JEE Journal of Ecological Engineering

Journal of Ecological Engineering 2024, 25(4), 360–369
https://doi.org/10.12911/22998993/185317
ISSN 2299–8993, License CC-BY 4.0

Received: 2024.02.03
Accepted: 2024.02.24
Published: 2024.03.01

Microscopic Studies of Activated Sludge Supported by Automatic Image Analysis Based on Deep Learning Neural Networks

Marcin Dziadosz[1], Dariusz Majerek[1], Grzegorz Łagód[2*]

[1] Department of Applied Mathematics, Faculty of Mathematics and Information Technology, Lublin University of Technology, ul. Nadbystrzycka 38, 20-618 Lublin, Poland

[2] Department of Water Supply and Wastewater Disposal, Faculty of Environmental Engineering, Lublin University of Technology, ul. Nadbystrzycka 40B, 20-618 Lublin, Poland

* Corresponding author's e-mail: g.lagod@pollub.pl

ABSTRACT

Paper presents microscopic studies of activated sludge supported by automatic image analysis based on deep learning neural networks. The organisms classified as *Arcella vulgaris* were chosen for the research. They frequently occur in the waters containing organic substances as well as WWTPs employing the activated sludge method. Usually, they can be clearly seen and counted using a standard optical microscope, as a result of their distinctive appearance, numerous population and passive behavior. Thus, these organisms constitute a viable object for detection task. Paper refers to the comparison of performance of deep learning networks namely YOLOv4 and YOLOv8, which conduct automatic image analysis of the afore-mentioned organisms. YOLO (You Only Look Once) constitutes a one-stage object detection model that look at the analyzed image once and allow real-time detection without a marked accuracy loss. The training of the applied YOLO models was carried out using sample microscopic images of activated sludge. The relevant training data set was created by manually labeling the digital images of organisms, followed by calculation and comparison of various metrics, including recall, precision, and accuracy. The architecture of the networks built for the detection task was general, which means that the structure of the layers and filters was not affected by the purpose of using the models. Accounting mentioned universal construction of the models, the results of the accuracy and quality of the classification can be considered as very good. This means that the general architecture of the YOLO networks can also be used for specific tasks such as identification of shell amoebas in activated sludge.

Keywords: activated sludge, automatic image analysis, deep learning, YOLO, Arcella vulgaris, biomarkers, bioindication.

INTRODUCTION

This paper is devoted to the automatic recognition and localization of activated sludge organisms from the municipal wastewater treatment plant. The paper presents the method of using automatic image analysis to identify the shell amoeba *Arcella vulgaris* among other organisms living in activated sludge. The analyzed species belong to, testate amoebae, and are a group mentioned in the context of assessing the stability of the operation of wastewater treatment plants with biological reactors, especially nitrification and denitrification processes [Fiałkowska et al., 2005; Pérez-Uz et al., 2010; Babko et al., 2023]. They are easy to observe and count under a microscope due to their characteristic appearance, large number and sedentary lifestyle. Thus, they are a good object for the detection task [Fiałkowska et al., 2005].

Automatic image analysis, also known as computer vision, belongs to the field of machine learning and artificial intelligence. It relies on extracting and processing data from digital images to be later interpreted. The main task of computer vision is recognizing and locating objects in the image and assigning them to relevant classes.

Computers "see" the world in a slightly different way than people. They interpret each image as two-dimensional arrays of numerical values, called pixels. This fact does not mean that there is no possibility of teaching them to recognize shapes and patterns. All what must be done is to figure out how a computer vision system can use numerical values to detect objects in an image and their characteristics, such as edges, colors, textures, sizes, and spatial arrangement [Girshick et al., 2014; Girshick, 2015; He et al., 2015].

The development of computer vision began in the mid-1990s. One of the first events in the field of artificial intelligence is considered to be the launch of the Summer Vision Project by Seymour Papert in 1966. The project aimed to create a special computer system that could be able to identify objects in the image. The implementation of this project consisted in manually specifying rules for detecting objects by programmers. The rules in early computer vision systems were typically based on specific patterns or features that programmer manually defined. These rules could include mathematical algorithms or logical conditions that helped the computer identify objects or characteristics in an image. For example, the rules could be created to detect edges by looking for rapid changes in pixel intensity values or to identify colors by comparing pixel values of predefined color thresholds. These rules were often derived from the knowledge and expertise of programmers and required a significant manual effort to be developed and fine-tuned. However, due to complexity and variability of real-world images, relying solely on manually specified rules proved to be limited in its efficiency [Papert, 2004].

In the 1970s and 1980s, researchers focused on developing algorithms to detect specific features like edges, corners, and textures in images. Techniques such as the Canny edge detector and the Harris corner detector were introduced. Scale-space theory, introduced by Witkin in the 1980s, aimed to analyze images at multiple scales to handle variations in object size and appearance. This theory led to the development of techniques like the Laplacian of Gaussian (LoG) and Difference of Gaussians (DoG) for blob detection. In the 1980s and 1990s, neural networks and machine learning gained popularity in computer vision.

Between 2000 and 2010, computer vision experienced several advancements and breakthroughs. Significant progress was made in object detection and recognition. The introduction of the Viola-Jones algorithm in 2001 enabled real-time face detection. Additionally, progress in the Histogram of Oriented Gradients (HOG) feature descriptor led to improved object detection in images. Various feature descriptors, including SIFT (Scale-Invariant Feature Transform) and SURF (Speeded-Up Robust Features) were introduced. These descriptors allowed for robust matching and recognition of objects across different views and images. The use of machine learning algorithms, particularly Support Vector Machines (SVMs) and Random Forests, became prevalent for image classification tasks. These algorithms, combined with large-scale datasets like ImageNet, significantly improved image classification accuracy [Viola and Jones, 2001; Lowe, 2004; Bay et al., 2008].

In 2012, the AlexNet convolutional neural network designed by Alex Krizhevsky won the ImageNet image recognition competition. This event generated a lot of interest and started a revolution in deep network learning. Since then, with the development of technology and greater access to information, the accuracy of object detection models has doubled and is still being improved [Redmon et al., 2016; Krizhevsky et al., 2017; Ren et al., 2017].

Computer vision is currently one of the fastest growing fields of machine learning and artificial intelligence. One of the main challenges of computer vision is to recreate the powerful abilities of the human visual system which consists not only in detecting and identifying objects in an image, but also in describing and understanding the scene enclosed in it. Automated image analysis algorithms are showing promising results in face recognition tasks and driving autonomous cars as well as agriculture, industry, sales, finance and healthcare [Stawarczyk and Stawarczyk, 2015; Wang and He, 2023]. Constant work on improving computer vision means that in the future this technology can perform an even wider range of functions and, in combination with artificial intelligence systems, create machines with human-like thinking and analyzing skills [Liu et al., 2018; Wang et al., 2020].

The convolutional neural network method was used to classify and locate the objects in digital images of microscopic samples of activated sludge. Automatic analysis can be used in the future to assess the condition of activated sludge and to determine the rate of changes in the population of organisms, just as it has already been applied in the case of flock structure parameters [Amaral et al., 2013; Babko et al., 2014]. Bioindicators in activated sludge range from several dozen to several hundred species of protozoa and metazoa, not including prokaryotes. The situation in the biofilm of technical beds may be even more complicated. Many of the species mentioned are more difficult to identify than *Arcella vulgaris* discussed in the work. The proposed algorithm is suitable for identifying other popular bioindicators of activated sludge, which was initially successfully tested by the authors of the study. However, in the case of supervised machine learning methods, the challenge is not only the appropriate algorithm for identifying objects, but often a qualified teacher to supervise the algorithm which must correctly identify thousands of visually very similar representatives of different species.

MATERIALS AND METHODS

The biological material used for the research was taken from the Hajdów municipal wastewater treatment plant in Lublin. This facility is one of the systems of mechanical and biological treatment plants that treat municipal sewage [Jaromin-Gleń et al., 2013].

The material for the tests (activated sludge) was collected twice a month for about one year period. The sampling was always done in the same way. Using a measuring vessel, 300 ml plastic containers were filled with activated sludge, up to a maximum of 150 ml per container. Half of the volume was left empty to provide an air supply during transport. The containers with the material prepared in this way were placed in a refrigerator at a temperature of 5 °C and then transported to the laboratory. The sampling and transport took no more than an hour, while the preparations for microscopic examination were conducted immediately after the samples were delivered. At least three in-vivo slices were prepared from each activated sludge sample. Using an automatic pipette (BIOHIT m 1000), the samples were applied to a primary glass which was covered with a cover glass. The specimens were observed with an Olympus CX41 optical microscope in transmitted light, in a bright field of view, using a ×10 objective and trinocular with a digital camera and no additional optical magnification.

For the purposes of research and analysis, the testate amoebas of the species A*rcella vulgaris* were selected (Figures 1-2).

The *Arcella vulgaris* shell is round, with a diameter of 50–150 μm and a height of 35–55 μm, and resembles a watch glass. The lower part of the shell is flat with a circular hole with a diameter of 20–30 μm in the middle, which is clear in Figure 1a. The hole is surrounded by a flange which is a continuation of the shell wall. The wall of the *Arcella vulgaris* shell consists of a monolayer of protein, mainly keratin, vesicles. It turns brown over time, but the level of its plasma filling is quite often incomplete in the shell, so visible empty space between plasma and shell inner boundary are formed. There are two nuclei, opposite each other. Pulsating vacuoles are usually numerous [Ogden and Hedley, 1980; Fiałkowska et al., 2005; Pérez-Uz et al., 2010].

For the discussed task, deep convolutional neural networks were used – YOLOv4 and YOLOv8. Figure 3 and Figure 4 present the architectures of YOLOv4 and YOLOv8, respectively. In YOLO (You Only Look Once) architectures, the terms “head,” “backbone,” and “neck” refer to different components of the network that play

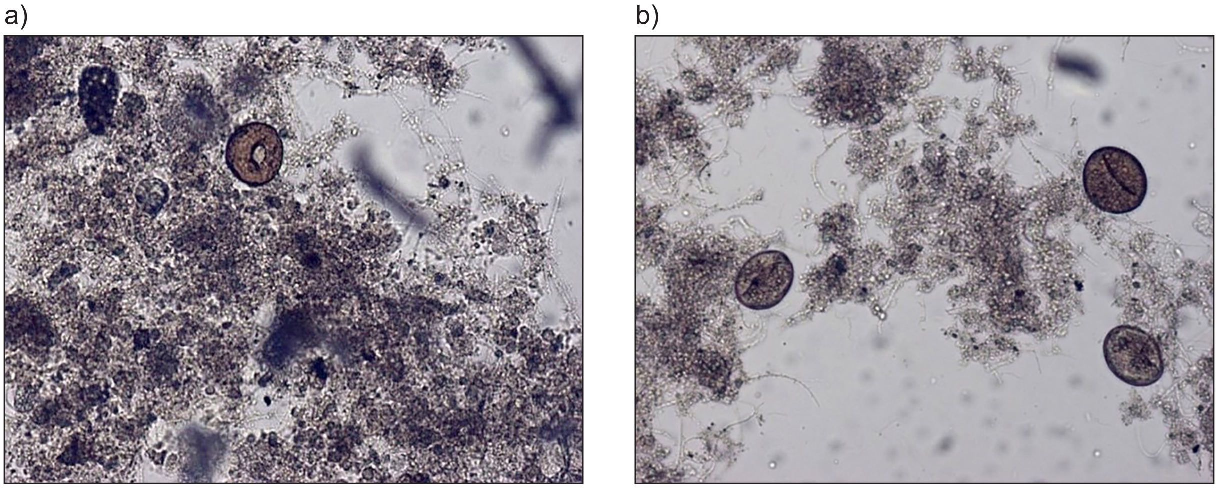

Figure 1. (a) Lower part of the shell with a round hole, (b) Three *Arcellas*

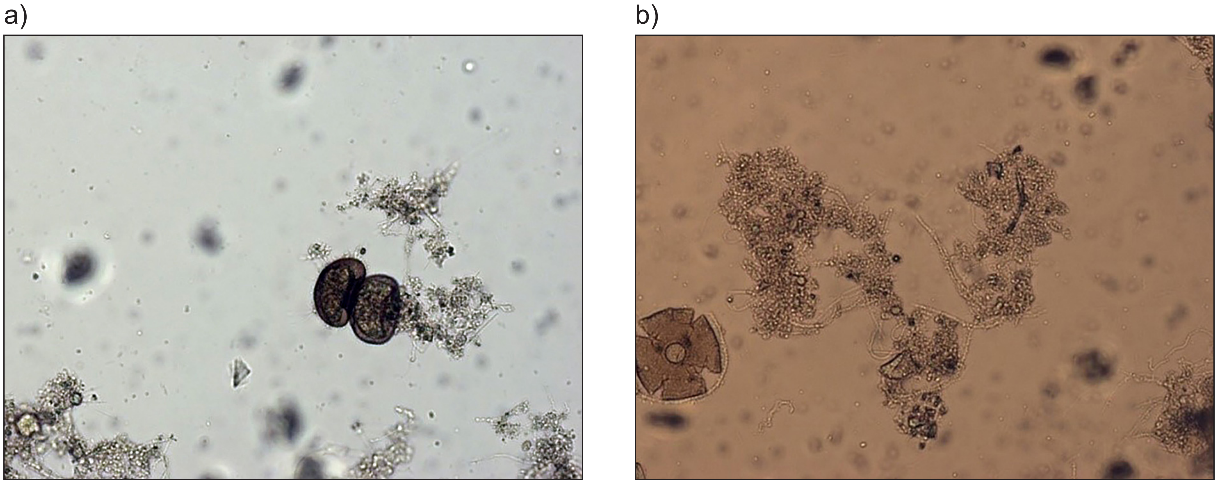

Figure 2. (a) Two *Arcellas* sideways, (b) broken shell

specific roles in the object detection process. The backbone network is responsible for extracting high-level features from the input image. It typically consists of multiple convolutional layers arranged in a deep neural network architecture. The backbone network learns to capture and represent various levels of visual information, starting from low-level features like edges and textures to higher-level semantic features. In YOLO architectures, the backbone network acts as the foundation for feature extraction. The neck network is an intermediate component that connects the backbone and head networks. Its main purpose is to enhance feature representation by aggregating features from multiple scales. The neck network helps capture multi-scale information, which is crucial for detecting objects of different sizes. It typically includes bottom-up and top-down pathways, lateral connections, and feature fusion modules. The neck network in YOLO architectures, such as PANet (Path Aggregation Network), aids in improving the accuracy of object detection by incorporating multi-scale features. The head network is responsible for generating the final predictions for object detection, including bounding box coordinates, class probabilities, and objectness scores. It takes the features extracted by the backbone network and refined by the neck network and processes them to produce detection results. The head network typically consists of multiple detection layers, each responsible for detecting objects at different scales. These detection layers use anchor boxes, which are predefined boxes of various sizes and aspect ratios, to localize and classify objects within the image. The head network in YOLO architectures, such as YOLOv3, combines the features from different scales to make accurate predictions for object detection [Song et al., 2023]. Overall, the backbone, neck, and head components in YOLO architectures work together to extract features, enhance representation, and generate predictions for object detection tasks.

YOLO is a popular real-time object detection algorithm. It revolutionized the field of computer vision by introducing a single-stage detection approach where object detection is performed in a single pass through the network.

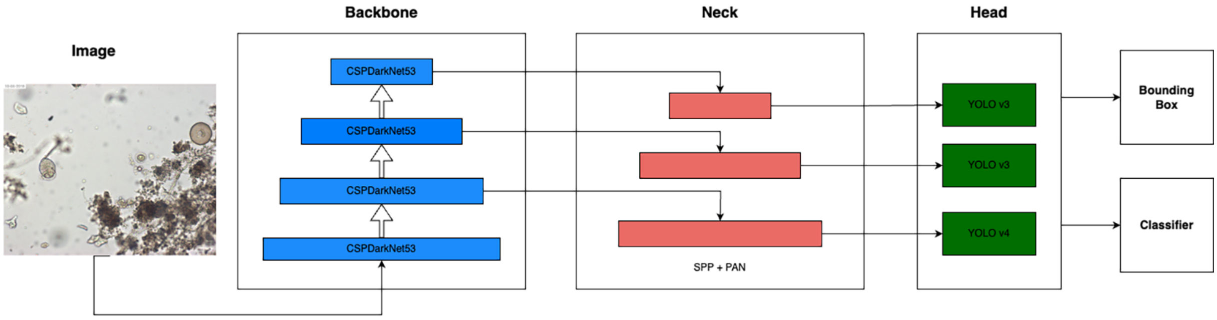

Figure 3. YOLOv4 network architecture (image source: own work)

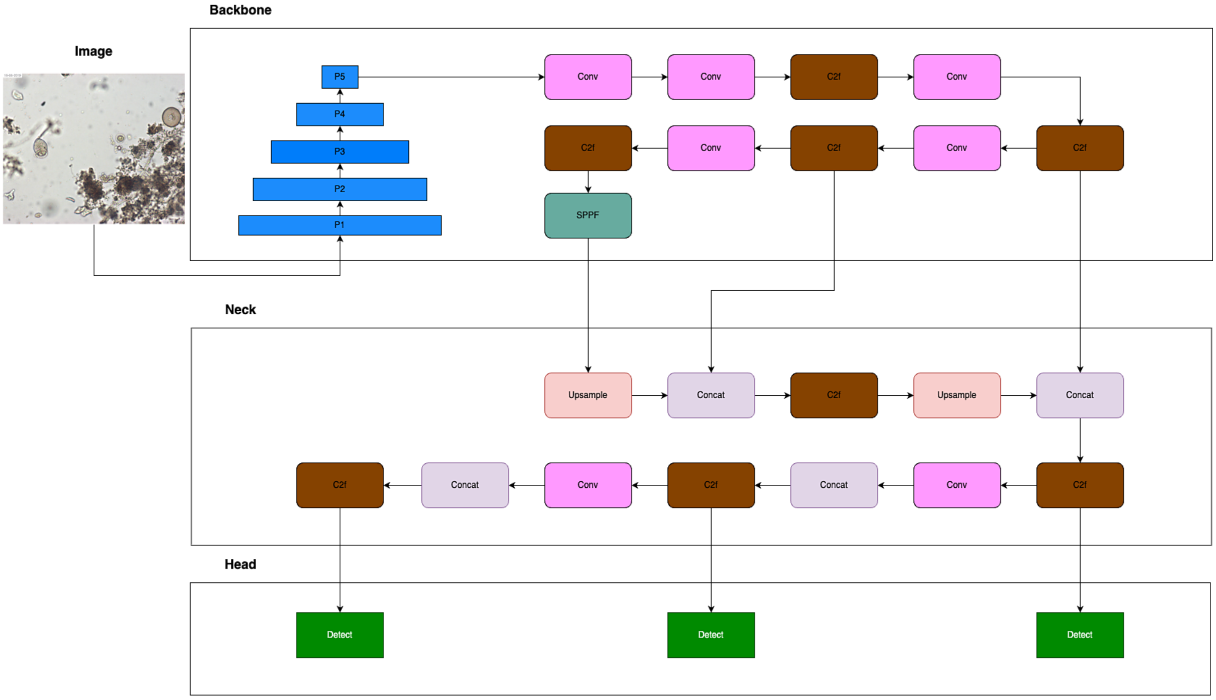

Figure 4. YOLOv8 network architecture (image source: own work)

Unlike traditional object detection methods like R-CNN, Fast R-CNN and Faster R-CNN [Lowe, 2004] that require multiple stages, YOLO performs object detection in a single step. It divides an input image into a grid and predicts bounding boxes and class probabilities for each grid cell. For each grid cell, YOLO predicts multiple bounding boxes that enclose objects in that cell. These bounding boxes consist of coordinates for the box's center, width (), height (), and confidence scores. Alongside the bounding boxes, YOLO predicts class probabilities for each detected object within each grid cell. These are the classes defined in the process of preparing data for learning. It assigns a probability to each predefined class label based on the objects present in the cell. YOLO assigns a confidence score to each predicted bounding box, reflecting model's confidence in the box with an object. A threshold (a constant used to filter out detections with confidence scores below a certain threshold value) is applied to filter out low-confidence detections. To eliminate redundant and overlapping detections, YOLO applies non-maximum suppression. It selects the most confident detection in each grid cell and removes duplicate detections based on their overlap using intersection-over-union (IoU) calculations. After non-maximum suppression, YOLO outputs the final set of bounding boxes and their associated class labels, providing a comprehensive detection result for the input image [Lecun et al., 1998; Lin et al., 2014; Ren et al., 2017; Titano et al., 2018].

YOLO is known for its impressive speed, enabling real-time object detection on various platforms, including embedded systems and drones. It achieves this by optimizing the network architecture and using techniques like anchor boxes. YOLO has evolved over time into different versions such as YOLOv1, YOLOv2 (also known as YOLO9000), YOLOv3, YOLOv4 up to YOLOv8. Each version introduces improvements in terms of accuracy, speed, and architectural enhancements [Redmon et al., 2016; Redmon and Farhadi, 2018].

YOLO is trained using labeled datasets where bounding box coordinates and class labels are provided for each object. The training process involves optimizing the network's parameters using techniques like backpropagation and gradient descent to minimize detection loss. YOLO has a wide range of applications, including object detection in images and videos, surveillance systems, autonomous vehicles, robotics, and more. Its real-time capabilities make it suitable for scenarios that require fast and accurate object detection. It has gained popularity partly due to its open-source implementation, allowing researchers and developers to access and modify the code. This has led to further advancements and adaptations of the YOLO algorithm. This allowed the authors to adapt (re-train) the model to recognize completely new objects such as *Arcella vulgaris* [Liu et al., 2016; Redmon et al., 2016; Redmon and Farhadi, 2018].

YOLOv4 is a significant advancement in the YOLO series of object detection models [Bochkovskiy et al., 2020]and theoretical justification of the result, is required. Some features operate on certain models exclusively and for certain problems exclusively, or only for small-scale datasets; while some features, such as batch-normalization and residual-connections, are applicable to the majority of models, tasks, and datasets. We assume that such universal features include Weighted-Residual-Connections (WRC. It introduces various improvements over previous versions, including YOLOv3. YOLOv4 incorporates a modified CSPDarknet53 backbone architecture which enhances feature extraction capabilities. It also introduces the PANet (Path Aggregation Network) module for multi-scale feature fusion, enabling a better detection of objects at different scales. YOLOv4 utilizes training techniques like mosaic data augmentation, CIOU (Complete Intersection over Union) loss function, and a cosine annealing scheduler to improve model training and detection performance. With its focus on accuracy, speed, and efficiency, YOLOv4 has achieved state-of-the-art performance in object detection tasks. Another evolution of YOLOv3 was YOLOv5 which was the first model in "YOLO family" to not be released with an accompanying paper. Glenn Jocher, the founder of Ultralytics, had been maintaining a version of YOLOv3 implemented in PyTorch, but as he continued to make improvements in the architecture itself, he ultimately decided to release a new repo branded as YOLOv5 [Liu et al., 2016; Redmon, 2023].

YOLOv8, developed by Ultralytics, is a model that builds upon the influential YOLOv5 and introduces various architectural and developer experience improvements. It follows an anchor-free approach so directly predicts the center of an object instead of the offset from a predefined anchor box. YOLOv8 offers developer-convenience features such as an easy-to-use command-line interface (CLI) and a well-structured Python package. These enhancements aim to provide a more user-friendly and efficient experience for developers utilizing the YOLOv8 model.

In the process of preparing material for training, the photos with the YOLO models were labeled in accordance with the format acceptable by the network [Dutta and Zisserman, 2019]. In the pictures with *Arcella*, rectangular frames were drawn manually to surround all of the occurring organisms. The frames were created in such a way that the inner edge gently adhered to the edge of the *Arcella vulgaris* wall. All visible organisms were labeled, alive or dead, or even an empty or broken shell. The images data set consisted of 990 images that were divided into train (70%), validation (20%) and test data set (10%). Accuracy, precision and recall were calculated for the model evaluation. The formulas for the metrics are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where: *TP* – the true positive cases, *TN* – the true negative ones, *FP* – the false positive cases, FN – the false negative ones.

The model training and the prediction were performed with the Python programming language, version 3.9.13. The package ultralytics was used.

RESULTS

Figures 5, 6 present the selected YOLO network predictions. The model detects *Arcella vulgaris* on the image, predicts the bounding boxes and labels them with certain probability (numbers attached to bounding boxes).

Table 1 compares the YOLO models performance in the *Arcella vulgaris* detection problem. One can observe that YOLOv8 delivered a significant performance improvement. YOLOv4, trained for 1500 epochs, achieved worse results that YOLOv8 trained on 100 epochs. The precision metric for YOLOv4 equals 0.9412, while for the newer YOLO version it is 0.9515. The same can be said about recall – 0.9412 for YOLOv4, 0.9515 for YOLOv8. It is worth noting that while accuracy is almost at the same level and recall still improved – from 0.8889 to 0.9074. In addition, in both cases precision and recall measures exceed the value of 0.9, which suggests a very good quality of network classification.

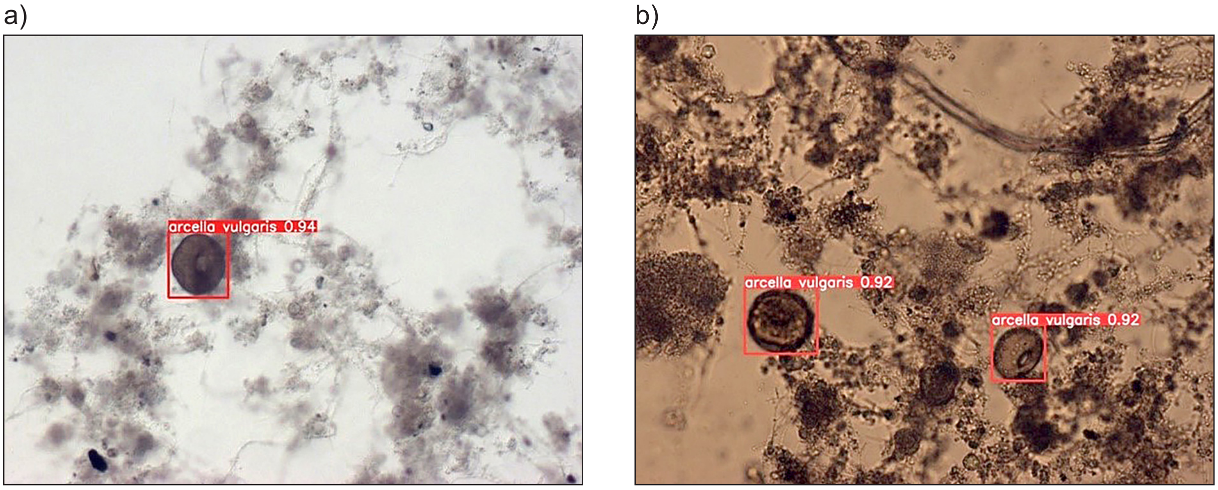

Figure 5. (a) One *Arcella vulgaris* detected, (b) two *Arcella vulgaris* detected

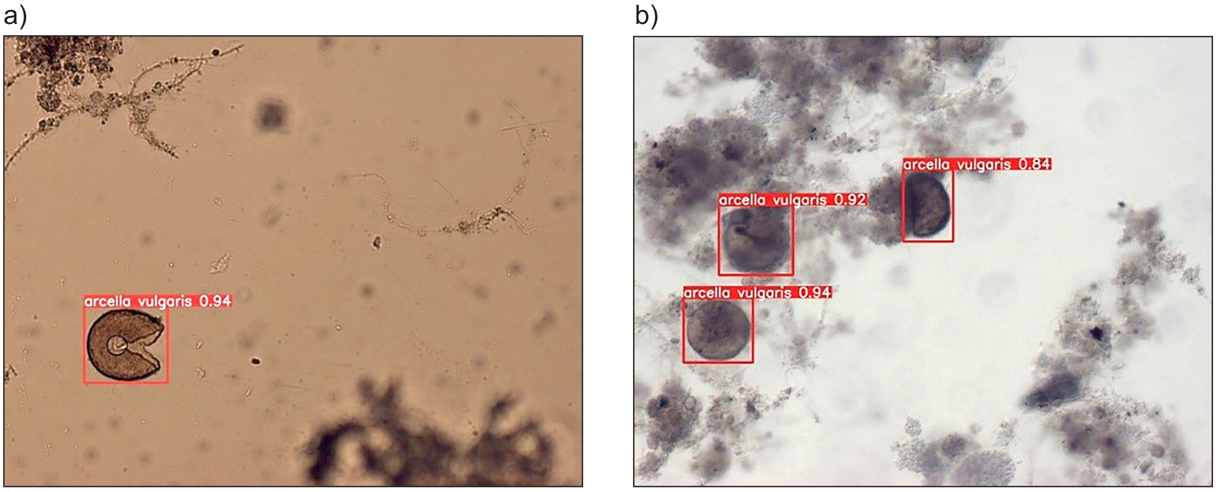

Figure 6. (a) One *Arcella vulgaris* detected, (b) three *Arcella vulgaris* detected

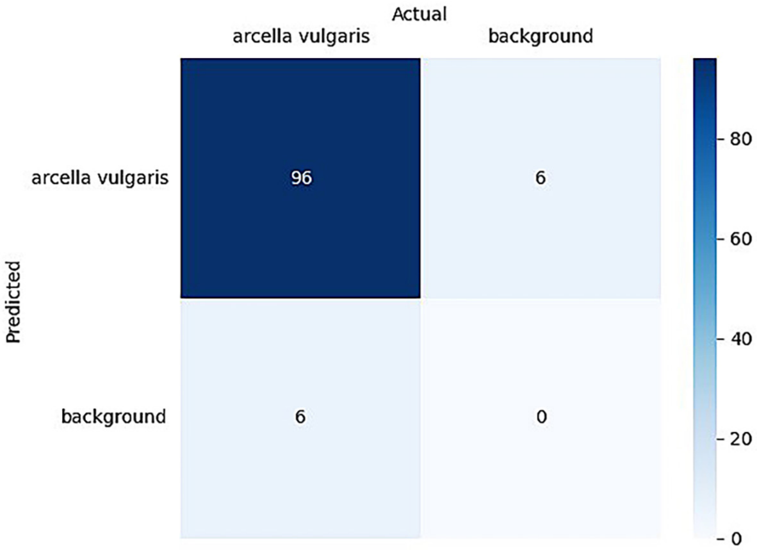

Figure 7. Confusion matrix for the YOLOv4 algorithm

Table 1. Comparison of object detection quality measures on the test set

Model	Epochs	Training time [h]	Precision	Recall	Accuracy
YOLOv4	1500	20.85	0.9412	0.9412	0.8889
YOLOv8	100	10.40	0.9515	0.9515	0.9074

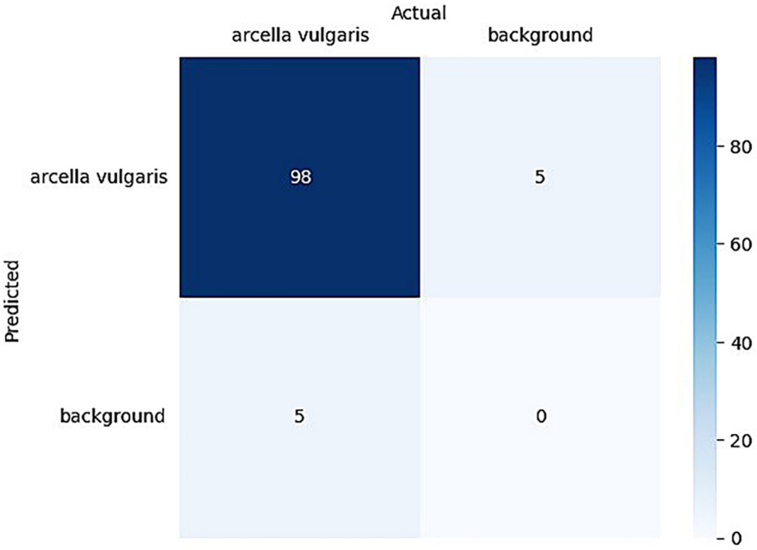

Figure 8. Confusion matrix for the YOLOv8 algorithm

All labels generated by the network show a high level of certainty, in most cases above 90%. The level of certainty is described by confidence score we refer to in the text. Incorrect detection was made by the YOLOv4 artificial network for 12 organisms, i.e. 6 of them marked *Arcella vulgaris* were not present in the photo and the remaining 6 were missing (Figure 7). Incorrect detection in the YOLOv8 prediction was done for 10 organisms, namely 5 of them labeled *Arcella vulgaris* were not present in the photo and the other 5 were missing (Figure 8).

CONCLUSIONS

The main purpose of the work was to present results of microscopic studies of activated sludge supported by automatic image analysis together with performance comparison of the YOLOv4 and the YOLOv8 deep learning networks based on task of identifying and classifying a selected group of testate amoebae. Digital images of the samples of the biological material with *Arcella vulgaris* collected at the Hajdów municipal wastewater treatment plant in Lublin were analyzed. The architecture of the networks built for the detection task was general, which means that the structure of the layers and filters was not affected in any way by the purpose of using the models.

Therefore, given the universal construction of the models, the results of the accuracy and quality of the classification can be considered as very good. This means that the general architecture of the YOLO networks can also be used for specific tasks such as identification of shell amoebas. However, YOLOv8 delivered a significant performance improvement – with fewer epochs, it provided better object detection quality than the older version of YOLO, namely YOLOv4.

The model identifying the species *Arcella vulgaris* in images of polluted water can be extended to classify more species of eukaryotic organisms. The model improved in this way could be used in the future in automatic studies into the condition of activated sludge and attempts to determine the rate of changes in the population of organisms. It could also be used in an automatic assessment of the stability of a wastewater treatment plant based on the number of analyzed individuals in each stage of wastewater treatment and in treated wastewater.

REFERENCES

1. Amaral A.L., Mesquita D.P., Ferreira E.C. 2013. Automatic identification of activated sludge disturbances and assessment of operational parameters. Chemosphere, 91(5), 705–710.
2. Babko R., Kuzmina T., Łagód G., Jaromin-Gleń K. 2014. Changes in the structure of activated sludge protozoa community at the different oxygen condition. Chemistry-Didactics-Ecology-Metrology, 19(1–2), 87–95. (in Polish)
3. Babko R., Łagód G., Kuzmina T., Danko Y. 2023. Arcella vulgaris (testacea) jako obiekt testowy osadu czynnego [online]. Available from: https://pub.pollub.pl/publication/15051/ [Accessed 22 Dec 2023]. (in Polish)
4. Bay H., Ess A., Tuytelaars T., Van Gool L. 2008. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, 110(3), 346–359.
5. Bochkovskiy A., Wang C.Y., Liao H.-Y.M. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection.
6. Dutta A., Zisserman A. 2019. The VIA Annotation Software for Images, Audio and Video. In: Proceedings of the 27th ACM International Conference on Multimedia, Nice, France: ACM, 2276–2279.
7. Fiałkowska E., Fyda J., Pajdak-Stós A., Wiąckowski K. 2005. Osad czynny: biologia i analiza mikroskopowa. Kraków: Oficyna Wydawnicza ‘Impuls’. (in Polish)
8. Girshick R. 2015. Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile: IEEE, 1440–1448.
9. Girshick R., Donahue J., Darrell T., Malik J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA: IEEE, 580–587.
10. He K., Zhang X., Ren S., Sun J. 2015. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904–1916.
11. Jaromin-Gleń K., Babko R., Łagód G., Sobczuk H. 2013. Community composition and abundance of protozoa under different concentration of nitrogen compounds at “Hajdow” wastewater treatment plant. Ecological Chemistry and Engineering S, 20(1), 127–139. (in Polish)
12. Krizhevsky A., Sutskever I., Hinton G.E. 2017. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90.
13. Lecun Y., Bottou L., Bengio Y., Haffner P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.
14. Lin T.Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick C.L. 2014. Microsoft COCO: Common Objects in Context. In: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, eds. Computer Vision – ECCV 2014. Cham: Springer International Publishing, 740–755.
15. Liu S., Qi L., Qin H., Shi J., Jia J. 2018. Path Aggregation Network for Instance Segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT: IEEE, 8759–8768.
16. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., Berg A.C. 2016. SSD: Single Shot MultiBox Detector, 21–37.
17. Lowe D.G. 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.
18. Ogden G.G., Hedley R.H. 1980. An atlas of freshwater testate amoebae. Soil Science, 130(3), 176.
19. Papert S. 2004. The Summer Vision Project.
20. Pérez-Uz B., Arregui L., Calvo P., Salvadó H., Fernández N., Rodríguez E., Zornoza A., Serrano S. 2010. Assessment of plausible bioindicators for plant performance in advanced wastewater treatment systems. Water Research, 44(17), 5059–5069.
21. Redmon J. 2023. Darknet: Open Source Neural Networks in C [online]. Available from: https://pjreddie.com/darknet/ [Accessed 22 Dec 2023].
22. Redmon J., Divvala S., Girshick R., Farhadi A. 2016. You Only Look Once: Unified, Real-Time Object Detection.
23. Redmon J., Farhadi A. 2018. YOLOv3: An Incremental Improvement.
24. 24. Ren S., He K., Girshick R., Sun J. 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149.
25. Song B., Wang Y., Lou L.P. 2023. SSD-based carton packaging quality defect detection system for the logistics supply chain. Ecological Chemistry and Engineering S, 30(1).
26. Stawarczyk M., Stawarczyk K. 2015. Use of the ImageJ program to assess the damage of plants by snails. Chemistry-Didactics-Ecology-Metrology, 20(1-2).
27. Titano J.J., Badgeley M., Schefflein J., Pain M., Su A., Cai M., Swinburne N., Zech J., Kim J., Bederson J., Mocco J., Drayer B., Lehar J., Cho S., Costa A., Oermann E.K. 2018. Automated deep-neural-network surveillance of cranial images for acute neurologic events. Nature Medicine, 24(9), 1337–1341.

28. Viola P., Jones M. 2001. Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA: IEEE Comput. Soc, I-511-I–518.
29. Wang C.Y., Mark Liao H.Y., Wu Y.H., Chen P.-Y., Hsieh J.-W., Yeh I.-H. 2020. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA: IEEE, 1571–1580.
30. Wang L.Y., He Y.P. 2023. Environmental landscape art design based on visual neural network model in rural construction. Ecological Chemistry and Engineering S, 30(2).