Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/41/20230714

Improved small-object detection using YOLOv8: A comparative study

Huadong Huang[1], Binyu Wang[2], Jiannan Xiao[3], and Tianyu Zhu[4,5,6]

[1]Beijing University of Posts and Telecommunications No.10 Xitucheng Road, Haidian District, Beijing, China
[2]DGUT-CNAM Institute, Dongguan University of Technology, Dongguan, China
[3]University of Science and Technology of China, No.96, JinZhai Road, Baohe District, Hefei, Anhui, China
[4]ShanghaiTech University,393 Middle Huaxia Road, Pudong, Shanghai, China

[5]zhuty1@shanghaitech.edu.cn
[6]Corresponding author

Abstract. In the last decade or so, deep neural networks have evolved at a rapid pace, where computer vision has been constantly refreshing its best performance and has been integrated into our lives. In the field of target detection, YOLO model is a popular real-time target detection algorithm model that is fast, efficient, and accurate. This research aims to optimize the latest YOLOv8 model to improve its detection of small objects and compare it with another different version of YOLO models. To achieve this goal, we used the classical deep learning algorithm YOLOv8 as a benchmark and made several improvements and optimizations. We optimized the definition of the detection head, narrowed its perceptual field, and increased its number, allowing the model to better focus on the detailed information of small objects. We compared the optimized YOLOv8 model with other classical YOLO models, including YOLOv3 and YOLOv5n. The experimental results show that our optimized model improves small object detection with higher accuracy. This research provides an effective solution for small object detection with good application prospects. With the continuous development and improvement of the technology, we believe that the YOLO algorithm will continue to play an essential role in object detection and provide a reliable solution for various real-time applications.

Keywords: object detection, small object, YOLO, detection head.

1. Introduction

Object detection is an essential task in the computer vision field, which is widely used in real-time video analysis, automatic driving, and intelligent security. Before 2014, traditional target detection algorithms required extracting features manually, which was time-consuming and unstable. The SOTA algorithm DPM [1] detector at that time, although inference speed was faster than others and could adapt to slight deformation, could not adjust to large-scale rotation and showed low robust ability.

In recent years, due to the emergence of convolution neural networks (CNN), deep learning-based object detection algorithms have made significant progress, and two branches of anchor-free methods and anchor-based methods have been developed [2]. Anchor-based methods consist of a one-phase

DOI: 10.54254/2755-2721/41/20230714

algorithm and a two-phase algorithm. The two-phase algorithm consists of generating a region proposal generated from the image and producing a bounding box from the region proposal. The representatives of this type of algorithm RCNN have high Mean Average Precision (mAP), but the inference speed is slow because it should train several networks to complete different jobs in different inference stages. This makes RCNN cannot meet the real-time requirement [3]. Although many improvements have been made to accelerate the inference speed of RCNN, such as Fast-RCNN [4], Faster-RCNN [5], and Mask-RCNN [6], they only change the structure of these networks, so the frame rate is still low. You Only Look Once (YOLO) [7] algorithm as a one-stage algorithm has attracted much attention. It uses only one network to predict the bounding box coordinate and the classification probability, so it is extremely fast. The original YOLO model has several drawbacks, such as the mAP is slightly low and cannot detect a large number of grouped objects. Then, researchers developed better versions of YOLO to improve its performance, and recently, YOLOv8 has come out. However, despite YOLOv8's impressive achievements in real-time and accuracy, there is still room for improvement.

YOLOv8 has some difficulties in dealing with small and dense targets and is prone to the problems of missed detection and overlapped detection, especially when the size of the object is smaller than 8*8. YOLOv8 uses a predefined detection head, which is insufficient to detect the details of small targets, while it is easy to produce overlapping detection frames for dense targets.

To solve the problem, this paper optimizes the definition of detection head, shrinking its perception field and increasing the number. After the reconstruction, YOLOv8 showed better performance on grouped small objects detection. Firstly, the datasets we used in inference has more than 30 objects on average. Normally it's a significant time cost to predict every bounding box parameter, but our model can figure it out at a fantastic speed and the average fps is 30. Secondly, while the original YOLOv8 performs a recall rate of less than 60%, the model can find almost all the object, and the recall rate is higher than 80%. Lastly, the model of this paper provides an example of a counting machine, which is designed to 'count the sand in a desert', and it can be used to do similar jobs which is time-consuming and easy to be done wrong.

2. Method

2.1. Models Architecture

The backbone and head of a convolutional neural network are the two fundamental components of the YOLOv8 architecture, which is an improvement over earlier iterations of the YOLO algorithm [8]. A revised Currently, CS architecture that consists of thirty-five convolutional layers and uses cross-stage fractional connections to enhance the transfer of data between layers serves as the foundation of YOLOv8. The bounding boxes, item evaluations, and probabilities of classes of recognized objects are anticipated by the YOLOv8 head, which is made up of a number of convolutional layers followed by fully connected layers. A noteworthy aspect of YOLOv8 is the inclusion of an apparatus for self-attention [9] in the network's head. This feature enables the model to concentrate on various areas of the imagery and change the value of elements according to relevance. Another noteworthy aspect of YOLOv8 is its capacity to recognize objects on many scales, which is accomplished via a characteristic hierarchy network [10]. The model can reliably recognize things of various sizes inside an image because to the network's numerous layers that detect objects at various scales. Figure 1 displays a typical YOLOv8 model structure.

DOI: 10.54254/2755-2721/41/20230714

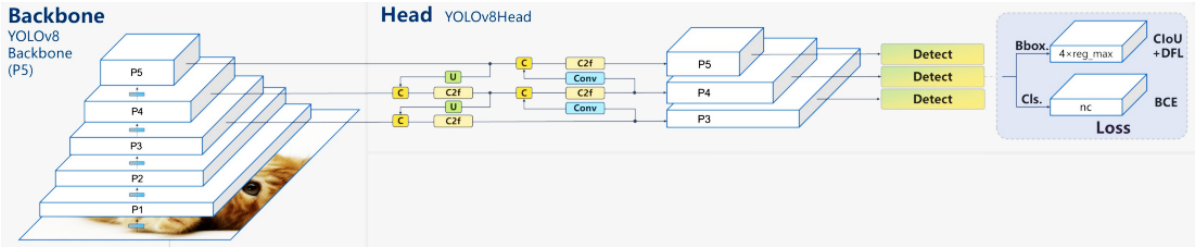

Figure 1. Structure of YOLOv8.

2.2. Head

To address the limitations of traditional methods, regression-based approaches have emerged as a research focus in the field of three-dimensional human pose estimation. These methods leverage deep learning techniques to learn the mapping relationship from images to pose and shape, enabling direct regression of human pose and shape. This approach achieves more accurate estimation results through extensive training data and deep neural networks.

In YOLOv8, the "head" part refers to the top-level hierarchical structure of the neural network model, which is responsible for processing the feature map after feature extraction from the basic level. Specifically, the "head" part of YOLOv8 mainly includes three key components: detection layers, up sample layers, and route layers. The detection layers are responsible for converting input feature maps into detection bounding boxes. Usually, the detection layer in YOLOv8 converts feature maps into bounding boxes of different scales and corresponding category prediction probabilities through convolution operations. Each detection layer is associated with an anchor box for detecting objects at different scales. Up sample layers are used to increase the resolution of the feature map. These layers typically use deconvolution operations to achieve up-sampling and convert low-resolution feature maps to high-resolution ones. The up-sampling layer is mainly used to increase the model's perception of small-sized objects. The route layer is used to connect feature maps of different levels. It can connect the previous layer's feature map with the earlier layer's feature map to obtain feature maps with different scale feature information. This multi-scale feature fusion helps the model to detect objects of different sizes and types. In summary, the "head" part in YOLOv8 is a key network hierarchy, which converts feature maps into detection bounding boxes through the combination of the detection layer, up sample layer and route layer, and provides multi-scale feature fusion ability to achieve efficient detection of targets of different sizes and types.

2.3. Deficiency and optimizer

In standard object detection tasks, the problem of missing detection or poor detection effect often occurs when there are small objects in the data set. The reason is stated as follows: The YOLOv8 model has 3 detection heads by default, which can perform multi-scale detection of targets. Among them, P3/8 corresponds to a detection feature map size of 80*80, which is utilized for recognizing items larger than 8*8; P4/16 relates to a detection feature map size of 40*40, which is applied to recognize items larger than 16*16; and P5/32 corresponds to a recognition characteristics map size of 20*20, which is used to identify items larger than 32*32, as illustrated below:

DOI: 10.54254/2755-2721/41/20230714

Updated head: detecting small objects	
1	[-1, 1, nn.Unsample, [None, 2, 'nearest']]
2	[[-1, 6], 1, Concat, [1]] # cat backbone P4
3	[-1, 3, C2f, [512]] # 12
4	
5	[-1, 1, nn.Upsample, [None, 2, 'nearest']]
6	[[-1, 4], 1, Concat, [1]] # cat backbone P3
7	[-1, 3, C2f, [256]] # 15 (P3/8-small)
8	
9	[-1, 1, Conv, [256, 3, 2]]
10	[[-1, 12]. 1, Concat, [1]] # cat head P4
11	[-1, 3, C2f, [512]] # 18 (P4/16-medium)
12	
13	[-1, 1, Conv, [512, 3, 2]]
14	[[-1, 9], 1, Concat, [1]] # cat head P5
15	[-1, 3, C2f, [1024]] # 21 (P5/32-large)
16	[[15, 18, 21], 1, Detect, [nc]] # Detect(P3, P4, P5)

Then it emerges instinctively that there may be a problem of poor capability for detecting tiny objects whose sizes are smaller than a particular scale or one of the dimensions (width and height) is not large enough. This study introduces a tiny object detection layer (160*160 detection feature map for identifying targets above 4*4, for example) to enhance the detection performance of small targets.

To achieve this improvement, we maintain the original results in the Backbone part but adjust the model structure of the head part, see below:

Optimizer： New detection head.	
1	[-1, 1, nn.Upsample, [None, 2, 'nearest']]
2	[[-1, 2], 1, Concat, [1]] # cat backbone P3
3	[-1, 3, C2f, [128]] # 18 (P2/4-xsmall)

2.4. Dataset

Generally speaking, we totally apply two different datasets to test the performance of our network. The first is the SOD (Small Object Detection) dataset [11], a collection of images specifically curated and annotated for small object detection tasks. Small object detection aims to identify and highlight an image's most visually distinctive objects or regions. The images in such dataset are scaled to 640*640 and the average size of objects is about 25*25. We train this dataset with multiple models with epochs = 30, image size = 640, and batch size = 3. The second one is the bacterial colony dataset [12], which is a collection of images specifically focused on bacterial colonies grown in a laboratory setting. It is commonly used in microbiology and bioinformatics research to study bacterial growth patterns, analyse colony characteristics, and develop automated colony recognition and classification algorithms. The image in such a dataset is scaled to 1280*1295. Various large-size bacteria (about 10*10) and tiny-size bacteria (about 2*2) constitute each image. We train this dataset with multiple models with epochs = 15, image size = 640, and batch size = 10.

In order to evaluate the performance of the optimized YOLOv8n network, we conducted validation on various datasets using YOLOv3 and YOLOv5 as well. This allowed us to analyse the validation speed and accuracy differences among the three models. From both structural and parametric perspectives, we compared YOLOv8 with YOLOv3 and YOLOv5 to determine the strengths and weaknesses of our modified YOLOv8n network. This comprehensive comparison enables us to assess the practical value of our optimized model. It is important to recognize that our modified YOLOv8n network may only outperform other networks in specific situations. The modification focused on adding small object detection layers to extract shallower features, which may result in inferior performance in

DOI: 10.54254/2755-2721/41/20230714

ordinary cases. Therefore, the comparison aims to explore further the niche where our model excels and its future development and potential.

3. Result and discussion

3.1. Performance

The recall rate, precision, and mAP are three essential criteria in object detection. Therefore, this paper emphasizes the importance of comparing these metrics to evaluate the performance of object detection models. The recall rate refers to the fraction of actual objects in an image that the model correctly detected. Precision relates to the fraction of detected objects correctly identified and not falsely detected. mAP (mean average precision at 50% Intersection Over Union) is a way to summarize precision and recall over multiple classes in object detection tasks, providing a holistic view of a model's performance. The P-R curve of our model is in Figure 4.

3.1.1. Comparison with YOLOv8n network

The YOLOv8n network underwent a series of optimizations, and the subsequent results have been encouraging. Upon comparing the optimized network with the original YOLOv8n model, its performance metrics showed a clear enhancement. Visual comparison with YOLOv8 is in Figure 5. A comparison of widely used metrics is shown in Figure 6 and Table 1. Specifically, when the improved model was trained on the SOD dataset, a marked improvement in both prediction accuracy and training velocity was observed, most notably during the initial 5 epochs. The optimized network's final precision is 92.4%, a 4% improvement over the original network. The best recall rate increased by 4% to 73.4% after optimizing the YOLOv8n network. After adding the detection head, the mAP50 increased from 74.2% to 78.4%. This means that the optimized network can predict the object's location more precisely.

It is worth emphasizing that these enhancements were not just confined to the accuracy metrics, they also appeared in the training speed. In the initial five epochs, the mAP50 of the optimized network is about 10% higher than the original network. Concurrently, precision and recall rates surged quicker in the optimized network, further accentuating its advantages over the original version. Due to the computational power limitations and the number of training epochs, the enhancements to the network are not significant. The authentic images of the SOD dataset are 1280*1295 pixels. However, to reduce the amount of calculation, the images were compressed to 640*640 pixels. Therefore, the improvement was not noticeable. Besides, the training loss and the validation loss didn't show any improvement. The optimized network seems to have some deficiencies when it is applied to different datasets. The optimized network does not show improvement when trained on the bacterial colony dataset. This needs further exploration because of the small quantity of datasets and training time.

DOI: 10.54254/2755-2721/41/20230714

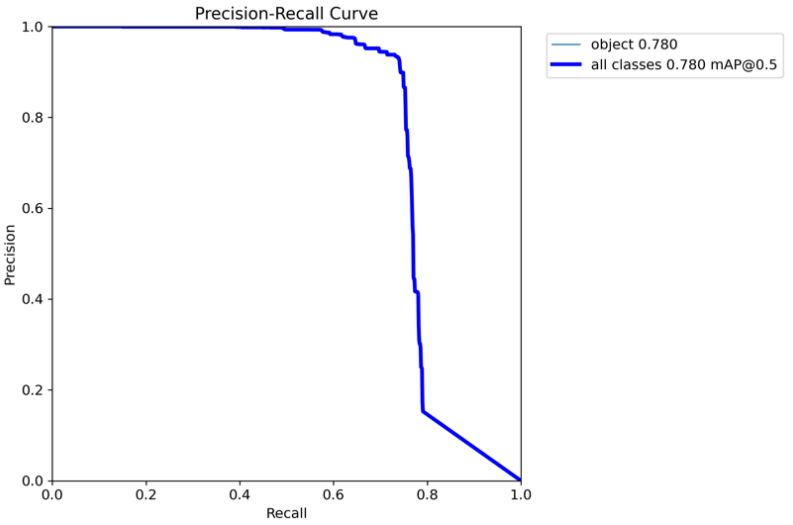

Figure 4. Precision-Recall curve.

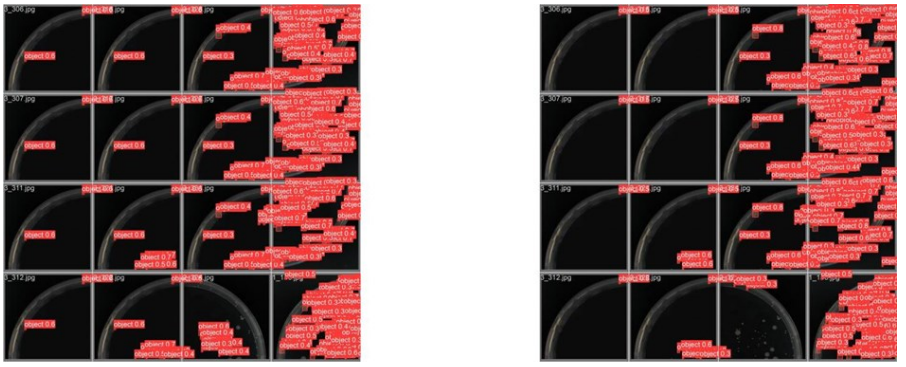

Figure 5. Detection results on bacteria colonies. (a) by optimized YOLOv8. (b) original YOLOv8

3.1.2. Comparison with other detection networks

Compared with the former version of the YOLO network, there are more significant improvements. After training 15 epochs on SOD dataset, the recall rate of the YOLOv5n is 65.5% and the optimized YOLOv8n achieved the rate of 72.4%, which is more obvious. The improvements in precision and mAP are also higher than the previous comparison. This is partly due to the YOLOv8n's performance itself. The original YOLOv8n network's training speed is almost the same as that of YOLOv5n and YOLOv3. Therefore, it proves that the training speed of the network had indeed increased, in the early training state. However, when it comes to the network loss, the optimized YOLOv8n network had even higher losses than the former version of the YOLO model. This indicates that the robustness of the optimized network hasn't improved after the optimization.

DOI: 10.54254/2755-2721/41/20230714

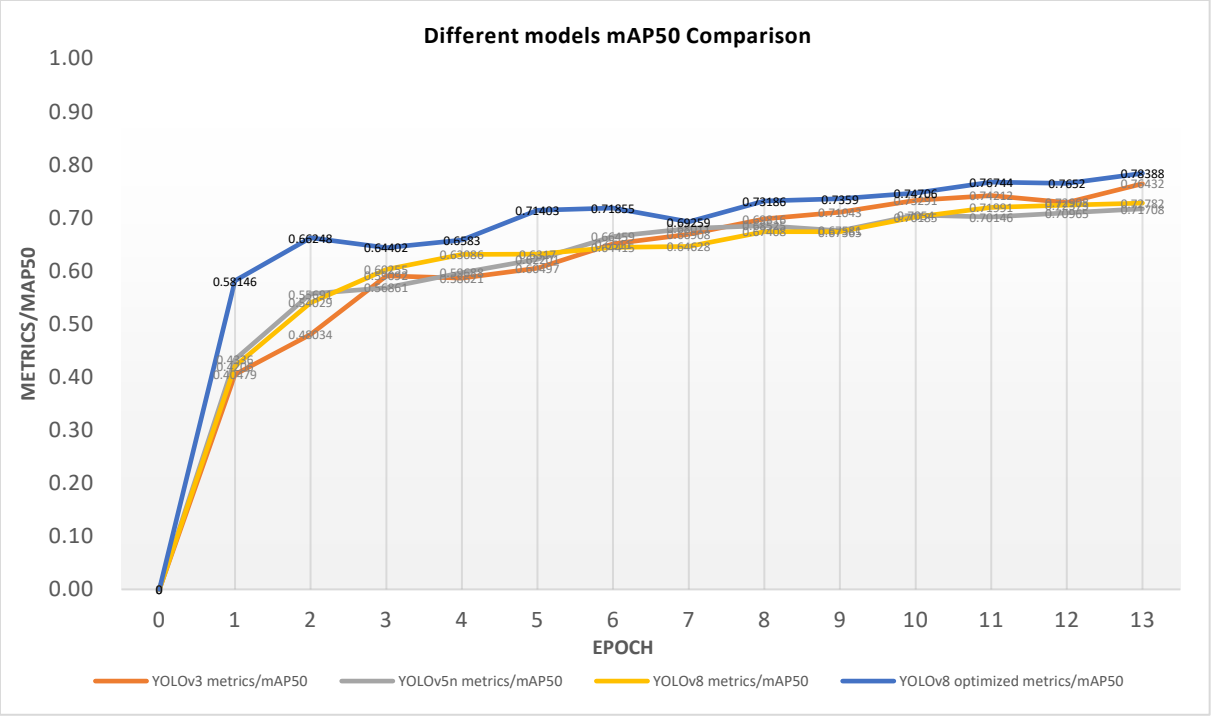

Figure 6. Different YOLO models' mAP metric curves.

Table 1. Other metrics of YOLO models.

Model	train/box_loss	validate/box_loss	metrics/recall	metrics/precision
YOLOv3	1.6007	1.5243	0.72661	0.91641
YOLOv5n	1.7389	1.6979	0.65559	0.88717
YOLOv8 original	1.6547	1.5848	0.69713	0.88717
YOLOv8 optimized	1.7455	1.7082	0.72483	0.92441

3.2. Application

Optimized YOLO that can detect small and dense objects will bring great benefits once widely applied practically. One of the most promising application fields is autonomous vehicles. Due to its small volume and fast speed, it is suitable for autonomous vehicles' real-time objection detection. The road signs and the pedestrians may sometimes be too small to detect. To ensure the safety of human beings, it is necessary for it to detect the road condition efficiently which contains many small objects. This model can be used in medical diagnosis. The tiny tumors, lesions, or other abnormal structures are hard to recognize by the naked eye or traditional object detection models. Early diagnosis leads to higher cure rates for these diseases. However, the medical images depicting these diseases are often small in scale. Therefore, the detection model presented in this paper can help increase the cure rate. The optimized model can also greatly aid in detecting objects in satellite aerial photos. Satellite aerial photos are widely used for weather prediction, forest fire monitoring, and other disaster predictions. Objects in satellite aerial images are relatively small for the original YOLO model to detect. However, the optimized network is better equipped to process these images.

DOI: 10.54254/2755-2721/41/20230714

3.3. Expectation

The YOLOv8n network can be further improved. Limited by the training epochs and the dataset, the enhancement was not good enough. Better datasets, for example, images with more objects under 4*4 pixels, can boost the optimization effect. Expanding the number of training epochs can also enhance the model's performance. Adding too many detection heads may cause some side effects. It may produce too many bounding boxes. Further exploration could focus on modifying the connections between the heads and the network's backbone. Changing the structure of the feature-transforming layers is also an effective way to optimize the network.

4. Conclusion

To improve the performance of YOLOv8, this paper adds a detection head to the head of the model while keeping the structure of the backbone. As a result, the modified model can find small objects as small as 4*4 pixels. Compared to the original YOLOv8 model, our model shows a 4.2% higher precision rate and 4.0% higher recall rate in the task of detecting bacterial colonies and a rise of 9.2% with regard to mAP. In fact, the model can detect almost every colony visually, which means the model achieves our primary goal, namely counting anything. The experiment proved that adding a specified detection head can improve the ability of YOLOv8 to detect small objects. The model can be used in multiple fields like calculating current traffic flow with satellite cameras, tracing the growth of bacterial colonies, etc. However, if this work adds too many detection heads, it's likely to slow down the training and inference process, which is an underlying drawback of our method. This paper suggests that deleting some detection head and making YOLO a specified model for particular tasks may be a good choice, or this paper can a simple layer before the input layer, which receives prompt words and modifies the detection head automatically. In another aspect, this model doesn't consider the cover of objects, which is extremely difficult with small things and would be a significant step toward the final goal of counting sands. This paper leaves it for future research.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Felzenszwalb, P.F.,, Girshick, R.B.,, McAllester, D., and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), pp.1627–1645.

[2] Jiao, L.,, Zhang, F.,, Liu, F.,, Yang, S.,, Li, L.,, Feng, Z., and Qu, R., 2019. A survey of Deep Learning-based object detection. IEEE Access, 7, pp.128837–128868. Redmon, J. et al. (2016) 'You only look once: Unified, real-time object detection', 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Preprint]. doi:10.1109/cvpr.2016.91.

[3] Girshick, R.,, Donahue, J.,, Darrell, T., and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition.

[4] Girshick, R., 2015. Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV).

[5] Ren, S.,, He, K.,, Girshick, R., and 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), pp.1137–1149.

[6] He, K.,, Gkioxari, G.,, Dollár, P., and Girshick, R., n.d. Mask R-CNN. Computer Vision and Pattern Recognition.

[7] Redmon, J.,, Divvala, S.,, Girshick, R., and Farhadi, A., 2016. You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

DOI: 10.54254/2755-2721/41/20230714

[8] Gongguo, Z., and Junhao, W., 2021. An improved small target detection method based on Yolo V3. 2021 International Conference on Electronics, Circuits and Information Engineering (ECIE).

[9] Lin, Z., Feng, M., Santos, C.N.D., Yu, M., Xiang, B., Zhou, B. and Bengio, Y., 2017. A structured self-attentive sentence embedding. 2017 International Conference on Learning Representations

[10] Lin, T.-Y.,, Dollar, P.,, Girshick, R.,, He, K.,, Hariharan, B., and Belongie, S., 2017. Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[11] Ma, Z.,, Lei Yu, and Chan, A.B., 2015. Small instance detection by integer programming on object density maps. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[12] Zhang, B.,, Zhou, Z.,, Cao, W.,, Qi, X.,, Xu, C., and Wen, W., 2022. A new few-shot learning method of bacterial colony counting based on The edge computing device. Biology, 11(2), p.156.