Real-Time Detection Algorithm of Marine Organisms Based on Improved YOLOv4-Tiny

YANLI SHI1, ZIRAN GAO2, AND SHA LI3

1School of Science, Jilin Institute of Chemical Technology, Jilin City 132022, China
2College of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin City 132022, China
3College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao City 266061, China

Corresponding author: Yanli Shi (syl@jlict.edu.cn)

This work was supported in part by the Natural Science Foundation of Jilin Province under Grant YDZJ2022201ZYTS602.

ABSTRACT

Marine organisms detection based on machine vision requires high real-time performance and accuracy. Network feature extraction is frequently made more difficult when underwater robots collect information on the seafloor because of the uneven distribution of light on the seafloor, the significant impact of water waves, and the complexity of the seafloor environment. How to detect marine organisms quickly and accurately is a great difficulty and challenge. To address the above problems, we propose a MODA (Marine Organism Detection Algorithm) based on an improved YOLOv4-tiny marine organism detection algorithm. Firstly, the Coordinate Attention module, an ultra-lightweight attention mechanism, is constructed and embedded into the backbone network to retain more information about the target of interest and enhance the feature extraction capability of the network. Secondly, the Hybrid Dilated Convolution (HDC) structure is constructed and added to the improved network to expand the feature map perception field and obtain richer semantic information to improve the network detection accuracy. Finally, a better MODA model is proposed based on the two methods mentioned above. The experimental results show that the improved model MODA improves the mAP metric from 74% to 76.62% on the URPC dataset and only increases the computational effort by 0.06 GM compared with the original YOLOv4-tiny model; the mAP metric improves from 92.37% to 98.41% on the Aquarium dataset. This improvement indicates that the MODA model is more suitable for marine organisms detection tasks.

INDEX TERMS
Deep learning, marine organisms, target detection, YOLOv4-tiny.

I. INTRODUCTION

The oceans cover about 70% of the Earth’s surface area and contain a variety of resources that can have a significant impact on human life. However, it is difficult to obtain data on zooplankton and benthic organisms [1], which regulate changes in the marine ecosystem, and we do not have a corresponding understanding of their species and population changes. To increase the mastery of marine information and promote the sustainable development of the ocean, the use of underwater robots to identify and locate marine organisms for detection is the most fundamental step [2]. The current increasing performance of computer hardware and software devices also continues to drive the development of target detection technology [3].

Early target detection was carried out by manual extraction of target features such that the time cost was high, the accuracy and speed were relatively low, and the practicability was low. Due to the complicated natural environment, underwater detection is typically more challenging, so it is crucial to gather target information accurately and rapidly. Deep learning target detection algorithms are gradually becoming popular research, and convolutional neural network (CNN)-based models use a backpropagation mechanism for learning, excellent feature extraction ability, and multilayer convolutional learning, which makes the models generalize well [4]. Deep learning target detection algorithms are mainly classified into two categories, one-stage, and two-stage, based on the presence or absence of candidate frame generation.
The two-stage detection algorithms are mainly based on the regional convolutional neural network RCNN [5], Faster-RCNN [6], and Faster-RCNN [7], series, which first extract target candidate frames from images and then classify and predict their positions. For example, in order to recognize and identify fish species, Ajagbe et al. [8] evaluated the efficiency of eight DL models in bioinspired object detection (BOD) using six metrics, and the results proved that the CNN model is the best. Li et al. [9] improved Fast-RCNN and deep ConvNets, which surpassed RCNN in terms of accuracy and target detection time. Han et al. [10] combined the maximum RGB method with the grayscale scale method to develop an improved underwater vision. Kumar et al. [11] designed and implemented a hybrid deep and machine learning model and evaluated the model using a new dataset, and the transformation of the Convolution Neural Network (CNN) classifier has good performance. Song et al. [12] propose a new underwater biometrics technique that combines the Mask R-CNN and MSRCR image enhancement algorithms. Fan et al. [13] propose a framework for underwater detection that improves the features and the anchor points. Chen et al. [14] built multiple high-resolution and semantically rich feature maps for small underwater objects, which were combined to form a sample-weighted super-network (SWIPENet). To reduce usual message loss and generate high accuracy detecting impact on sea cucumbers, Zeng et al. [15] developed an adversarial occlusion network (AON) and quicker RCNN. The Shortcut Feature Pyramid Network (SFPN) proposed by Peng et al. [16] improves existing multi-scale feature fusion strategies through shortcut connections. Liu et al. [17] reduced small and dense marine benthos omission rate. In conclusion, the two-stage method performs better in terms of detection accuracy. However, the network structure is typically more complicated, with many parameters and computations, and the detection speed is slow.

The one-stage detection algorithms are mainly regression-based target detection algorithms, such as YOLO [18] series, SSD [19], RetinaNet [20], and EfficientDet [21], which generate detection results by computing directly on the image without generating candidate frames [22], [23]. Using a modified YOLOv2 model, Xia et al. [24] put a sea cucumber detection system into practice. Hu et al. [25] embedded dense units in YOLOv4 and applied high-resolution feature maps to achieve the detection of dense microparticles underwater. NgoGia et al. [26] used YOLOv4-tiny and transfer learning to implement a real-time cultured sea cucumber detector to the autonomous underwater vehicle (AUV) and proposed a method based on improved Mosaic data augmentation. Faster MSSDLite [27] was proposed by Cao et al. as a real-time and trustworthy target detector for finding underwater live crabs. Zhang et al. [28] introduced AFFM attention for feature fusion in YOLOv4 to obtain richer semantic information. YOLOv4, with the addition of EC components, is used for underwater dense small particle detection [29]. A transformer mechanism was introduced in the backbone feature extraction network and feature fusion part of YOLOv4 [30]. CornerNet [31] creates a bounding box by combining the upper left and lower right corners of the target. RepPoint [32] determines the location of the nearest bounding box around these points by directly predicting nine representative points. The lightweight design of complex networks is gradually becoming hot to have faster detection speed, such as YOLOv3-tiny, YOLOv4-tiny, YOLO-Fastest. MobileNet [33] based on depth-separable convolution, ShufflNet [34] using inverted residual modules. A lightweight backbone network model was designed to replace the original Yolov5s backbone network using group convolution and inverse residual block [35]. A high-precision and lightweight end-to-end target detection model based on deformable convolution and improved YOLOv4 [36]. Feature fusion module to simplify the design, such as YOLOF that utilizes only one layer of features [37]. In summary, the one-stage method significantly improves detection speed. However, low detection accuracy remains a concern.

According to the above study, there are four prominent problems with underwater target detection algorithms. 1) the uneven distribution of underwater light and large water wave impact lead to more difficult network feature extraction; 2) the complex underwater environment, the low distinction between marine organisms and the environment, which can easily cause the problem of target detection miss detection; 3) the network structure of high precision detection effect is usually complex, and the number of parameters and computation is relatively large; 4) the detection accuracy of lightweight models is usually low and the speed is difficult to improve, and the generalization ability is weak.

Based on the above discussion, we propose the marine organism detection algorithm MODA (Marine Organisms Detection Algorithm), which obtains high-quality prediction results and faster detection speed. For this purpose, we focus on a lightweight network YOLOv4-tiny with high detection accuracy for a one-stage algorithm and make improvements. The contributions and benefits of the method are as follows:

A. A Coordinate Attention module (CA) is constructed, embedded, and connected in the backbone network to obtain accurate target position information. By incorporating the position information into the channel attention mechanism, a feature map that is more sensitive to the direction and position is obtained, enhancing the network’s ability to extract features.

B. The Hybrid Dilated Convolution (HDC) module is constructed and added to the feature pyramid to improve the network’s ability to extract features. Through the HDC module, the deep layer feature map expands the receptive field. It then fuses with the shallow layer feature map to create a new feature map, which accomplishes the goal of expanding the receptive field without overly complicating the network and enhances the effectiveness of target detection.

C. A marine organisms detection method called MODA (Marine Organisms Detection Algorithm) is proposed in order to address the issues of uneven light distribution, huge water wave influence, and complex seabed environments in.
the detection of marine animals. The MODA model has excellent generalization capabilities, requires little processing, and has great accuracy.

The remainder of this paper is laid out as follows: The fundamental network YOLOv4-tiny and the proposed MODA method are introduced in Section II. The datasets and the experimental setup are described in Section III. In Section IV, the experimental results are analyzed, and the method’s applicability is confirmed. The paper is concluded in Section V.

II. MARINE ORGANISMS DETECTION ALGORITHM

In this section, we mainly introduce the marine organisms detection algorithm MODA (Marine Organisms Detection Algorithm) proposed in this paper, as shown in Figure 1. We will introduce it one by one from the following aspects: 1) analyzing the YOLOv4-tiny detection algorithm to prepare for the subsequent improvement of the YOLOv4-tiny network; 2) designing the Coordinate Attention (CA) module for the characteristics of the marine organisms dataset, embedding and connecting it in the network; 3) constructing the Hybrid Dilated Convolution (HDC) structure to obtain the improved YOLOv4-tiny network structure; 4) based on the YOLOv4-tiny network, we propose the marine organisms detection algorithm MODA, as shown in Figure 1.

A. YOLOv4-TINY DETECTION ALGORITHM

The YOLOv4-tiny structure is a condensed form of the YOLOv4 model, which is a lightweight model with a significant increase in detection speed and only one-tenth of the original parameters. Figure 2 (a) depicts the YOLOv4-tiny structure. Its backbone network, the CSPDarknet53-tiny structure, is made up of the residual unit Resblock and the basic convolution DBL, where DBL is a combination of the network’s batch normalization (BN) and activation function Leaky Relu. The residual structure created by ResNet and CSPNet in Figure 2 (b) nests four DBL residuals before performing maximum pooling. As a result, additional features can be recovered without the need for an excessively deep network thanks to the residual structure’s ability to lengthen the gradient flow route and solve the gradient disappearance problem.

In order to extract more features from the YOLOv4-tiny structure, the Feature Pyramid (FPN) structure is utilized. The FPN structure is a top-down and side-to-side connected hierarchical structure. Through the fusion of information from diverse features, it builds high-level semantic features at different scales. Figure 3 depicts its structure. In a convolutional network, shallower feature maps have higher resolution, less semantic information, but stronger detailed information, whereas deeper feature maps have lower resolution and stronger semantic information. The recognition of multi-scale items in photos, especially small objects, is difficult because as the network depth increases, fine details like a position in shallow layers rapidly disappear. While FPN uses nearest neighbor upsampling to gradually increase the size of the feature map in reverse and fuse it with the upper-layer feature map and predict the multi-scale feature maps, the backbone network uses continuous convolution to extract features and downsample, which results in richer feature information and better detection performance on targets of various scales.

B. EMBEDDED CONNECTION COORDINATE ATTENTION MODULE DESIGN

The detection speed is greatly improved when using the YOLOv4-tiny model for marine organism target detection.
Given an input $x$ and adaptive recalibration of channel relations, respectively. Squeezing and excitation, for global information embedding gains a new global attention mechanism, thus strengthening the interdependence between information.

Spatial location information based on channel information, calculate spatial attention, the CBAM mechanism acquires it only has inter-channel data. Later, using convolution to with the model YOLOv4-tiny, we construct the CA module, embed it, and concatenate it for visual activities (SE, CBAM). To enhance performance, the recording of inter-channel interactions is crucial.

As shown in Figure 4, it is simple to integrate into mobile networks so that it may access global information instead of only local information, preventing convolution from doing so. The issue caused by relationships’ absence of long-distance obligations. The recording of inter-channel interactions is crucial for visual activities (SE, CBAM). To enhance performance, we construct the CA module, embed it, and concatenate it with the model YOLOv4-tiny.

Lightweight networks can benefit from performance gains from the still popular SE attention mechanism, although it only has inter-channel data. Later, using convolution to calculate spatial attention, the CBAM mechanism acquires spatial location information based on channel information, thus strengthening the interdependence between information. By incorporating location data into channel attention, CA also gains a new global attention mechanism.

Structurally, the SE block can be split into two steps: squeezing and excitation, for global information embedding and adaptive recalibration of channel relations, respectively. Given an input $x$, the squeezing step for the $c$-th channel can be expressed as shown in 1:

$$Z_c^c(h) = 1 \frac{1}{H \times W} \sum_{0 \leq i < W} x_c(i, h)$$ (1)

where $x$ in $x_c$ comes directly from a convolutional layer with a fixed kernel size, $c$ is the channel and thus can be viewed as a collection of local descriptors, and the squeezing operation makes it possible to collect global information. $H$, $W$ comes from two spatially scoped pool kernels ($H$, 1) or (1, $W$), and we encode each channel along horizontal and vertical coordinates, respectively, given an input $x$, $Z_c^c$ is the output associated with the $c$-th channel of the output.

Global pooling is typically used for channel attention to encoding global spatial information, but it compresses global spatial information into channel descriptors, making it difficult to preserve positional information, which is critical for capturing spatial structure in vision tasks. To encourage the attention block to capture remote interactions spatially through accurate location information, we decompose the global pooling formulated in 1 into a pair of one-dimensional feature encoding operations. Specifically, given an input $x$, we use the two spatial ranges ($H$, 1) or (1, $W$) of the pooling kernel to encode each channel along the horizontal and vertical coordinates, respectively. Thus, the output of the $c$-th channel at height $h$ can be expressed as shown in 2:

$$Z_c^c(h) = \frac{1}{W} \sum_{0 \leq j < W} x_c(h, j)$$ (2)

Likewise, the output of the $c$-th channel of width $w$ can be written as shown in 3:

$$Z_c^c(w) = \frac{1}{H} \sum_{0 \leq i < H} x_c(i, w)$$ (3)

As mentioned above, 2 and 3 enable the global sensory field and encode precise location information. We propose a second transformation called coordinate attention generation to take advantage of the resulting expressive representation. Our design is informed by the following three criteria. First, concerning applications in mobile environments, the new transformation should be as simple and inexpensive as possible. Secondly, it can fully use the captured location information to highlight the region of interest accurately. Last but not least, it should also be able to effectively capture the relationships between channels, which is essential in existing studies.

We first fuse the feature maps in the two directions, then use $1 \times 1$ convolution to transform it to get $F_1$, which is implemented as shown in 4:

$$f = \delta (F_1 ([Z_c^h, Z_c^w]))$$ (4)

where $[\cdot, \cdot]$ represents the fusion operation along the spatial direction, $\delta$ is the nonlinear activation function, here we use the Relu function for activation, and $f \in R^{C \times (H+W)}$ is the intermediate feature map that encodes spatial information in...
the horizontal and vertical directions. Here \( r \) is used to control the reduction ratio of the block size, similar to the SE block. Then, slice \( f \) into two separate tensors \( f^h \in \mathbb{R}^{C/r \times H} \) and \( f^w \in \mathbb{R}^{C/r \times W} \) along the spatial dimension. Then use two \( 1 \times 1 \) convolutions \( F_h \) and \( F_w \) to transform the feature maps \( f^h \) and \( f^w \) to the same number of channels as the input \( x \). The implementation is as follows:

\[
\begin{align*}
g^h &= \sigma \left( F_h \left( f^h \right) \right) \\
g^w &= \sigma \left( F_w \left( f^w \right) \right) 
\end{align*}
\]

where \( \delta \) is the sigmoid activation function. In order to reduce the complexity of the model, an appropriate reduction rate \( r \) is usually used to reduce the number of channels of \( f \). The outputs \( g^h \) and \( g^w \) are then separately expanded and used as attention weights. Finally, the output of the Coordinate Attention module \( Y \) is as shown in 7:

\[
y_c(i, j) = x_c(i, j) \times g^h(i) \times g^w(j) \quad (7)
\]

The superiority of the Coordinate Attention module is as follows. It can be easily integrated into the network to avoid the problem that convolution can only obtain local relations and lacks long-distance dependence; detection results show that the Coordinate Attention module can effectively reduce the phenomenon of missed and false detection.

For YOLOv4-tiny, the initial feature extraction network is the backbone, which establishes a crucial framework for future feature fusion and target location and identification. In order to retain more target information of interest and improve network feature extraction, this article embeds and connects the CA model to the YOLOv4-tiny network.

### C. CONSTRUCTION AND DESIGN OF HYBRID DILATED CONVOLUTIONAL STRUCTURE

As the depth of network continues to increase, we found that feature semantic information is easily lost while extracting features for the purpose of detecting marine organism targets. We attempt to alter the FPN structure to extract features with wider receptive fields in order to address this issue.

The area of a feature on the output feature map that is impacted by the input image is referred to as the receptive field. Increasing the receptive field is a useful strategy if you wish to detect more targets because the effective receptive field is only a portion of the theoretical receptive field and a larger effective receptive field can cover more effective information about the target. When we calculate the receptive field from the back to the front, we only need to regard the layer that needs to be calculated as the output layer \( RF_i \), and then deduce it forward. Its simple formula can be expressed as follows:

\[
RF_i = (RF_{i+1} - 1) \times \text{stride}_i + \text{Ksize}_i 
\]

where \( \text{stride}_i \) represents the step size of the convolution of the \( i \) layer, and \( \text{Ksize}_i \) is the size of the convolution kernel of the \( i \) layer. The receptive field size of the \( i \) convolutional layer is related to the convolution kernel size and stride size of the \( i \) layer, as well as the receptive field size of the \( i+1 \) layer.

The receptive field of a network can theoretically be increased linearly by adding more layers, and the size of the convolution kernel increases the receptive field of each layer. However, too much network depth will complicate the network and slow down the training. Although the spatial resolution would be lower and more semantic information will be lost, downsampling can also quadruple the receptive field. Therefore, in order to increase the receptive field without significantly raising network complexity, we develop a hybrid dilated convolutional structure.

Unlike normal convolution, the dilation convolution sets a parameter called the dilation rate (\( dr \)), which refers to the number of intervals in the kernel (\( dr = 1 \) for a normal convolution). The specific meaning is to fill the convolution kernel with several zeros of \( dr \), which brings a larger receptive field than the original convolution kernel, thus obtaining more information. But single-scale dilation convolution also brings some problems. Because filling too many zeros tends to make the convolution results lose the correlation between them, this result may be useful for some large objects, but it may be detrimental for small targets. To solve this problem, we designed the Hybrid Dilated Convolution (HDC) structure, which uses the convolution of voids with different dilation rate sizes to obtain different sizes of receptive fields, avoiding the problem of local information loss caused by ignoring the information of some pixels. To avoid useless convolution superposition, we set \( dr \) to a sawtooth structure, which can cover as many pixel points as possible. Because we mainly aim at enhancing the capture of small target information, this paper uses \( 3 \times 3 \) dilation convolution with \( dr = 1 \) and \( dr = 2 \), which are connected in series to form a dilation convolution block. The deep layer feature map is fused with the shallow layer feature map into a new feature map after the HDC block increases the receptive field to improve the target detection accuracy. The convolutions we designed for different dilation rate sizes are shown in Figure 5.

We construct the Hybrid Dilated Convolution structure by mixing different dilated rate convolutions. The superiority of the HDC structure: the dilated convolution with different size of dilated rate gets different size of receptive field; the deep features in the feature map increase the receptive field and fuse with the shallow features to form a new feature map; the detection results show that the HDC structure can effectively
extract the feature semantic information and improve the detection accuracy.

Additionally, we visualize feature maps to confirm the HDC’s effect on receptive field expansion. We have visually identified the sea urchins to achieve this. We decide to use a higher resolution feature map for visualization because there are a lot of tiny sea urchins in the image. As seen in Figure 6, there is a clear contrast impact, and the right image can have more goals highlighted on it. This phenomenon demonstrates the beneficial impact of the HDC structure we designed.

D. DESIGN OF THE MARINE ORGANISMS DETECTION MODEL MODA

The most efficient way to address the issues of distorted and blurred underwater image imaging and poor biological detection accuracy is to boost network performance as a whole. To improve the capacity of convolutional feature extraction, the Coordinate Attention mechanism is built into the YOLOv4-tiny model and embedded and connected in the backbone network. Based on this, the FPN portion is developed with a dilation convolutional structure to increase the feature map’s receptive field and preserve more abundant feature information, enhancing the accuracy of the detection of marine species.

The MODA marine organisms detection model is proposed in this paper and is based on the YOLOv4-tiny network. The MODA model architecture is shown in Figure 7.

The following is a discussion of each stage of the model architecture.

1) In Backbone, the DarknetConv2D-BN-Leaky layer is a convolutional layer made by the Normalization layer and activation function LeakyReLU; the Resblock body is composed of four BasicConvolution layers residual nesting, and one MaxPool layer..

2) Input the 416 × 416 × 3 image into Backbone. After two Darknet-Conv2D-BNLeaky layers, convolve to get a 52 × 52 × 128 feature layer; feed this feature layer into Resblock body, and calculate spatial attention to obtain spatial location information through the Coordinate Attention layer. Enhance the interdependence between information, retain more information about the target of interest, and strengthen the network’s ability to extract features.

3) Continue going through the Resblock body and convolve to get the first effective feature layer (26 × 26 × 256).

4) The first effective feature layer is convolved after two Coordinate Attention layers, the Resblock body and DarknetConv2D-BN-Leaky layer, to obtain the second effective feature layer (13 × 13 × 512), which is fed into the Feature Pyramid Network (FPN) network.

5) The second effective feature layer goes through the Hybrid Diated Convolution layer. The deeper features in the feature layer increase the receptive field and merge with the shallow features into a new feature layer, which aims to avoid the problem of missing local information caused by ignoring some pixel information in the feature extraction process, thus improving the detection accuracy. After that, the enhanced feature layer (13 × 13 × 512) is obtained.

6) The enhanced feature layer is passed through the Conv2D layer to obtain a feature layer (13 × 13 × 512) with two routes. The first route is fed into YOLO Head for Detection.

7) The second route is to pass the enhanced feature layer through the Conv2D layer and UpSampling layer, concatenate it with the first effective feature layer, convolve it to get a feature layer (26 × 26 × 256) and feed it into YOLO Head for Detection.

III. EXPERIMENTAL SETUP

In this experimental model training, an i5-12600k CPU, an NVIDIA RTX3080 (12G) GPU, and a CUDA11.4 server were employed as the hardware platform. Windows 10, Python 3.8, and the PyTorch 1.10.2 deep learning framework are all used in the software.

A. DATASETS

In our research, two datasets were used. The first dataset utilized is the official dataset of the 2020 Underwater Robotics Competition (URPC 2020) [27] in China, where all photos were captured in actual underwater settings. The full dataset consists of four different kinds of marine organisms: sea stars, scallops, sea urchins, and sea cucumbers. The collection includes 5400 jpeg photos of marine life with a resolution of 1920 × 1080. One-tenth of the image data from the training set is separated into the validation set, and we divide them into the training set and test set in a ratio of 9:1. This means that 4860 photos are used for the training set, 486 images are used for the validation set, and 540 images are used for the test set.

The second dataset is an open-source collection of marine life data from roboflow called Aquarium (https://public.roboflow.com). The entire dataset consists of 640 unique photos and covers seven different species of marine life: fish, jellyfish, penguins, puffins, sharks, starfish, and stingrays. The volume of increased image data reached 4670 photos using data improvement techniques like flip, rotation, and exposure to increase the original data’s capacity. It was split into a training set and a test set in a 9:1 ratio, as shown in Table 1.
We apply transfer learning to solve the problem of a few labels in marine organisms datasets. With little marine organisms data, the trained model does not work well and leads to overfitting of the model. This problem of training large capacity classifiers on small datasets is solved using transfer learning, which migrates the extracted generic features to the marine organisms detection domain using the network weights after pre-training from other large training sets.

In addition, all information files in the experiment are stored in XML format, containing the corresponding image name, target category name, image size, and location information of the target per frame.

B. NETWORK TRAINING

YOLOv4-tiny usually uses images of a specific size as its inputs, such as 416 × 416 and 608 × 608, because the feature maps of two different scale sizes obtained after downsampling are multiples of 32. In this paper, we use large-scale images such as 1920 × 1080 and 3840 × 2160 for training, so we use 416 × 416 size as input. For more initial model settings refer to Table 1, the total number of epochs is 200, and the initial learning rate is set to 0.005, where the learning rate is reduced to 10% of the original after 50 iterations, as shown in Table 2.

The following explains our decision to use 200 epochs:

1) Epochs are defined as all batches of single training iterations in forward and backward propagation. Suppose the number of epochs is too small. In that case, the network is likely to underfit, and if the number of...
epochs is too large, it is likely to overfit, so it is essential to select the correct number of epochs in training.

2) When training and validating the model, the loss value is a more direct metric to evaluate the model fit and convergence, so we use the Loss-Epochs plot to measure the fit and convergence of the model.

3) As shown in Figure 8, the training loss and validation loss curves flatten out near 170 epochs; near 200 epochs, the curve does not have a decreasing trend and remains stable, the model fits well, and the convergence of the model is good. This proves that the model converges near 200 epochs, which means that the model does not need to continue training and validation to avoid overfitting.

C. EVALUATION INDICATORS

The average precision (AP), mean average precision (mAP), parameters (Param), floating point operations (FLOPs), frame rate (Frame Per Second, FPS), and five performance indicators were used in this experiment to assess the network’s performance.

1) AP is the average accuracy, which is used to evaluate the accuracy of image category detection with a single label. The calculation formula is as follows:

\[
P = \frac{TP}{TP + FP} \tag{9}
\]

\[
R = \frac{TP}{TP + FN} \tag{10}
\]

\[
AP = \int_0^1 P(R)dr \tag{11}
\]

where TP (True Positive) is the number of positive samples included in the anticipated positive samples, FP (False Positive) is the number of negative samples included in the predicted positive samples, and P (precision) is the precision rate. The number, FN (False Negative), denotes how many positive samples were mixed in with the anticipated negative ones.

2) The mAP indicator is a crucial tool for measuring the overall detection accuracy of the model in multi-category target detection and is utilized for evaluating multi-label image classification tasks. The performance of the classification improves with increasing mAP values. The process for calculating mAP is as shown in 12:

\[
mAP = \frac{1}{n} \sum_{n=1}^{n} AP \tag{12}
\]

where n is the number of recognized dataset categories, AP is the area under the PR curve, and P(R) is the PR curve.

3) Rate of frame FPS stands for frames per second, and the higher the FPS, the quicker the detection speed.

4) FLOPs indicate the computational volume, which is used to measure the complexity of the algorithm model, and the computational cost of the algorithm.

IV. EXPERIMENT ANALYSIS

A. MODEL PARAMETER SELECTION

In order to demonstrate better detection, we have experimentally selected the model parameters. As shown in Figure 9, in the first dataset, we experimentally validate not only the tuning of the learning rate (lr) but also the utility of the mosaic data enhancement algorithm and weight decay regularization. Mosaic data enhancement and regularization are usually used to prevent the overfitting problem of the model due to too little data, but since we have data are sufficient, it can be seen that adding mosaic and weight decay (wd), respectively, has a decrease in accuracy, especially the regularization makes the model more unstable. In addition, during the learning rate adjustment process, we found that when lr was 0.01, the loss explosion occurred due to its excessive size, so the model could not be adequately trained after 120 rounds. So we set the lr to 0.005 again to verify the accuracy change brought by a slightly larger learning rate. The accuracy change of the validation set brought by fine-tuning the learning rate in the figure is not significant, but it can still be seen that the accuracy is higher when the lr is 0.001. This also happens in the second dataset.

B. ABLATION EXPERIMENTS

The ablation experiments we designed were used to verify the effectiveness of CA and HDC modules for YOLOv4-tiny. Given the large and dense number of small targets in the
first dataset (objects with an area less than $32^2$ are usually referred to as small targets), the first dataset is chosen for the ablation experiments. For the experiments, we evaluate the model performance using mAP and AP (50:95). The default for all mAPs in this paper is a threshold of 0.5, and AP(50:95) is the average of AP values between 50 and 95 for the threshold. The targets are divided into small (small) (area < $32^2$), medium (medium) ($32^2$ < area < $96^2$), and large (large) (area > $96^2$) categories to be evaluated separately. As shown in Table 3 its evaluation experiment 01 is YOLOv4-tiny without any improvement; in experiment 02 only the CA module is added and it can be seen in the evaluation metrics that compared to experiment 01, the AP metrics of small and medium targets have slightly improved and the mAP metrics have improved by 1.6%. In Experiment 03, only the HDC module is added and compared with Experiment 01, the AP metrics of medium and large targets are slightly improved, and the mAP metrics are improved by 1.7%. In Experiment 04, both CA and HDC modules were added and compared with Experiment 01, we can see that the AP metrics of the small, medium, and large targets have improved significantly, and the mAP metrics have improved by 2.62%, which is a very positive effect.

The attention mechanism is frequently utilized in jobs involving computer vision. The other three attention mechanisms SE, CABM, and ECA are each integrated with the YOLOv4-tiny backbone network in order to demonstrate that the CA module is more effective for our network. Investigate the dataset. Table 4 from the experiments on the URPC dataset reveals that SE and ECA with only a channel attention module have improved accuracy. However, it is not immediately apparent, while CBAM with both channel attention and spatial attention modules have improved mAP value, but the CA performance is superior in comparison.

The mAP values of the seven categories are high and close to the studies on the Aquarium dataset (as shown in Table 5), but the effect of adding the CA module is even better. As can be seen, embedding the CA module in the backbone network is more efficient.

Figure 10 compares the loss and inference accuracy of the upgraded model and the YOLOv4-tiny model; Figure 10 (a) shows that our model converges more quickly and creates less loss. It is clear from Figure 10 (b) that our model is more excellent than the original model. Additionally, as shown in Figure 11, comparison of the AP value and F1 value demonstrates that our model is more stable and has a better impact on detection.

Figure 12 illustrates how we use Heatmaps to detect different models. The detection effect of our model is depicted in the third column of the image; in comparison to other models, our heat area is larger and darker, demonstrating that our model is able to extract a much larger feature receptive field, which helps to improve the targets detection accuracy.

In the first and second columns of the URPC detection results, the original YOLOv4-tiny model missed and made false detections, as shown in Figure 13, we use circle marked. Figure 14 depicts the missed detection of the YOLOv4-tiny model on the Aquarium dataset. The detection of our model is

### TABLE 3. Validity verification experiments.

| Experiment | CA | HDC | mAP(50:95)(%) |
|------------|----|-----|--------------|
|            |    |     | small | medium | large | mAP  |
| 01         |    |     | 13.9  | 31.9   | 38.8  | 74.00 |
| 02         | ✓  |     | 14.4  | 32.5   | 38.8  | 75.60 |
| 03         | ✓  | ✓   | 14.0  | 32.6   | 38.9  | 75.80 |
| 04         | ✓  | ✓   | 15.4  | 33.3   | 40.1  | 76.62 |

### TABLE 4. Attention mechanism effect experiment (URPC).

| Module  | mAP(%) | fish | jellyfish | penguin | puffin | shark | starfish | stingray |
|---------|--------|------|-----------|---------|--------|-------|----------|----------|
| SE      | 60.54  | 75.23| 75.21     | 74.30   | 34.1   |
| CBAM    | 65.00  | 84.64| 80.70     | 74.44   | 32.4   |
| ECA     | 45.25  | 80.98| 75.82     | 48.90   | 62.74  | 32.5   |
| CA      | 65.79  | 85.51| 82.39     | 67.71   | 75.60  | 33.6   |

### FIGURE 10. Comparison of loss and accuracy of different models.

(a) F1 values

(b) AP values

### FIGURE 11. Comparison of performance of different models.

VOLUME 10, 2022

IEEE Access
FIGURE 12. Comparison with the original model of receptive fields. The second column of the image is the original model (YOLOv4-tiny) and the third column is our model.

FIGURE 13. Detection results (URPC dataset). The first column of the image is the original model (YOLOv4-tiny), the second column of the image is the YOLOv4 model and the third column is our model. The missed and false detection are marked with an orange oval in the figure.

depicted in the last column of the image. This demonstrates our model’s strong capacity to find marine organisms. The fact that MODA has a lightweight structure, which lowers training costs while increasing detection speed and accuracy, demonstrates that our model is excellent.

C. TRAINING AND TEST ACCURACY GRAPHS
The experiments were repeated on both datasets, and the results were displayed as Accuracy-Epochs plots.

In the URPC data, as shown in Figure 15, Train accuracy increases rapidly in the first 50 epochs, gradually increases in 50-160 epochs, and remains stable in 160-200 epochs. Test accuracy fluctuates in the first 120 epochs, gradually increases in 120-160 epochs, and stays stable in 160-200 epochs.

In the Aquarium dataset, as shown in Figure 16, the Train accuracy fluctuates in the first 120 epochs, gradually increases in the 120-170 epochs, and remains stable in the 170-200 epochs. Test accuracy fluctuates in the first 115 epochs, gradually increases in the 115-160 epochs, and remains stable in the 160-200 epochs.

D. COMPARISON EXPERIMENT OF DIFFERENT ALGORITHMS
In this section, we compared the MODA model with Faster RCNN [7], SSD [19], YOLOv3 [38], YOLOv4 [39], YOLOv4-tiny [40], YOLOv5 [41] models on URPC dataset.
and Aquarium dataset respectively, and compare Backbone, Params, mAP, and FPS metrics at 200 epochs for each model.

In the URPC dataset, as shown in Table 6. In terms of Backbone, Faster RCNN, YOLOv3 selected large backbone networks, YOLOv4-tiny, YOLOv5, MODA selected lightweight networks, and YOLOv4 used three models: YOLOv4-1 selected large backbone network CSPdarknet53, YOLOv4-2 and YOLOv4-3 selected lightweight networks MobileNetV1 and GhostNet. Regarding Params metrics, the YOLOv4-1 model has the largest number of parameters, reaching a maximum of 69.13M, YOLOv3 has 61.63M, and Faster RCNN and SSD also have a large number of parameters, although they are much lower. In contrast, the number of parameters in the lightweight network is considerably reduced, with the number of parameters in YOLOv4-2 and YOLOv4-3 dropping to less than 20, and the number of parameters in YOLOv5, YOLOv4-tiny, and MODA dropping to less than 10 M. In terms of FPS metrics, the detection speed of the lightweight model is significantly better than that of the large model, a speed improvement of 31 compared to the quickest large model, YOLOv3, and quicker than YOLOv4-tiny allowed MODA model to achieve the fastest FPS metric of 90, the same as YOLOv5. In terms of mAP metrics, among the large networks, the mAP metrics of YOLOv4-1 and YOLOv3 are 75.63% and 73.49%, respectively, which are 9 and 10 percentage points better than Faster RCNN and SSD, respectively; among the lightweight networks, MODA has the best mAP metric of 76.62, which is 2.6 percentage points better than the original model YOLOv4-tiny, and 2.6 percentage points better than the original model YOLOv4-tiny. In the lightweight network, MODA achieves the best mAP index of 76.62, which is 2.6 percentage points better than the original model YOLOv4-tiny, and 2.6 percentage points better than the original model YOLOv4-tiny. It can be found that the lightweight model MODA performs better, with almost the lowest number of parameters and computation, superior in mAP metrics, and much faster, and its high efficiency, lightweight and accuracy make marine biological detection more reliable and low consumption.

In the Aquarium dataset, as shown in Table 7. Regarding Backbone, Faster RCNN, YOLOv3 used a large backbone network, and YOLOv4 used three models: YOLOv4-1 selected the large backbone network CSPdarknet53, YOLOv4-2 and YOLOv4-3 selected the lightweight network MobileNetV1 and GhostNet. In the Params metric, the number of parameters for the lightweight model is much smaller than that of the large network, with only 8.31 for the YOLOv4-tiny model, 6.62 for the YOLOv5 model, and MODA the best 5.95 for MODA, which is one-twelfth of the number of YOLOv3 parameters. In terms of computational cost FLOPs, the large detection model Faster RCNN computationally costs 942 FLOPs (billions of floating point operations), which is costly, while in the lightweight network, the computational cost of YOLOv5 is 3.48, YOLOv4-tiny is 3.45, and the computational cost of our proposed MODA model is only 3.42, which achieves the best computational cost index. In terms of detection speed FPS, the lightweight model is much faster than the large model. The FPS index of YOLOv4-tiny is 81 and YOLOv5 is 85, and the MODA model achieves the same fastest detection speed as YOLOv5, which is improved by 42 compared to YOLOv4 and faster than YOLOv4-tiny. In terms of mAP metrics, the MODA model improves 7% compared to Faster RCNN, 1% compared to YOLOv5, 1% compared to YOLOv4 model, and 6% compared to the original model YOLOv4-tiny.

### Table 6. Comparison of experimental results for different algorithms. (URPC).

| Algorithm       | Backbone          | Params(M) | FPS | mAP(%) |
|-----------------|-------------------|-----------|-----|--------|
| Faster RCNN     | Resnet50          | 28.47     | 11  | 66.16  |
| SSD             | VGG               | 26.15     | 29  | 63.31  |
| YOLOv3          | Darknet53         | 61.63     | 51  | 73.49  |
| YOLOv4-1        | CSPdarknet53      | 69.13     | 40  | 75.63  |
| YOLOv4-2        | MobileNetV1       | 13.13     | 63  | 66.52  |
| YOLOv4-3        | GhostNet          | 11.02     | 55  | 69.96  |
| YOLOv4-tiny     | CSPdarknet53-tiny | 8.31      | 88  | 74.00  |
| YOLOv5          | YOLOv5s           | 6.62      | 90  | 74.56  |
| Ours            | CSPdarknet53-tiny | 5.88      | 90  | 76.62  |

### Table 7. Comparison of experimental results for different algorithms. (Aquarium).

| Algorithm       | Backbone          | Params(M) | PLOPs(G) | FPS | mAP(%) |
|-----------------|-------------------|-----------|----------|-----|--------|
| Faster RCNN     | Resnet50          | 28.48     | 941.16   | 11  | 91.46  |
| YOLOv3          | Darknet53         | 61.95     | 66.17    | 50  | 61.62  |
| YOLOv4-1        | CSPdarknet53      | 24.63     | 32.32    | 39  | 97.81  |
| YOLOv4-2        | MobileNetV1       | 14.12     | 20.21    | 64  | 95.38  |
| YOLOv4-3        | GhostNet          | 12.03     | 18.32    | 55  | 96.52  |
| YOLOv4-tiny     | CSPdarknet53-tiny | 8.31      | 3.45     | 85  | 92.37  |
| YOLOv5          | YOLOv5s           | 6.62      | 3.48     | 85  | 97.62  |
| Ours            | CSPdarknet53-tiny | 5.95      | 3.42     | 85  | 98.41  |
achieving the highest detection accuracy. It is discovered that the lightweight MODA model performs better than other models thanks to its low computational cost, quick processing time, quick detection, superior mAP index, and high detection accuracy.

According to a combination of Params, FLOPs, mAP, and FPS, the MODA model performs better and is more appropriate for the underwater target detection task among the two alternative marine organisms detection data.

V. CONCLUSION
In this research paper, we design the CA mechanism for the characteristics of marine life detection, and also to meet the requirements of accuracy and real-time marine life detection, embedded and connected in the backbone network of YOLOv4-tiny, in addition to designing and constructing the HDC structure, and finally propose the marine life detection algorithm MODA, and validate it on two marine life datasets after performing data enhancement. The effectiveness of MODA is verified on two marine biological datasets after data enhancement. The research contributions of this paper are as follows:

1) For the characteristics of marine target detection, the location information is embedded in the channel attention mechanism, and the CA module is designed, embedded, and connected to the backbone network to strengthen the interrelationship between the information and enhance the convolutional feature extraction capability. To solve the accuracy problem, the HDC structure is designed and constructed to expand the feature map sensory field so as to extract more significant contextual features, and the MODA model is proposed. The model and method proposed in this paper extend the ideas for marine biological detection and have some reference value.

2) Compared with the original YOLOv4-tiny model, the MODA model achieved 76.62% mAP metrics in the URPC dataset, an improvement of 2.6 percentage points, and was able to achieve 98.41% mAP metrics in the Aquarium dataset, an improvement of 6% compared with the original model, indicating that the MODA model has stronger target detection for marine organisms detection. This indicates that the MODA model has better target detection ability for marine organism detection.

3) Compared with other target detection algorithms, the MODA model has more superior detection effect, less computation, high accuracy, and fast speed, which indicates that the MODA model has stronger generalization ability and is more suitable for marine organisms detection tasks.

In future research:
1) In target detection, there are still tiny targets missed. Future model improvements should focus on detecting all targets.
2) Even after using a data augmentation method, the underwater dataset still exhibits blurry images. To further raise the quality of the dataset, we must develop more effective data augmentation techniques.
3) Learn the latest target detection techniques to improve the network structure further and improve the detection accuracy.

REFERENCES
[1] A. Raphael, Z. Dubinsky, D. Iluz, and N. S. Netanyahu, “Neural network recognition of marine benthos and corals,” Diversity, vol. 12, no. 1, p. 29, Jan. 2020.
[2] H. Huang, Q. Tang, J. Li, W. Zhang, X. Bao, H. Zhu, and G. Wang, “A review on underwater autonomous environmental perception and target grasp, the challenge of robotic organism capture,” Ocean Eng., vol. 195, Jan. 2020, Art. no. 106644.
[3] Y. H. Eng, K. M. Teo, M. Chitre, and K. M. Ng, “Online system identification of an autonomous underwater vehicle via in-field experiments,” J. Ocean. Eng., vol. 41, no. 1, pp. 5–17, Jan. 2016.
[4] Y. Li, “A deep learning-based hybrid framework for object detection and recognition in autonomous driving,” IEEE Access, vol. 8, p. 13, 2020.
[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, Jun. 2014, pp. 580–587.
[6] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/GirshickFastRCNN_ICCV_2015_paper.html
[7] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[8] S. A. Ajagbe, O. A. Oki, M. A. Oladipupo, and A. Nwanakwugwu, “Investigating the efficiency of deep learning models in bioinspired object detection,” in Proc. Int. Conf. Electr. Comput. Energy Technol. (ICECET), Prague-Czech Republic, Jul. 2022, pp. 1–6.
[9] X. Li, M. Shang, H. Qin, and L. Chen, “Fast accurate fish detection and recognition of underwater images with fast R-CNN,” in Proc. OCEANS, Washington, DC, USA, Oct. 2015, pp. 1–5.
[10] F. Han, J. Yao, H. Zhu, and C. Wang, “Underwater image processing and object detection based on deep CNN method,” J. Sensors, vol. 2020, pp. 1–20, May 2020.
[11] J. Sensors, vol. 2020, pp. 1–20, May 2020.
[12] J. Sensors, vol. 2020, pp. 1–20, May 2020.
[13] T. A. Kumar, R. Rajmohan, M. Pavithra, S. A. Ajagbe, R. Hodhod, and T. Gaber, “Automatic face mask detection system in public transportation in smart cities using IoT and deep learning,” Electronics, vol. 11, no. 6, p. 904, Mar. 2022.
[14] S. Song, J. Zhu, X. Li, and Q. Huang, “Integrate MSRCR and mask R-CNN to recognize underwater creatures on small sample datasets,” IEEE Access, vol. 8, pp. 172848–172858, 2020.
[15] B. Fan, W. Chen, Y. Cong, and J. Tian, “Dual refinement underwater object detection network,” in Proc. ECCV, vol. 12365, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, Oct. 2020, pp. 275–291.
[16] L. Chen, Z. Liu, L. Tong, Z. Jiang, S. Wang, J. Dong, and H. Zhou, “Underwater object detection using invert multi-class AdaBoost with deep learning,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Glasgow, U.K., Jul. 2020, pp. 1–8.
[17] L. Zeng, B. Sun, and D. Zhu, “Underwater target detection based on faster R-CNN and adversarial occlusion network,” Eng. Appl. Artif. Intell., vol. 100, Apr. 2021, Art. no. 104190.
[18] F. Peng, Z. Miao, F. Li, and Z. Li, “S-FPN: A shortcut feature pyramid network for sea cucumber detection in underwater images,” Exp. Syst. Appl., vol. 182, Nov. 2021, Art. no. 115306.
[19] X. Liu and S. Wang, “A quantitative detection algorithm based on improved faster R-CNN for marine benthos,” Ecol. Informat., vol. 61, Mar. 2021, Art. no. 101228.
[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788.
[21] W. Liu, “SSD: Single shot MultiBox detector,” in Proc. ECCV, vol. 9905, Dec. 2016, pp. 21–37.
[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. ICCV, Oct. 2017, pp. 2980–2988.

[21] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and efficient object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. 2020, pp. 10778–10787.

[22] J. Kang, S. Tariq, H. Oh, and S. S. Woo, “A survey of deep learning-based object detection methods and datasets for overhead imagery,” IEEE Access, vol. 10, pp. 20118–20134, 2022.

[23] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” IEEE Access, vol. 7, pp. 128837–128868, 2019.

[24] C. Xia, L. Fu, H. Liu, and L. Chen, “In situ sea cucumber detection based on deep learning approach,” in Proc. OCEANS, Kobe, Japan, May 2018, pp. 1–4.

[25] X. Hu, Y. Liu, Z. Zhao, J. Liu, X. Yang, C. Sun, S. Chen, B. Li, and C. Zhou, “Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network,” Comput. Electron. Agricult., vol. 185, Jun. 2021, Art. no. 106135.

[26] T. NgoGia, Y. Li, D. Jin, and J. Guo, “Real-time sea cucumber detection based on YOLOv4-tiny and transfer learning using data augmentation,” in Proc. ICSI, Qingdao, China, Jul. 2021, pp. 119–128.

[27] S. Cao, D. Zhao, X. Liu, and Y. Sun, “Real-time robust detector for underwater live crabs based on deep learning,” Comput. Electron. Agricult., vol. 172, May 2020, Art. no. 105339.

[28] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, “Lightweight underwater object detection based on YOLO v4 and multi-scale attentional feature fusion,” Remote Sens., vol. 13, no. 22, p. 4706, Nov. 2021.

[29] G. J. Mao, “A shallow sea creature detection model based on improved YOLO-V4 network,” Trans. Chin. Soc. Agricult. Eng., vol. 37, no. 12, pp. 118–120, Jun. 2021.

[30] X. Fu, Y. Liu, and Y. Liu, “A case study of utilizing YOLOT based quantitative detection algorithm for marine benthos,” Ecol. Inform., vol. 70, Sep. 2022, Art. no. 101603.

[31] H. Law and J. Deng, “CornerNet: Detecting objects as paired keypoints,” in Proc. ECCV, Sep. 2018, p. 17.

[32] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, “RepPoints: Point set representation for object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Seoul, South Korea, Oct. 2019, pp. 9656–9665.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

[34] X. Zhang, X. Zhou, M. Lin, and J. Sun, “ShuffleNet: An extremely efficient convolutional neural network for mobile devices,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 6848–6856.

[35] S. Li, C. Li, Y. Yang, Q. Zhang, Y. Wang, and Z. Guo, “Underwater scallop recognition algorithm using improved YOLOv5,” Aquacultural Eng., vol. 98, Aug. 2022, Art. no. 102273.

[36] S. Zhao, S. Zhang, J. Lu, H. Wang, Y. Feng, C. Shi, D. Li, and R. Zhao, “A lightweight dead fish detection method based on deformable convolution and YOLOv4,” Comput. Electron. Agricult., vol. 198, Jul. 2022, Art. no. 107098.