DOI 10.15588/1607-3274-2024-4-13

the work uses Python programming language. Source [1] explains in details (www.overleaf.com).

3 MATERIALS AND METHODS

The task of patrolling water surfaces and detecting as well as identifying objects relies on interaction with a neural network, the detailed workings, architecture, and training of which are discussed in [20]. For this particular neural network, it is necessary to develop tools for generating an on-board flight log and producing reports on the detected objects. The Ultralytics API was selected for model interaction. Additionally, it is important to mention that the OpenCV library was utilized for image processing due to its ability to easily transition from handling test data from media files to real-time camera data.

Since the task conditions require a difficult environment of use, namely working near water bodies and short missions with the transfer of the payload, it was decided to use a quadcopter. The basis for the choice was: a smaller number of parts to be serviced, compared to other members of the family, and optimal indicators of stability, speed and maximum indicators of the weight that can be carried. The use of the scheme of distributed current controllers was chosen due to high wear resistance and economy. Due to the predicted high rates of wear of the parts, this scheme will not replace the board with all four current controllers, but replace only the controller that has failed.

To build a UAV, you need to have a flight controller or the ability to write all the necessary functionality. Since writing your own flight controller is a time-consuming task, which, in terms of time and effort, far exceeds the defined time limits of the diploma project, we consider FCs available on the market. Existing flight controllers use a variety of software.

Software leaders include software from ArduPilot, PX4 – open source projects with a large audience of users, and DJI – licensed software used on more expensive UAV models and designed for flexible use with different UAV models and configuration variations. It should also be noted that ArduPilot is the most ‘general’ project, since the range of tasks to be solved for this project ranges from the design of UAV rotor group control systems to the programming of autopilot systems. The PX4 is only a draft autopilot system.

The family of platforms that use the ArduPilot code as firmware is quite large, and usually the decision to use a certain controller model is determined only by the user’s budget. Thus, the decision was made to use the Pixhawk 2.4.8 flight controller. Among its advantages are: the possibility of connecting 2x GPS to combine data and increase positioning accuracy, a built-in telemetry filtering system, a large number of peripheral interfaces supporting I2C, UART, SPI protocols, and a large number of drivers for peripheral devices that are not can be ignored when considering the perspective of development/complication of the created information system during its life cycle.

Also, a separate advantage is the presence of the MavLink protocol, which allows you to organize the exchange of messages with peripheral devices, and thus connect the on-board computer, which will be able to use such data of the flight controller as: location assessment, positioning quality, battery charge level, weather conditions for making own decisions about the continuation of the mission.

Since the flight controllers from Pixhawk do not provide functionality for deploying a neural network and a database, and the presence of a network card is an exception, it was accordingly decided to install an on-board microcomputer on the UAV for use as a control, processing, storage and information transfer device. This approach will allow you to get a full-fledged operating system, access to the package manager and, accordingly, to a convenient setting of the environment, will facilitate the process of writing and testing software.

The GPIO module will allow you to organize a connection with your own module for water intake. The built-in software application RealVNC and the ability to set a static IP address will provide an opportunity to connect to the UAV through the mobile network in the absence of such computer components as a monitor and keyboard. Built-in USB ports provide a convenient connection with a flight controller for data exchange using the MavLink protocol.

The use of a neural network on RaspberryPI is a typical step in the tasks of a limited number of resources for building a robotic information system for analyzing the state of water bodies and recognizing objects with the help of AI technologies. This gives an absolute advantage to this platform due to the large volume of available materials on solving problems and setting up processes.

Since the RaspberryPi has an OS with a package manager capable of installing all the necessary functionality with the help of sudo apt install commands, and because of the low cost of the hardware and the prevalence of a free license for the software, it was decided to use it.

Anything related to the use of AI can be classified into one of the following categories: application use, application development, model development, or infrastructure design (Fig. 1).

Research is carried out in the area of application development, since it is necessary to introduce artificial intelligence methods into the user environment, and to train models, since the requirements for the use of any specific model have not been set.

Since the experience of working with the ultralytics API was gained during the training, it was decided to use it. This module provides opportunities for training a neural network and for its integration into the application. Also, during the training process, statistics are collected and the metrics of a certain trained model are automatically built. This simplifies the process of iterating between training stages, evaluating and changing training conditions.

DOI 10.15588/1607-3274-2024-4-13

Figure 1 – AI technology stack


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 – Comparative characteristics of the accuracy of dimensional variations of two models depending on the type of task

The objects of observation are boats, ships, buoys, people and piles of garbage on water bodies. Accordingly, the task of the model is detection, since segmentation is not suitable for this type of work, although it can be used to improve accuracy or increase the functionality of the application being created.

According to the conditions of the task, in which there are significant limitations in computing power and the ability of UAVs to carry loads, it is necessary to use the second class of detection models, namely, single-stage detectors, which, accordingly, filter the image once. This approach reduces the accuracy of the forecast, reducing the amount of consumed computing resources. Also, in conditions when the dataset is collected from open sources, training anomalies are inevitable, which will lead to a decrease in accuracy even with a two-stage detector. Accordingly, the dual use of UAVs is expedient: both to perform the specified task and as a source of data for the formation of a better dataset.

Since it was necessary to achieve such a classification result that would not be completely wrong, with limited resources in the computer vision problem, it was decided to use the YOLOv8 model, as well as the API from Ultralytics, which was created in order to give the opportunity to train typical models on cash registers-volume datasets for various tasks.

A compilation of video footage from a fishing boat, video surveillance cameras of the pool, and photos from ecologists' boats was used as a dataset.

As it was mentioned in [24], 2 contenders were singled out among the family of YOLO models: YOLOv8, YOLOv5. The developers independently compared the accuracy of the classification, detection and segmentation tasks of the data models (Fig. 2).

Accordingly, it was decided to make our own tests for selected models in the Small dimension. Since we are facing an image detection task, the expected advantage towards YOLOv8 is 20% on 640x640 images.

For training, an example from the developers was used with arguments to improve the quality of training for the YOLOv8 and YOLOv5 models, respectively. Training was carried out for 100 epochs. The training results were evaluated based on such quality indicators as:

– the confusion matrix is a metric that characterizes how the neural network distinguishes between objects of different classes;

– P curve – a curve that characterizes the process of establishing the accuracy value as the number of correctly defined objects of the class in relation to the total number of defined objects of this class;

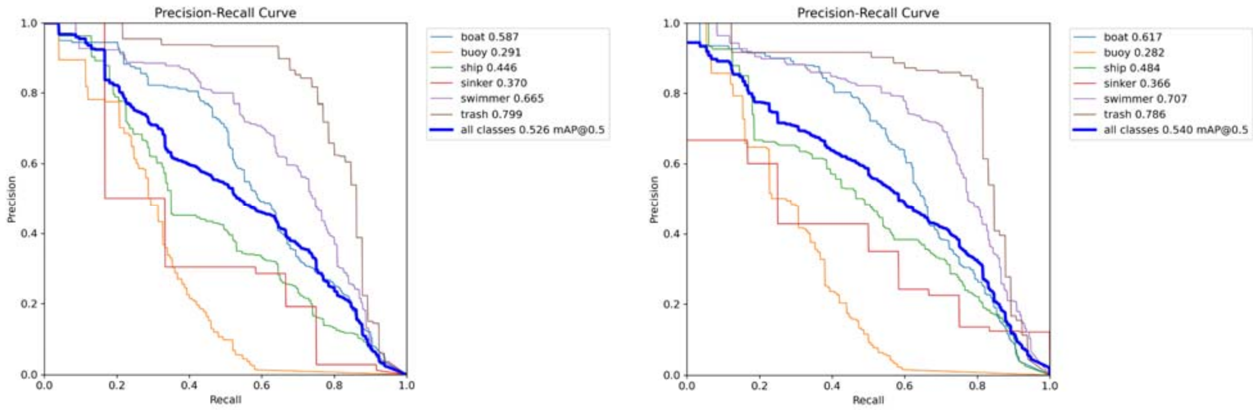
– R curve – a curve that characterizes the process of setting the recall-y value, which characterizes the ratio of the number of correctly defined objects of a certain class to their total number in the data set;

– mAP (:50, 50–95) curve – which characterizes the formation of the average accuracy value for the confidence threshold of 50% and in the interval from 50 to 95%, respectively. This metric provides an opportunity to evaluate how the model copes with different thresholds for accuracy/

Based on the results of training on the created dataset, the following diagrams describing the performance of 2 models were obtained. The results of the study of PR-metrics of the models, which characterize the change in the precision and recall of the parameters of the YOLOv5 and YOLOv8 models are presented in Fig. 3.

For boats, the degree of correlation between precision and recall indicators is 0.587 for the NN YOLOv5 model and 0.617 for the NN YOLOv8 model. The recognition of buoys occurs with a degree of correlation between the precision and recall indicators of 0.291 and 0.282 for YOLOv5 and YOLOv8 NN models, respectively. For ships, this indicator is 0.446 and 0.484, when recognizing drowning people – 0.370 and 0.366, when identifying swimmers, the correlation coefficient of precision and recall indicators is 0.665 and 0.707 (there is an advantage of the NN YOLOv8 model), and when recognizing on the other hand, with coefficients of 0.799 and 0.786, the YOLOv5 NN model is preferred in the bath of garbage images.

DOI 10.15588/1607-3274-2024-4-13



a b

Figure 3 – PR-metrics of models NN:
a – YOLOv5; b – YOLOv8

AP is the precision averaged over all completeness values between 0 and 1[25]. AP is interpreted as finding the area of the region below the precision-completeness curve. The mAP50-95 indicator is the average value of the average accuracy, calculated at different IoU threshold values, varying from 0.50 to 0.95. mAP@0.5 is the average accuracy calculated at an IoU threshold of 0.50. This is a measure of the accuracy of the model, taking into account only ‘light’ detections.

$$Precision = \frac{TP}{TP + FP}. \quad (1)$$

$$Recall = \frac{TP}{TP + FN}. \quad (2)$$

$$IoU = \frac{area \quad of \quad overlap}{area \quad of \quad union}. \quad (3)$$

When studying the operation of the prediction filtering algorithm for each of the detected objects, the ratio of the planes of the rectangles shown in Fig. 4.

Figure 4 – Intersection of rectangles

$$AP = \int_0^1 p(r)dr. \quad (4)$$

Precision and recall values are normalized in range from 0 up to 1 that leads to AP being between values of 0 and 1 also. In additions it is a common practice to smooth zigzag pattern before calculating AP.

At every recall level, the precision value is adjusted to match the highest precision observed at any subsequent recall level on the graph:

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}). \quad (5)$$

The curve spans all recall values (r_1, r_2, …), with a drop occurring each time the maximum precision value decreases. Such method provides exact area under pr-curve after pre-processing steps. With this change, we are measuring the exact area under the precision-recall curve after the zigzags are removed. General AP definition matches the definition of discrete integration of pr-curve.

There’s no need for approximation or interpolation. We calculate AP by sampling $p(r_i)$ at each drop and summing the resulting rectangular areas:

$$AP = \sum (r_{n+1} - r_n) \cdot p_{interp}(r_{n+1}), \quad (6)$$

$$p_{interp}(r_{n+1}) = \max_{\tilde{r} \geq r_{n+1}} p(\tilde{r}). \quad (7)$$

The generalized index of precision and recall for all varieties of studied object classes is higher for YOLOv8 compared to YOLOv5 and is 0.540 versus 0.526, respectively.

The results of the study of RC-metrics of the models, which characterize the change in the recall and confidence of the parameters of the YOLOv5 and YOLOv8 models are presented in Fig. 5.

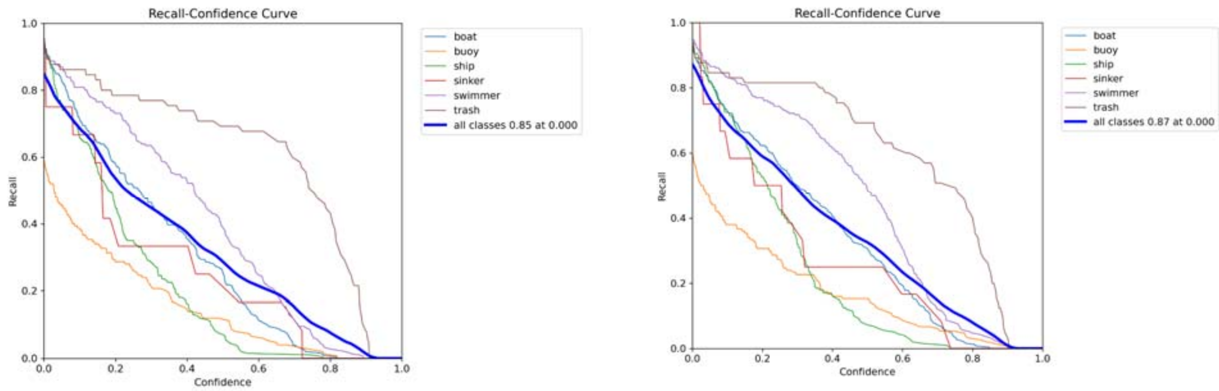
The generalized indices of recall and confidence for all varieties of the studied object classes have roughly equivalent values for YOLOv8 compared to YOLOv5 and are 0.87 versus 0.85, respectively.

The results of the study of PC-metrics of the models, which characterize the change in the precision and confidence of the parameters of the YOLOv5 and YOLOv8 models are presented in Fig. 6.

DOI 10.15588/1607-3274-2024-4-13

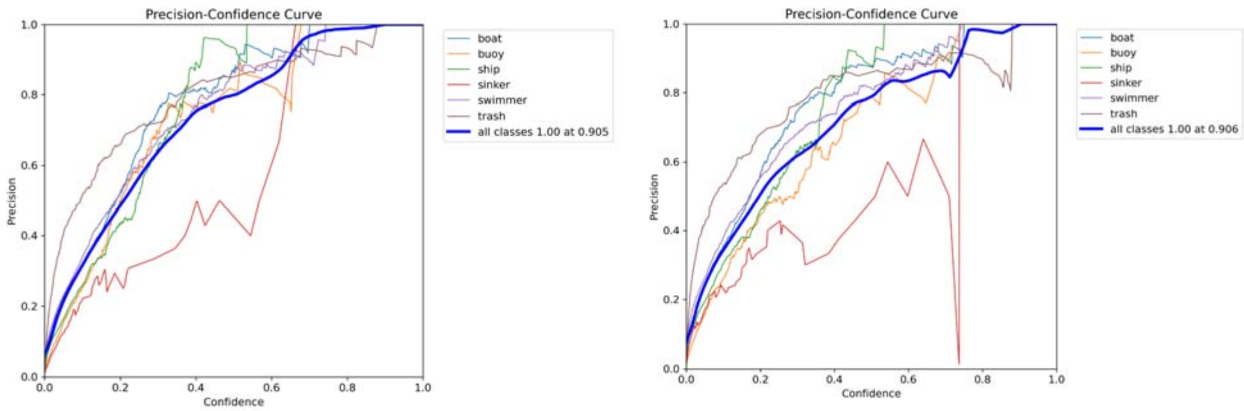
The generalized indices of precision and confidence for all varieties of the studied object classes have roughly equivalent values for YOLOv8 compared to YOLOv5 and are 0.906 versus 0.905, respectively.

The confusion matrices shown in Fig. 7 a and b, illustrate the percentage with which an object of a certain class will be determined (classified) by an object of another class for the YOLOv5 and YOLOv8 models, respectively.



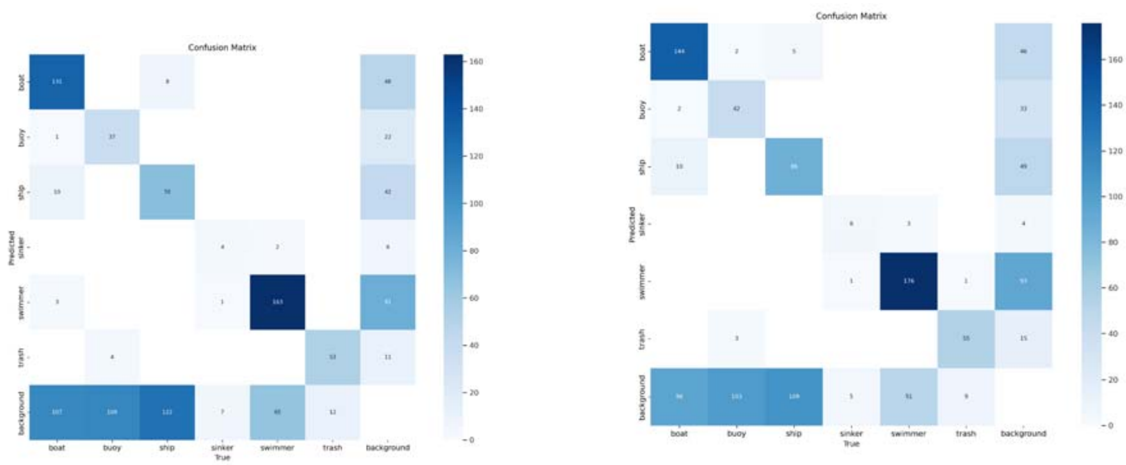
a b

Figure 5 – RC-metrics of models NN
a – YOLOv5; b – YOLOv8



a b

Figure 6 – PC-metrics of models NN
a – YOLOv5; b – YOLOv8



a b

Figure 7– Confusion matrix for NN
a – YOLOv5; b – YOLOv8

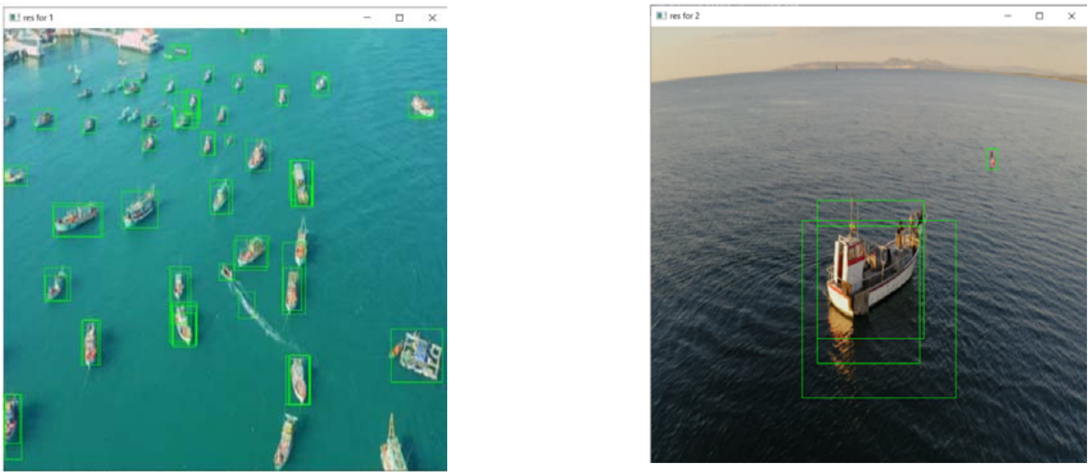
DOI 10.15588/1607-3274-2024-4-13

Examining in more detail the confusion matrices for the corresponding neural network models for each of the types of images recognized by objects, we get: the number of correctly recognized images of boats using YOLOv5 is significantly less than YOLOv8 and is 131 compared to 144 (9% difference); the number of correctly identified buoys is significantly different, 37 versus 42 (almost 12%); ships are recognized in the number of 70 to 86 (a large difference – about 19%); the number of images of drowning people is 4 compared to 6 (a very large difference of 33%), which is due to the small (unrepresentative) volume of the studied images; the number of photos of correctly recognized swimmers is 163 against 176 (7%); recognition of garbage images was performed with a slight difference of 53 vs. 55 (almost 4%), which leads to greater reliability and reliability of using the YOLOv8 neural network model compared to YOLOv5.

4 EXPERIMENTS

Examples below show classification process with reduced accuracy threshold down to 0.1 and with classification task dismissed. Recognized objects located correctly without explicit classification examples.

Test results depicted at Fig. 8.



a b

Figure 8 – Reduced accuracy threshold test results with classification function turned off
a – YOLOv5; b – YOLOv8

For test purposes, images were mostly marked correctly without exceptions dependent on object scale. It can be seen that model detects multiple objects at areas of actual objects. That type of behavior may be changed with implementation of bounding boxes filtering algorithm. Such algorithm reacts on a large cluster of overlapping detected objects, expanding single bound up to the borders of the biggest overlapping container so, grouping object cluster into one large object.

Selective object allocation pattern can be seen with objects being closer to the imaginary middle horizontal line being recognized better than others sticking closer to the top and bottom of the image. Such phenomenon can be explained with train dataset nature, where most of objects were placed at horizon level. In advance it can be seen that false object recognition is completely absent.

For the neural network, the training process of which is described in the corresponding section, it was necessary to write means for forming the on-board flight log and reporting on the found objects. As already mentioned in the analysis section, the Ultralytics API was chosen to interact with the model. It contains varitety of modules for dataset gathering, processing and filtering and some modules implement functionality connected with NN training. API depends on external libraries for some of its functions that installs automatically while using pip. Thus, the script that processes data in OpenCV image format looks like this:

```
import numpy as np
import cv2
from ultralytics import YOLO
yolo5_file = "./models/best5.pt"
yolo8_file = "./models/best8.pt"
under_test = YOLO(yolo5_file)
test_set =
[cv2.resize(cv2.imread(f"test_images/{i}.pn
g",cv2.IMREAD_COLOR),(640,640)) for for
i,data in enumerate(test_set):
r = under_test(data,imgsz = 640,conf=0.1)
print(r)
for s in r:
print(f"\n\nIMAGE #{i}")
boxes = s.boxes.xyxy.tolist()
for box in boxes:
box = list(map(int,box))
print(box)
test_set[i] =
cv2.rectangle(test_set[i],pt1=(box[0],box[1
]),pt2=(box[2],box[3]),color=(0,0,255),thic
kness=cv2.imshow('hello',test_set[i])
cv2.waitKey(0)
# closing all open windows
cv2.destroyAllWindows()
```

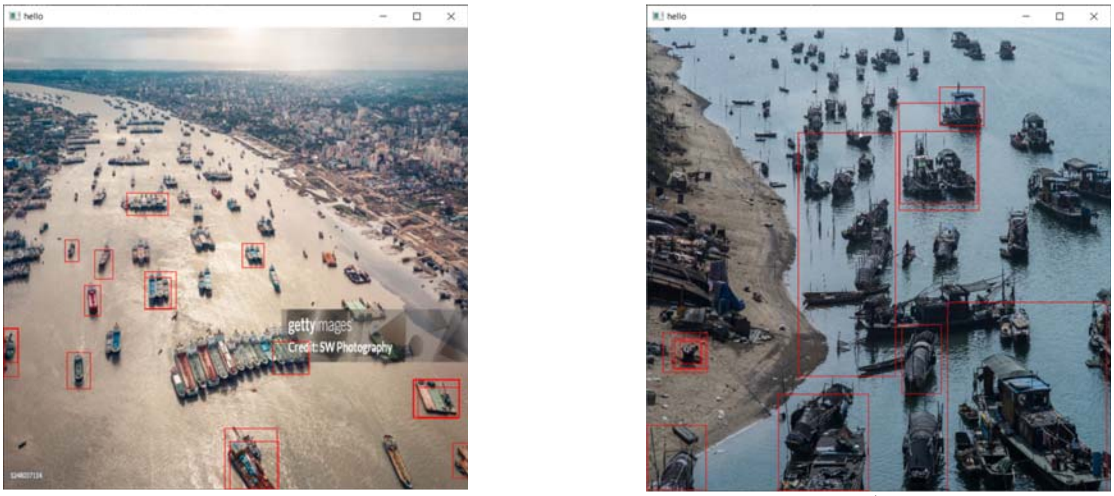

DOI 10.15588/1607-3274-2024-4-13

Fig. 9 a and b show images obtained after neural network processing. Certain objects are surrounded by several rectangles. This occurs due to the lowering of the accuracy threshold, as a result of which one object is classified into different classes. This is usually remedied by better training, but for the reasons mentioned earlier, another solution must be devised. This can be the implementation of a filter for post-processing of predictions of a neural network that will expand the boundaries of an existing rectangle in the case of a sufficient percentage of overlapping planes or create a new one in another case.

Fig. 10 a and b illustrate the uncertainty of the model regarding object ownership. We can see how due to the uncertainty of the model regarding the belonging of the object to a certain class, double allocation of objects appears. This can be avoided in two ways: retrain the model on a new dataset (which is ineffective for the given conditions of solving the problem) or write a filter that will process the selected objects and combine rectangles in an elementary way.

The specified block diagram of the algorithm performs the functions of filtering objects obtained after recognition by the neural network, namely, it analyzes the presence of an intersection of the limiting border in the form of a rectangle with a similar one for the case of the inability of the neural network to qualitatively classify this object.

Fig. 11 The block diagram of the algorithm shown in the figure 25 performs the functions of filtering objects obtained after recognition by the neural network, namely, it analyzes the presence of an intersection of the limiting border in the form of a rectangle with a similar one for the case of the inability of the neural network to qualitatively classify this object.



a b

Figure 9 – Images obtained after neural network processing

a b

Figure 10 – An illustration of the model's uncertainty about whether an object belongs to a certain class

DOI 10.15588/1607-3274-2024-4-13

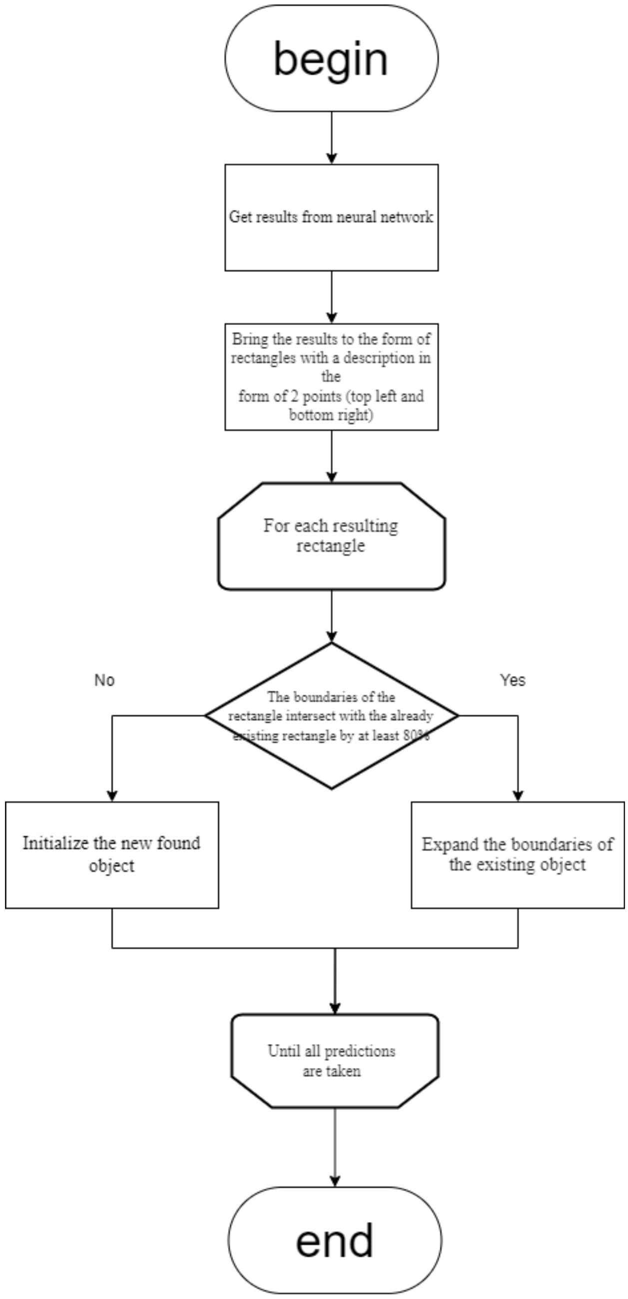
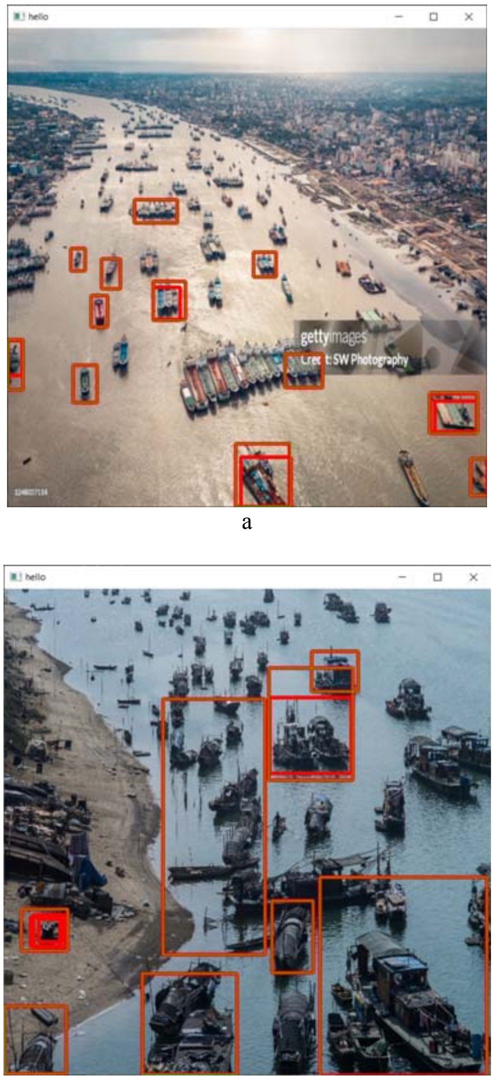


Figure 11 – Block diagram of the post-processing algorithm

The computational complexity of the algorithm was O(n). since the algorithm iterates through all received predictions, the memory complexity is O(log(n)) on average, O(n) in the worst case, because the resulting number of predictions either decreases or remains constant.

As a result of the implementation of the proposed algorithm, the clusters detected by the neural network are expanded. The set of rectangles in the previous step for certain objects is combined into one, thus we get rid of redundant predictions. The specified modification of the process of recognizing objects against the background of a reservoir, designed to eliminate redundancy in the prediction process, led to changes in the program script.



a

b

Figure 12 – Combining and filtering groupings from multiple predictions

As a result of the functioning of the proposed script, the number of recognized images was reduced. Fig. 12 a and b show how groups from several predictions are combined and filtered. Unprocessed predictions are marked in red, combined using the proposed algorithm and implemented using the created script are in green.

Original predictions are marked in red, post-processed ones in green. Test results reuire storage in some kind of logging system in order to perform system evaluation tests these. TinyDb – built-in Python module was used to secure image pathes in a single file. Mentioned tiny-db module does not support image storage unlike more complex systems such as PostgreSQL but this limitation may be avoided by storing file path in log file separately from the real extracted image. Limitations, provided in AI training paragraph, tell that original images also must be stored since such data may be used for gathering new dataset.

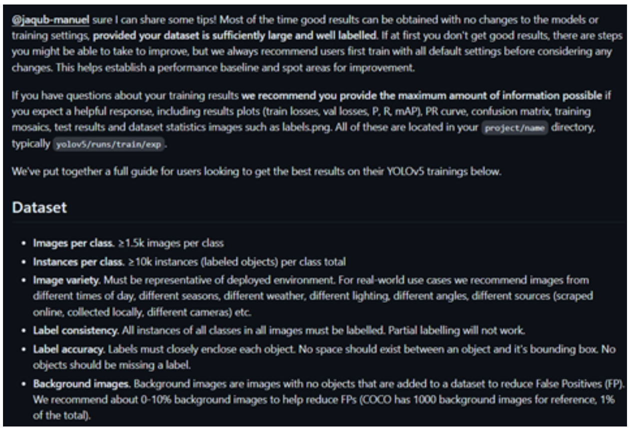
The general image processing algorithm was adjusted by implementing a filter to postprocess neural network

DOI 10.15588/1607-3274-2024-4-13

predictions, which improved prediction results by reducing the number of duplicate recognitions of the same objects. Scripts are offered that implement the specified post-processing and allow you to fix log directories of recognized objects and input images.

5 RESULTS

Gathering all previously said we can come to a conclusion objects detection part of the system works fine with exception of classification part being completely non-functional. Explanation for this involves dataset architecture, data sources, labelling process. These issues are common among API users and one of authors commented on ways of solving them. It is said that gathering bigger dataset with relevant data and proper labeling may reduce rate of misclassification cases.



@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).

Described system can be used with few extensions: as a originally designed system without function of object classification and also as a system for gathering relevant data for further implementation in a form of new train dataset since the current dataset contains of images taken from sailor boat and cannot perfectly fit goals of UAV reconnaissance.

A block diagram of the algorithm and corresponding scripts are proposed, which perform the functions of filtering the objects obtained after recognition by the neural network, namely, analyze the presence of an intersection of the limiting border in the form of a rectangle with a similar one for the case of the inability of the neural network to qualitatively classify this object. As a result of the implementation of the proposed algorithm, the clusters detected by the neural network are expanded. The set of rectangles in the previous step for certain objects is combined into one, thus we get rid of redundant predictions. For dataset gathering purposes logging subsystem with usage of tinydb was introduced.

The general image processing algorithm was adjusted by implementing a filter to postprocess neural network predictions, which improved prediction results by reducing the number of duplicate recognitions of the same objects. Scripts are offered that implement the specified post-processing and allow you to fix log directories of recognized objects and input images.

CONCLUSIONS

The issue of reoccuring object recognition, caused by lowering the neural network's accuracy threshold, has been addressed. An algorithm for filtering predictions generated by a neural network, which excludes duplication of predictions, was proposed and tested, and a cut-off threshold was determined at which the object would be detected, but there would be no duplication. This achieves preservation of detection quality while eliminating duplication.

The scientific novelty consists in adding post-processing stage to NN predictions pipeline, which is based on an algorithm similar to the filtering algorithm already implemented in NN, which supresses NN-provided reoccuring object detections with provided algorithm based on image area intersection.

The practical significance of obtained results is implementation of post-processing algorithm and the created script based on it, a result was obtained where groups from several predictions are combined and filtered. As a result, these findings are saved in the database in the form they were obtained.

Prospects for further research are to improve the system by using UAVs as mobile cameras that will send video data to a central station where a sophisticated detector trained on relevant data will be deployed. This will make it possible to reduce the load on the UAV (both physical and informational), as well as reduce the cost of the UAV by the cost of the platform for deploying AI and move from the perspective of considering the UAV as a direct executor of the mission to the perspective of using it as a manipulator for the human-system AI.

ACKNOWLEDGEMENTS

This article was made possible thanks to Oleksandr Rolik and Mykola Shynkevych, staff members of the Department of Information Systems and Technologies at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

REFERENCES

1. Schunfelder P., Stebel F., Andreou N., Kunig M. Deep learning-based text detection and recognition on architectural floor plans, *Automation in Construction,* 2024, № 157. pp. 105–156. DOI: 10. 1016/j.autcon.2023.105156 (date of access: 06.05.2020)..
2. Giakoumoglou N., Pechlivani E. M., Tzovaras D. Generate-paste-blend-detect: Synthetic dataset for object detection in the agriculture domain, S*mart Agricultural Technology,* 2023, №5, pp. 100–105. DOI:10.1016/j.atech.2023.100258.
3. Ashourpour M., Azizpour G., Johansen K. Real-time defect and object detection in assembly line: A case for in-line quality inspection, *Lecture Notes in Mechanical Engineering,* 2024, pp. 99–106. DOI: 10.1007/978-3-031-38241-3_12.
4. Azevedo P., Santos V. Comparative analysis of multiple yolo-based target detectors and trackers for adas in

DOI 10.15588/1607-3274-2024-4-13

edge devices, *Robotics and Autonomous Systems*, 2024, № 171, pp. 87–95. DOI: 10.1016/j.robot.2023.104558.
5. Sanjai Siddharthan M., Aravind S., Sountharrajan S. Real-time road hazard classification using object detection with deep learning, *Lecture Notes in Networks and Systems*, 2024, № 789, LNNS, pp. 479– 492. DOI: 10.1007/978-981-99-6586-1_33.
6. Wei Z., Zhang Y., Wang X., Zhou J., Dou F., Xia Y. A yolov8-based approach for steel plate surface defect detection, *Metalurgija,* 2024, № 63, pp. 28–30. DOI: 10.3390/app14125325
7. Wu F., Zhang Y., Wang L., Hu Q., Fan S., Cai W. A deep learning-based lightweight model for the detection of marine fishes, *Journal of Marine Science and Engineering,* 2023, №11, pp. 19–34. DOI: 10.3390/jmse11112156.
8. Zhang G., Tang Y., Tang H., Li W., Wang L. A global lightweight deep learning model for express package detection, *Journal of Intelligent and Fuzzy Systems*, 2023, № 45. pp. 12013–12025. DOI: 10.3233/JIFS-232874.
9. Wang J., Dai H., Chen T., Liu H., Zhang X., Zhong Q., Lu R. Toward surface defect detection in electronics manufacturing by an accurate and lightweight yolo-style object detector, *Scientific Reports*, 2023, № 13, pp. 7–26. DOI: 10.1038/s41598-023-33804-w.
10. Li A., Zhang Z., Sun S., Feng M., Wu C. Multinet-gs: Structured road perception model based on multi-task convolutional neural network, *Electronics (Switzerland),* 2023, № 12 (19), pp. 39–59. DOI: 10.3390/electronics12193994.
11. Han L., Ma C., Liu Y., Jia J., Sun J. Sc-yolov8: A security check model for the inspection of prohibited items in x-ray images, *Electronics (Switzerland),* 2023, № 12 (20), pp. 208–223. DOI: 10.3390/electronics12204208.
12. Mao J., Wang L., Wang N., Hu Y., Sheng W. A novel method of human identification based on dental impression image, *Pattern Recognition,* 2023, № 144, pp. 109–126. DOI: 10.1016/j.patcog.2023.109864.
13. Kara E., Zhang G., Williams J. J., Ferrandez-Quinto G., Rhoden L. J., Kim M., Kutz J. N., Rahman A. Deep learning based object tracking in walking droplet and granular intruder experiments, *Journal of Real-Time Image Processing*, 2023, № 20, pp. 269–311 DOI: 10.1007/s11554-023-01341-4.
14. Zhou S., Zhong M., Chai X., Zhang N., Zhang Y., Sun Q., Sun T. Framework of rod-like crops sorting based on multi-object oriented detection and analysis, *Computers and Electronics in Agriculture,* 2024, № 216, pp. 108–145. DOI: 10.1016/j.compag.2023.108516
15. Shan P., Yang R., Xiao H., Zhang L., Liu Y., Fu Q., Zhao Y. Uavpnet: A balanced and enhanced uav object detection and pose recognition network, *Measurement: Journal of the International Measurement Confederation*, 2023, № 222, pp. 113–132. DOI: 10.1016/j.measurement.2023.113654.
16. Talaat F. M., ZainEldin H. An improved fire detection approach based on yolov8 for smart cities, *Neural Computing and Applications,* 2023, № 35, pp. 20939–20954. DOI: 10.1007/s00521-023-08809-1.
17. Liu S., Fan Q., Zhao C., Li S. Rtad: A real-time animal object detection model based on a large selective kernel and channel pruning, *Information (Switzerland),* 2023, № 14 (10), pp. 535–550 DOI: 10.3390/info14100535.
18. Smolij V. About features of management preproduction of electronic vehicles, *Problems of Modeling and Design Automatization,* 2019, № 11, pp. 33–42. DOI: 10.31474/2074-7888-2019-1-33-42.
19. Su Y., Tan W., Dong Y., Xu W., Huang P., Zhang J., Zhang D. Enhancing concealed object detection in active millimeter wave images using wavelet transform, *Signal Processing*, 2024, № 216, pp. 109–122. DOI: 10.1016/j.sigpro.2023.109303.
20. Liu C., Wang K., Li Q., Zhao F., Zhao K., Ma H. Powerful-iou: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism, *Neural Networks*, 2024, № 170, pp. 276–284. DOI: 10.1016/j.neunet.2023.11.041.
21. Xu W., Liu C., Wang G., Zhao Y., Yu J., Muhammad A., Li D. Behavioral response of fish under ammonia nitrogen stress based on machine vision, *Engineering Applications of Artificial Intelligence,* 2024, № 128, pp. 107–134. DOI: 10.1016/j.engappai.2023.107442.
22. Dimauro G., Barbaro N., Camporeale M. G., Fiore V., Gelardi M., Scalera M. Deepcilia: Automated, deep-learning based engine for precise ciliary beat frequency estimation, *Biomedical Signal Processing and Control,* 2024, № 90, pp. 1002–1019. DOI: 10.1016/j.bspc.2023.105808.
23. Zhao X., Song Y. Improved ship detection with yolov8 enhanced with mobilevit and Gsconv, *Electronics (Switzerland),* 2023, № 12(22), pp. 46–66. DOI: 10.3390/electronics12224666.
24. Smolij V. M., Smolij N. V., Sayapin S. P. Search and classification of objects in the zone of reservoirs and coastal zones, *CEUR Workshop Proceedings*, 2024, No. 3666, pp. 37–51. EID: 2-s2.0-85191443231
25. Hui J. mAP (mean Average Precision) for Object Detection. Medium. URL: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173 (date of access: 15.06.2024).

Received 25.06.2024.
Accepted 28.10.2024.

DOI 10.15588/1607-3274-2024-4-13

УДК 681.325

ПОСТ-ОБРОБКА ПРОГНОЗІВ ДЛЯ ПОЛІПШЕННЯ ЯКОСТІ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ПОВЕРХНІ ВОДОЙМ

Смолій В. М. – д-р техн. наук, професор, професор кафедри інформаційних систем та технологій Національного університету біоресурсів і природокористування України, Київ, Україна.

Смолій Н. В. – студент кафедри інформаційних систем та технологій Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Мокрієв М.В. – канд. екон. наук, доцент, доцент кафедри інформаційних систем та технологій Національного університету біоресурсів і природокористування України, Київ, Україна.

АНОТАЦІЯ

Актуальність роботи обумовлена масштабним поширенням технологій штучного інтелекту на процес виявлення і детекції об'єктів на поверхні водойм за допомогою БПЛА. Сучасні потреби в моніторингу водойм, особливо в контексті екологічного нагляду, охорони та управління ресурсами, вимагають точних і надійних рішень. Ця робота демонструє методи покращення роботи нейронних мереж і пропонує підходи до обробки передбачень НМ, навіть якщо вони натреновані на нерелевантних даних, що підвищує універсальність і ефективність технології.

Мета роботи – вирішення проблеми помилкового розпізнавання об'єктів на поверхні водойм, що обумовлено зменшенням порогу точності для нейронної мережі. Це забезпечує точнішу і надійнішу детекцію, зменшуючи кількість хибнопозитивних передбачень і підвищуючи ефективність системи загалом.

Метод. Запропоновано додавання етапу постобробки передбачень НМ, який базується на алгоритмі, схожому на вже впроваджений в НМ алгоритм фільтрації. Цей алгоритм пригнічує повторне виявлення об'єкта мережею та спирається на площі перетину виявлених прямокутників. Він використовує порогове значення 0.8 для двох точок прямокутника, що дозволяє ефективно знижувати кількість повторних передбачень та покращувати точність.

Результати. В результаті впровадження запропонованого алгоритму і створеного на його основі скрипта було досягнуто результат, при якому угруповання з кількох передбачень поєднуються та фільтруються. Отримані дані зберігаються в базі даних як знайдені та детектовані об'єкти. Запропонований алгоритм постобробки ефективно усуває надлишкові передбачення, при цьому зберігаючи точність прогнозу. Це забезпечує надійність системи і підвищує її продуктивність в реальних умовах.

Висновки. Детектовані зображення об'єктів на поверхні водойм зберігаються в базі даних у вигляді записів з унікальними ідентифікаторами файлових імен. Завдяки запропонованому алгоритму постобробки зображень формуються журнали для відпрацьованої місії з використанням тестових зображень, що забезпечує виключення можливості дублювання інформації. Це підвищує ефективність та надійність системи моніторингу, забезпечуючи точну і своєчасну детекцію об'єктів на поверхні водойм.

КЛЮЧОВІ СЛОВА: БПЛА, детекція, розпізнавані об'єкти, поверхня водойм, нейронна мережа, набір даних, модель, розподіл зображень, матриця помилок, навчальні метрики, збільшення, мозаїчне розміщення зображень, постобробка зображень, скрипт реалізації, журнал відпрацьованої місії, база даних.

ЛІТЕРАТУРА

1. Deep learning-based text detection and recognition on architectural floor plans / [P. Schunfelder, F. Stebel, N. Andreou, M. Kunig] // Automation in Construction. – 2024. – № 157. – P. 105–156. DOI: 10. 1016/j.autcon.2023.105156 (date of access: 06.05.2020)..
2. Giakoumoglou N. Generate-paste-blend-detect: Synthetic dataset for object detection in the agriculture domain / N. Giakoumoglou, E. M. Pechlivani, D. Tzovaras // Smart Agricultural Technology. – 2023. – № 5. – P. 100–105. DOI: 10.1016/j.atech.2023.100258.
3. Ashourpour M. Real-time defect and object detection in assembly line: A case for in-line quality inspection / M. Ashourpour, G. Azizpour, K. Johansen // Lecture Notes in Mechanical Engineering. – 2024. – P. 99–106. DOI: 10.1007/978-3-031-38241-3_12.
4. Azevedo P. Comparative analysis of multiple yolo-based target detectors and trackers for adas in edge devices / P. Azevedo, V. Santos // Robotics and Autonomous Systems. – 2024. – № 171. – P. 87–95. DOI: 10. 1016/j.robot.2023.104558.
5. Sanjai Siddharthan M. Real-time road hazard classification using object detection with deep learning/ M. Sanjai Siddharthan, S. Aravind, S. Sountharrajan // Lecture Notes in Networks and Systems. – 2024. – № 789. LNNS. – P. 479–492. DOI: 10.1007/978-981-99-6586-1_33.
6. A yolov8-based approach for steel plate surface defect detection / [Z. Wei, Y. Zhang, X. Wang et al.] // Metalurgija. – 2024. – № 63. – P. 28–30. DOI: 10.3390/app14125325
7. A deep learning-based lightweight model for the detection of marine fishes/ [F. Wu, Y. Zhang, L. Wang et al.] // Journal of Marine Science and Engineering. – 2023. – № 11. – P. 19–34. DOI: 10.3390/jmse11112156.
8. A global lightweight deep learning model for express package detection / [G. Zhang, Y. Tang, H. Tang et al.] // Journal of Intelligent and Fuzzy Systems. – 2023. – № 45. – P. 12013–12025. DOI: 10.3233/JIFS-232874.
9. Toward surface defect detection in electronics manufacturing by an accurate and lightweight yolo-style object detector/ [J. Wang, H. Dai, T. Chen et al.] // Sci-

DOI 10.15588/1607-3274-2024-4-13

entific Reports. – 2023. – № 13. – P. 7–26. DOI: 10.1038/s41598-023-33804-w.
10. Multinet-gs:Structured road perception model based on multi-task convolutional neural network/ [A. Li, Z. Zhang, S. Sun et al.] // Electronics (Switzerland). – 2023. – № 12 (19). – P. 39–59. DOI: 10.3390/electronics12193994.
11. Sc-yolov8: A security check model for the inspection of prohibited items in x-ray images/ [L. Han, C. Ma, Y. Liu et al.] // Electronics (Switzerland). – 2023. – № 12 (20). – P. 208–223. DOI: 10.3390/electronics12204208.
12. A novel method of human identification based on dental impression image/ [J. Mao, L. Wang, N. Wang et al.] //Pattern Recognition. – 2023. – № 144. – P. 109 – 126. DOI: 10.1016/j.patcog.2023.109864.
13. Deep learning based object tracking in walking droplet and granular intruder experiments/ [E. Kara, G. Zhang, J. J. Williams et al.] // Journal of Real-Time Image Processing. – 2023.– № 20.– P. 269–311 DOI: 10.1007/s11554-023-01341-4.
14. Framework of rod-like crops sorting based on multi-object oriented detection and analysis/ [S. Zhou, M. Zhong, X. Chai et al.] //Computers and Electronics in Agriculture. – 2024. – № 216. – P. 108–145. DOI: 10.1016/j.compag.2023.108516
15. Uavpnet: A balanced and enhanced uav object detection and pose recognition network/ [P. Shan, R. Yang, H. Xiao et al.] // Measurement: Journal of the International Measurement Confederation. – 2023. – № 222. – P. 113–132. DOI: 10.1016/j.measurement.2023.113654.
16. Talaat F. M. An improved fire detection approach based on yolov8 for smart cities/ F. M. Talaat, H. ZainEldin // Neural Computing and Applications. – 2023. – № 35. – P. 20939–20954. DOI: 10.1007/s00521-023-08809-1.
17. Rtad: A real-time animal object detection model based on a large selective kernel and channel pruning / [S. Liu, Q. Fan, C. Zhao, S. Li] // Information (Switzerland). – 2023. – № 14 (10). – P. 535–550. DOI: 10.3390/info14100535.
18. Smolij V. About features of management preproduction of electronic vehicles / V. Smolij // Problems of Modeling and Design Automatization. – 2019. – № 11. – P. 33–42. DOI: 10.31474/2074-7888-2019-1-33-42.
19. Enhancing concealed object detection in active millimeter wave images using wavelet transform / [Y. Su, W. Tan, Y. Dong et al.] // Signal Processing. – 2024. – № 216. – P. 109–122. DOI: 10.1016/j.sigpro.2023.109303.
20. Powerful-iou: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism/ [C. Liu, K. Wang, Q. Li et al.] // Neural Networks. – 2024. – № 170. – P. 276–284. DOI: 10.1016/j.neunet.2023.11.041.
21. Behavioral response of fish under ammonia nitrogen stress based on machine vision / [W. Xu, C. Liu, G. Wang et al.] // Engineering Applications of Artificial Intelligence. – 2024. – № 128. – P. 107–134. DOI: 10.1016/j.engappai.2023.107442.
22. Deepcilia: Automated, deep-learning based engine for precise ciliary beat frequency estimation/ [G. Dimauro, N. Barbaro, M. G. Camporeale et al.] // Biomedical Signal Processing and Control. – 2024. – № 90. – P. 1002–1019. DOI: 10.1016/j.bspc.2023.105808.
23. Zhao X. Improved ship detection with yolov8 enhanced with mobilevit and Gsconv / X. Zhao, Y. Song // Electronics (Switzerland). – 2023. – № 12(22). – P. 46–66. DOI: 10.3390/electronics12224666.
24. Smolij V. M. Search and classification of objects in the zone of reservoirs and coastal zones / V. M. Smolij, N. V. Smolij, S. P. Sayapin // CEUR Workshop Proceedings – 2024. – No. 3666. – P. 37–51. EID: 2-s2.0-85191443231
25. Hui J. mAP (mean Average Precision) for Object Detection. Medium. URL: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173 (date of access: 15.06.2024).

DOI 10.15588/1607-3274-2024-4-13