water

MDPI

Article

Deep Learning-Based Algal Detection Model Development Considering Field Application

Jungsu Park [1,†], Jiwon Baek [2,†], Jongrack Kim [2], Kwangtae You [2] and Keugtae Kim [3,*]

[1] Department of Civil and Environmental Engineering, Hanbat National University, 125 Dongsuh-Blvd, Yuseong-gu, Daejeon 34158, Korea; parkjs@hanbat.ac.kr
[2] UnU Inc., #1005, Samsung IT Valley, 27, Digital-ro 33-gil, Guro-gu, Seoul 08380, Korea; jwbaek.unu@gmail.com (J.B.); jrkim.unu@gmail.com (J.K.); kty.unu@gmail.com (K.Y.)
[3] Department of Environmental & Energy Engineering, Suwon University, 17 Wauan-gil, Bongdam-eup, Hwaseong-si 18323, Korea
* Correspondence: kkt38@suwon.ac.kr; Tel.: +82-31-229-8355; Fax: +82-31-220-2533
† These authors contributed equally to this work.

Abstract: Algal blooms have various effects on drinking water supply systems; thus, proper monitoring is essential. Traditional visual identification using a microscope is a time-consuming method and requires extensive labor. Recently, advanced machine learning algorithms have been increasingly applied for the development of object detection models. The You-Only-Look-Once (YOLO) model is a novel machine learning algorithm used for object detection; it has been continuously improved in newer versions, and a tiny version of each standard model presented. The tiny versions applied a less complicated architecture using a smaller number of convolutional layers to enable faster object detection than the standard version. This study compared the applicability of the YOLO models for algal image detection from a practical aspect in terms of classification accuracy and inference time. Therefore, automated algal cell detection models were developed using YOLO v3 and YOLO v4, in which a tiny version of each model was also applied. The cell images of 30 algal genera were used for training and testing the models. The model performances were compared using the mean average precision (mAP). The mAP values of the four models were 40.9, 88.8, 84.4, and 89.8 for YOLO v3, YOLO v3-tiny, YOLO v4, and YOLO v4-tiny, respectively, demonstrating that YOLO v4 is more precise than YOLO v3. The tiny version models presented noticeably higher model accuracy than the standard models, allowing up to ten times faster object detection time. These results demonstrate the practical advantage of tiny version models for the application of object detection with a limited number of object classes.

Keywords: algal detection; object detection; You-Only-Look-Once algorithm; water quality management; water supply

Citation: Park, J.; Baek, J.; Kim, J.; You, K.; Kim, K. Deep Learning-Based Algal Detection Model Development Considering Field Application. *Water* **2022**, *14*, 1275. https://doi.org/10.3390/w14081275

Academic Editors: Zheng Duan and Babak Mohammadi

Received: 16 March 2022
Accepted: 13 April 2022
Published: 14 April 2022

1. Introduction

Algae are important components of the natural ecosystem. However, their excessive growth causes various harmful effects on drinking water supply systems, including unfavorable taste and odor and release of toxic substances [1–3]. The management of algal blooms is an ongoing issue in water resource management, and the observation of algal cells in water resources is a fundamental approach.

Visual identification using a microscope is one of the most traditional and common methods for algal cell detection, which requires considerable expertise in research as well as time and labor. The development of an automated algal cell detection technology could potentially improve the efficiency of drinking water supply systems by reducing the labor and cost required for in-field algal monitoring.

In the past decade, advanced machine learning models have been continuously developed for the detection of target objects. The convolutional neural networks (CNN) is the

most widely used fundamental algorithm for object detection, and various models based on CNN architecture have been developed [4,5]. The CNN process is composed of feature extraction using convolution and pooling processes. In the convolution process, various types of filters are used to extract the characteristics of the input features, and in the pooling process, the size of the input data is reduced to improve the practical applicability of the complicated deep learning model [6,7].

The main frameworks of CNN-based object detection models have been developed into two different types: one-stage and two-stage models [8]. The two-stage models were first developed where the object detection processes are separated by region proposal and consequent object detection [8]. The two-stage object detection models have been continuously improved from two aspects: to improve the model precision and to reduce the inference time required for object detection. Regions with a CNN (R-CNN) are considered to be one of the first developed two-stage object detection algorithms [9]. In the first stage of the model, approximately 2000 category-independent regions are proposed, where a target object is located using a selective search method. In the second stage of the model, a CNN is used to compute the features for each proposal, and each region is classified using a support vector machine (SVM). One limitation of R-CNN is the relatively slow inference time for object detection, and various algorithms based on R-CNN algorithms have been developed to improve both the model precision and inference time, such as Fast R-CNN and Faster R-CNN [10,11]. R-CNN computes the CNN process for each object proposal, which slows the object detection time of R-CNN. Fast R-CNN was developed to overcome the limitations of R-CNN. Fast R-CNN takes the entire image as an input for the computation of the CNN to produce a feature map. Then, a region of interest (ROI) pooling layer is used to extract the feature vector from the feature map. The extracted feature vector is flattened to produce one-dimensional fully connected layers for the classification of objects using softmax and regression of the boundary box of the object. From these modifications, Fast R-CNN considerably reduces the object detection time compared to R-CNN [11]. Faster R-CNN further reduces the object detection time of the model by the unified use of the region proposal network and Fast R-CNN [10] while these two-stage object detection models still consider the limitation of reducing object detection time.

The reduction in inference time of an object detection model is essential to improve the practical applicability of the model. The You-Only-Look-Once (YOLO) algorithm was developed to reduce the inference time of the object detection process [12]. The YOLO processes the region proposal and object classification at a single stage and is often referred to as a one-stage object detection model. YOLO v1 was the first version of the YOLO model. It was one of the first one-stage object detection models, still considered as a representative in this field. Although YOLO v1 reduced the time required for object detection (i.e., inference time), the relatively low accuracy was considered as a limitation compared to the two-stage model. The second version of the YOLO model, YOLO v2, offered comparable precision with up to 100 times faster inference time than the two-stage models available at that time [4,8,13]. In the past decade, various one-stage object detection models, including the YOLO model series, have been continuously developed. YOLO v3 and v4 models have been the most popular and widely used object detection models up to date [14–16].

CNN-based deep learning is a relatively recent approach in the detection of algae and has been applied to only a limited number of algal classes. Several recent studies have applied deep learning models for the detection of algal cells [17–19]. Medina et al. [18] compared the algal detection accuracy of multilayer perceptron (MLP) and CNNs in underwater pipelines, where CNN showed better accuracy up to 99.46% for testing data. More recently, Sonmez et al. [17] applied various CNN models to classify two different types of algae, Cyanobacteria and Chlorophyta, where a support vector machine (SVM) was used following a CNN process to improve classification accuracy from 98% to 99.66%. These studies have demonstrated the possibility of using CNN-based deep learning models for algal detection.

In practice, both precision and inference time are important factors for evaluating the performance of the object detection models. The YOLO v3 and v4 models offer a tiny version with less accuracy but up to tens of times faster inference time than the standard versions, which is an important advantage, especially for the implementation of the model in field object detection devices for practical use.

Previous studies evaluated the performance of models for algal cell detection; however, studies that analyze the effect of input variable characteristics (e.g., relative size of algal cell) on the algal detection model are rare. This study compared the model performances of YOLO v3, YOLO v3 tiny, YOLO v4, and YOLO v4 tiny, considering the field applicability of the models, emphasizing two aspects. First, the model performances of the standard and tiny versions were compared. Second, the effect of relative object size within the input image was tested. The model performances were evaluated based on the classification of 30 algal genera. The characteristics of each model were compared from a practical aspect in terms of classification accuracy and inference time.

2. Material and Methods

2.1. Model Selection

The YOLO models were improved from the previous version. The first version, YOLO v1, processes object detection as a single regression to provide fast object detection, using rectangular bounding boxes for the detection of objects [12]. The input image was resized to 448 × 448 pixels, and the resized image was divided into an S × S grid, where each grid cell had B bounding boxes for the detection of an object. The model was optimized to minimize the total loss (loss $_{\text{total}}$) using a loss function composed of multiple parts, which compares the observed location and size of the target object referred to as ground truth with model prediction using Equations (1)–(4) [12]. Each grid cell calculated the confidence score (OC), which is the multiplication of the probability that the bounding box contains an object and the intersection over union (IOU). The IOU is the ratio of the overlapped area and union area between the ground truth and the predicted bounding box. Equation (2) represents the sum of the loss from the deviation between the bounding box and ground truth of the target object, Equation (3) represents the sum of the object confidence score loss in each bounding box, and Equation (4) represents the sum of object classification loss in each bounding box.

$$\text{Loss}_{\text{total}} = \text{Loss}_{\text{coordinate}} + \text{Loss}_{\text{object confidence}} + \text{Loss}_{\text{classification}} \quad (1)$$

$$\begin{aligned}\text{Loss}_{\text{coordinate}} = {} & \alpha_{coord} \sum_{i=0}^{S\times S} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(x_{i,obj} - x_{i,pred}\right)^2 + \left(y_{i,obj} - y_{i,pred}\right)^2\right] \\ & + \alpha_{coord} \sum_{i=0}^{S\times S} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_{i,obj}} - \sqrt{w_{i,pred}}\right)^2 + \left(\sqrt{h_{i,obj}} - \sqrt{h_{i,pred}}\right)^2\right]\end{aligned} \quad (2)$$

$$\text{Loss}_{\text{object confidence}} = \sum_{i=0}^{S\times S} \sum_{j=0}^{B} 1_{ij}^{obj} \left(OC_{i,obs} - OC_{i,pred}\right)^2 + \alpha_{noobj} \sum_{i=0}^{S\times S} \sum_{j=0}^{B} 1_{ij}^{noobj} \left(OC_{i,obs} - OC_{i,pred}\right)^2 \quad (3)$$

$$\text{Loss}_{\text{classification}} = \sum_{i=0}^{S\times S} 1_{i}^{obj} \sum_{OC\in\, classes} \left(p_{i,obj}(c) - p_{i,pred}(c)\right)^2 \quad (4)$$

where the subscript *obj* represents the parameters related to the ground truth; the subscript *pred* represents the parameters predicted from the model; x_i, y_i coordinates represent the center of each bounding box relative to the bounds of the grid cell; w_i, h_i are the width and height of the bounding box relative to the entire image size, respectively; *OC* is the object confidence score; $p_i(c)$ is the object classification score that represents the probability of class prediction, α_{coord} and α_{noobj} are the weight factors; 1_i^{obj} denotes that the loss value is computed if an object appears in cell *i*; 1_{ij}^{obj} denotes that the loss value is computed when the bounding box *j* predictor in cell *i* is responsible for that prediction; 1_{ij}^{noobj} denotes that the loss value is computed when the bounding box *j* predictor in cell *i* is not responsible

for that prediction. The value of α_{noobj} is much smaller than that of α_{coord} which limits the effect of loss for boxes that do not contain objects.

The second version of YOLO, called YOLO9000, made several significant changes from the previous version. The main model framework often referred to as the backbone of YOLO9000 is called Darknet-19 [13]. One of the critical changes from the previous version was replacing the fully connected layer after feature extraction to a convolutional layer, and anchor boxes were used to predict the bounding boxes. In YOLO9000, the priors of the bounding box dimensions were determined by k-means clustering of the input images in the training dataset.

The third version of YOLO model, YOLO v3, also followed the approach of YOLO9000 to predict the bounding boxes. The main model framework was replaced with a more complicated network called the Darknet-53 [20]. The model structure of the YOLO v3 is shown in Figure 1. The YOLO v3 model performance was better than the previous versions, especially for small-size target objects, by using three different scaled feature maps for object detection, a similar concept to the feature pyramid network [14,20,21].

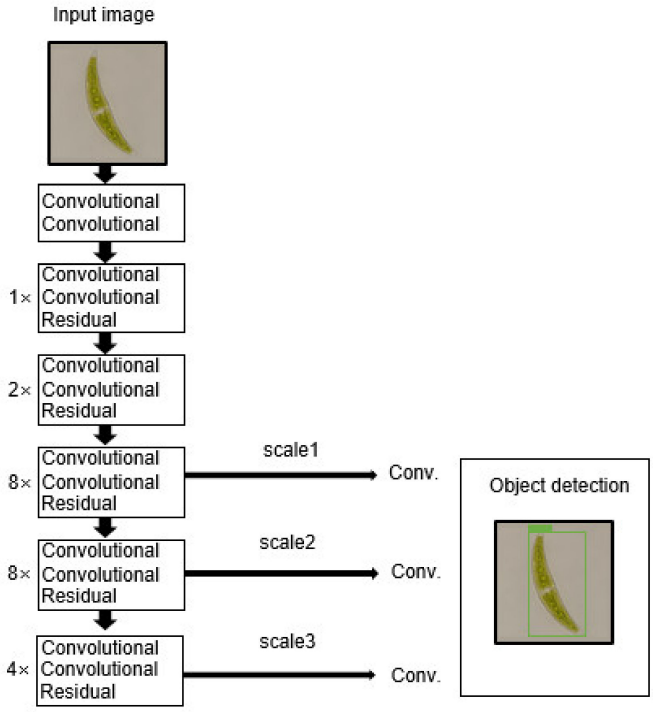

Figure 1. A schematic of YOLO v3 architecture.

Bochkovskiy et al. [22] developed an improved one-stage detection model called YOLO v4 based on the core concept of YOLO v3. The YOLO v4 model used CSPDarknet-53 as the backbone. The accuracy and speed of object detection were improved by implementing novel techniques to YOLO v4, such as spatial pyramid pooling (SPP) and path aggregation network (PAN). The SPP improved the object detection accuracy by increasing the receptive field while maintaining the operation speed, and the PAN provided a shortcut path in the feature extraction process, resulting in improved fusion of extracted features [16,22]. YOLO v4 also applied Mosaic, a new input data augmentation method, where the four different images were mixed during training, thus enabling the detection of objects outside of the normal shape [22].

Both YOLO v3 and v4 provided a tiny version [23–25]. The tiny version models followed the structure of the standard versions while reducing the number of convolutional layers, as shown in Table 1. YOLO v3 used three different scales (i.e., 52 × 52, 26 × 26, 13 × 13) outputs for prediction, while the tiny versions used two different scales (i.e., 26 × 26, 13 × 13) outputs [23–25]. The developer of YOLO v3 compared the model

performance of the standard version YOLO v3 and the tiny version [24]. The mean average precision (mAP) of the standard version YOLO v3 ranged from 51.5 to 60.6 based on the resolution of input images, where the detection speed ranged from 20 to 45 fps. The mAP and detection speed of YOLO v3 tiny were 33.1 and 220, respectively. The fast object detection time is an important advantage of the tiny version model for field applications including real-time monitoring. In this study, the open-source one-stage object detection models, YOLO v3, YOLO v4, YOLO v3 tiny, and YOLO v4 tiny, were used for the development of an algal cell detection model.

Table 1. The comparison of the models used for algal detection in this study.

Model	YOLO v3	YOLO v3 Tiny	YOLO v4	YOLO v4 Tiny
Backbone (Number of convolutional and pooling layers)	Darknet-53 (75)	7 convolutional and 6 max pooling layers (13)	CSPDarknet-53 (110)	CSPDarknet-53-tiny (21)
Resolution of input image	416 × 416	416 × 416	608 × 608	416 × 416

2.2. Data Sources

The images of 30 algal genera collected by the Nakdonggang National Institute of Biological Resources in South Korea were used for model development (Table 2). Images were collected using a microscope (Eclipse Ni, Nikon, Tokyo, Japan). The number of images was 437, and the total number of labeled algae was 1164.

Table 2. Algal images used for model development.

Genera	Number of Image		Number of Label	
	Train	Test	Train	Test
Acutodesmus obliquus	8	2	62	11
Ankistrodesmus falcatus	12	4	20	6
Chlamydomonas asymmetrica	16	6	55	6
Chlorella vulgaris	12	3	60	11
Chlorococcum loculatum	12	3	53	8
Chroomonas coerulea	12	4	98	15
Closterium sp.	7	2	11	2
Coelastrella sp.	7	3	14	5
Coelastrum astroideum var. rugosum	6	2	27	5
Cosmarium sp.	13	5	46	6
Cryptomonas lundii	22	6	22	6
Desmodesmus communis	16	3	18	3
Diplosphaera chodatii	2	1	32	6
Eudorina unicocca	6	1	40	6
Euglena sp.	28	7	28	7
Kirchneriella aperta	13	3	30	17
Lithotrichon pulchrum	8	2	42	11
Micractinium pusillum	9	1	44	5
Micrasterias sp.	6	2	6	2
Monoraphidium sp.	21	8	40	15
Mychonastes sp.	9	4	39	13
Nephrochlamys subsolitaria	6	2	24	5
Pectinodesmus pectinatus	32	7	39	8
Pediastrum duplex	11	4	13	4
Pseudopediastrum boryanum	10	3	10	3
Scenedesmus sp.	6	3	9	4
Selenastrum capricornutum	6	3	31	7
Sorastrum pediastriforme	8	1	8	1
Tetrabaena socialis	11	2	16	2
Tupiella speciosa	4	1	25	2
Total	339	98	962	202

2.3. Model Training Environment

In this study, an algal cell detection model was developed for implementation on a field detection device. Thus, the model was developed using a different programming language compatible with the individual devices used for each step.

The algal cells within an image were labeled using a labeling program developed in this study, where the coordinate ground truth and class were saved in txt format. The labeling program was developed using C # program language. The ratio of the data used for training and testing of the model was 78:22 for the 437 images. The hyperparameters for the model simulation were determined as the default values provided by the developer of each model (Table 3).

Table 3. Hyperparameters used for model simulation.

Models	YOLOv3	YOLOv3 Tiny	YOLOv4	YOLOv4 Tiny
Batch size	64	32	64	32
Learning rate	0.001	0.001	0.0013	0.00261
max_batch class	60,000	60,000	60,000	500,200

The YOLO v3, YOLO v3 tiny, YOLO v4, and YOLO v4 tiny models were trained in the environment of OpenCV 4.4 and NVIDIA GPU toolkit 1.1 using a computer Intel Xeon Silver 4216, 2.1 GHz CPU, 64GB RAM with an NVIDIA Quadro RTX 4000 8GB GPU. The models were programmed using the C++ language.

The trained weight file was downloaded and used to test the model inference. The inference program was programmed using C # language to provide the genus of detected algal images, object classification score, and coordinates of the bounding box so that the output result can be visualized, as shown in Figure 2.

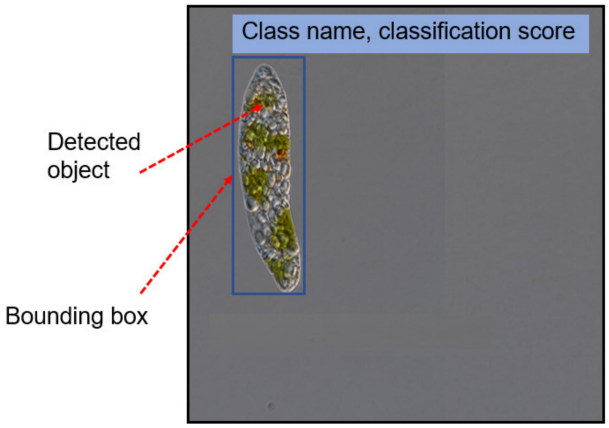

Figure 2. Illustration of the object detection result.

2.4. Effect of the Relative Object Size within the Image

The effect of the relative size of algal cells within the image on model performance was tested using images of various relative object sizes within an image. An image including only one algal cell object was selected where the object was enlarged or reduced so that the length of the longest axis of the bounding box became 10%, 30%, 50%, 70%, and 90% of the entire image size (Figure 3). One image per algal genera, *Closterium* sp. and *Pseudopediastrum boryanum*, was selected for the analysis.

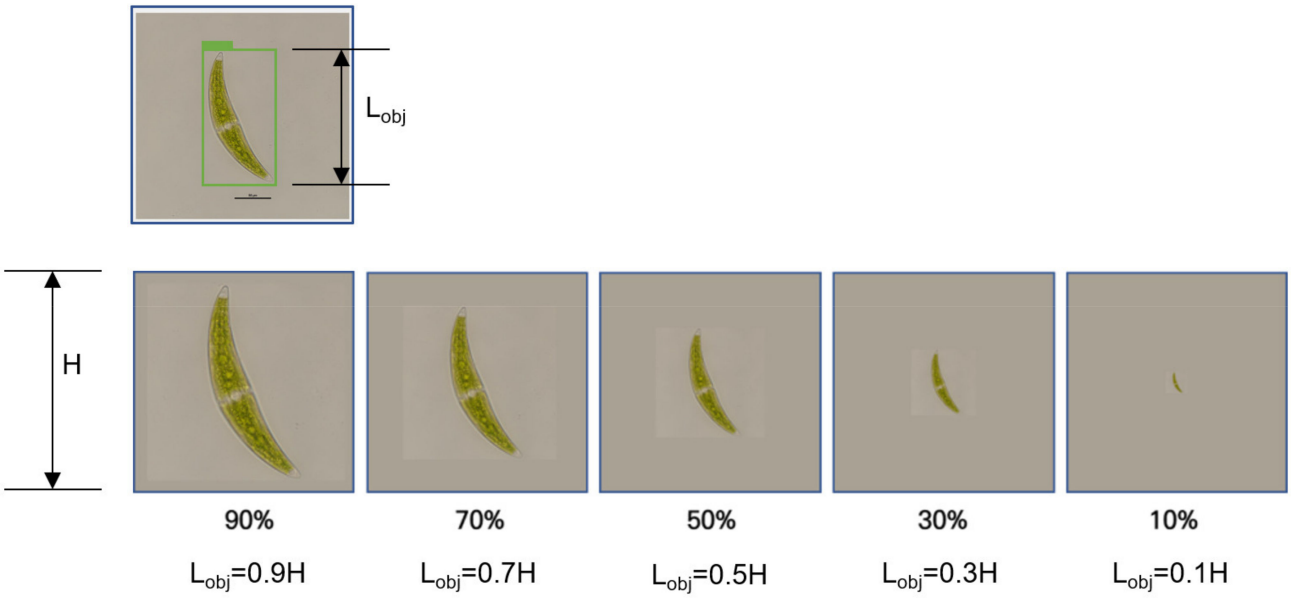

Figure 3. An example image set for testing effect of object size.

2.5. Model Evaluation

The model performance was evaluated using the mAP with an IOU of 0.5, which is the most commonly used indicator for the evaluation of object detection models.

The average precision (AP) of each object class is calculated from the sum of the area under the precision–recall curve.

Precision and recall are defined as in Equations (5) and (6).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

$$\mathrm{R}ecall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

where true positive (TP) is the number of observed positive values that were correctly predicted; false positive (FP) is the number of observed positive values that were incorrectly predicted; and, false negative (FN) is the number of observed negative values that were incorrectly predicted.

A precision–recall curve presents the relationship between precision and recall as a 1 to 1 curve with recall as the *x*-axis and precision as the *y*-axis, which shows the changed value of precision through the change of recall from 0 to 1. The AP of the model was calculated from the sum of the area under the precision–recall curve [26]. The mAP was calculated from the average AP for all genera.

3. Results and Discussion

3.1. Model Simulation Results

The performances of the four models are compared in Table 4. The training time of models were 145 h, 19 h, 213 h and 5 h for YOLO v3, YOLO v3 tiny, YOLO v4, and YOLO v4 tiny, respectively. The YOLO v3-tiny showed more than seven times faster training time than the standard version, and YOLO v4-tiny showed more than 40 times faster training time.

Table 4. Summary of model evaluation results.

Model	YOLO v3	YOLO v3 Tiny	YOLO v4	YOLO v4 Tiny
mAP (IOU 0.5)	40.9	88.8	84.4	89.8
Inference speed (fps)	2.0	4.1	1.7	4.0

The results revealed that tiny versions of the models offered higher accuracy and required lower detection time, in contrast to previous findings (Table 5). In this study, YOLO v3-tiny showed approximately two times higher accuracy and object detection speed. The YOLO v4-tiny also shows better performance compared to the standard version with a detection speed that is approximately two times faster detection speed as 4.0 fps. The tiny versions showed a considerably short model training time, which is also an important advantage for model development and optimization.

Table 5. YOLO model performances (Accuracy(mAP)/Speed(fps)) comparison in previous studies.

YOLO v3	YOLO v3 Tiny	YOLO v4	YOLO v4 Tiny	Dataset	References
51.5–60.6/ 20–45	33.1/ 220			MS COCO (80 classes, 83K images)	Remon [24]
52.5/49	30.5/277	64.9/41	38.1/270	MS COCO (80 classes, 83K images)	Jiang et al. (2020)

Remon (2020), the original developer of the YOLO models(v1–v3), presented that YOLO v3-tiny shows up to approximately ten times faster object detection speed, while only about half the accuracy (Map) of the standard version for the model using MS COCO data with 80 classes (Table 5). In another previous study, Jiang, Zhao, Li and Jia [25] compared the model performance of YOLO models using the same MS COCO dataset. The mAPs of the tiny versions were reduced to approximately 58% of the standard versions (Table 5).

The results of this study showed that the performances of tiny version models were better than those of the standard version, in contrast to the results of previous studies (Tables 4 and 5). The MS COCO dataset was composed of 80 classes of objects, including persons, dogs, and cars, having complicated characteristics. On the other hand, the dataset used in this study had a much smaller number (30) of classes. The shape of the algal cell is also relatively simple, such as a circular shape and an eclipse shape, compared to objects in MS COCO dataset (e.g., person, bicycle, bird, etc.).

The convolutional layers in the YOLO model extract the characteristics of an input image. An increase in the number of convolutional layers results in a decrease in the resolution of images and an increase in the channel depth. The tiny version models use fewer convolutional layers than the standard version. Thus, the tiny version models tend to have less information while maintaining relatively higher-resolution images. It is estimated that the algal cell images used in this study required less information for classification because they had a smaller number of classes and a relatively simple morphological shape than the MS COCO dataset used in previous studies. Thus, the tiny version models exhibited better performance in this study. The results verify the advantage of tiny version models in practical aspects, such as the implementation of the model in field object detection devices that require both accuracy and fast detection time.

3.2. Effect of the Relative Object Size

The model performances varied considerably through the relative object size within the image as shown in the example images (Figures 4–6) where the percent value in the top of the bounding box represents probability of class prediction. The model performance with relatively large-size ($\geq$10% of the image size), densely distributed relatively small-size, sparsely distributed relatively small-size algal cell objects are presented in Figures 4–6, respectively. The tiny version models showed considerably better performance than the

standard version models for the detection of relatively small sized objects (<10% of the image size), in this case, algal cells (Figures 5 and 6). The two tiny version models (v3 tiny and v4 tiny) well detected algal cells, where YOLO v3 failed to detect both the small-size densely distributed target objects and the small-size sparsely distributed objects (Figures 5 and 6). YOLO v4 showed a better performance than YOLO v3 for the detection of small-size objects. These examples suggested that the accuracy of the models for algal cell detection is influenced by the relative size of the object within the image rather than the crowdedness.

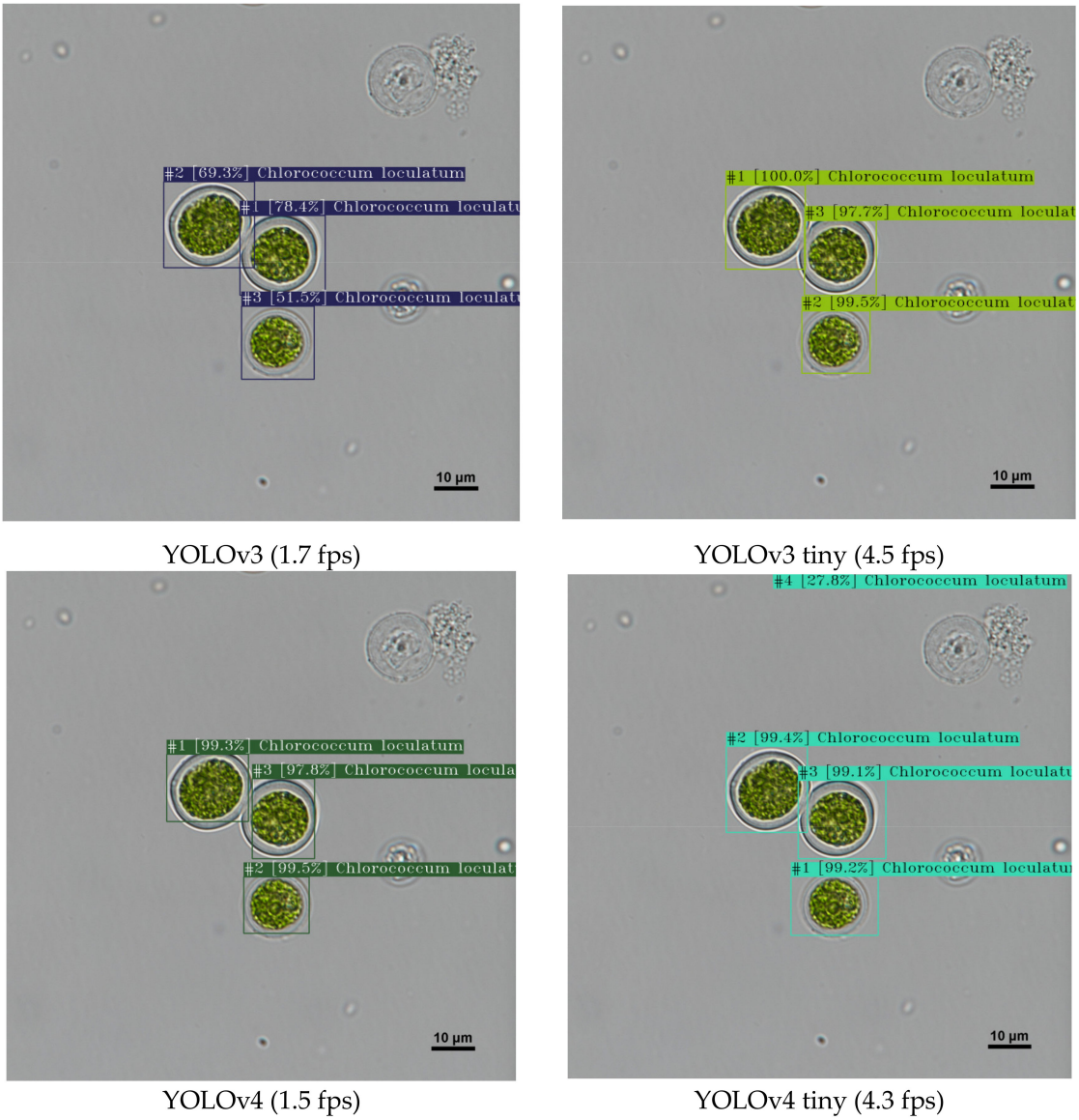

Figure 4. Comparison of the algal detection model performance with relatively large-size (≥10% of the image size) objects.

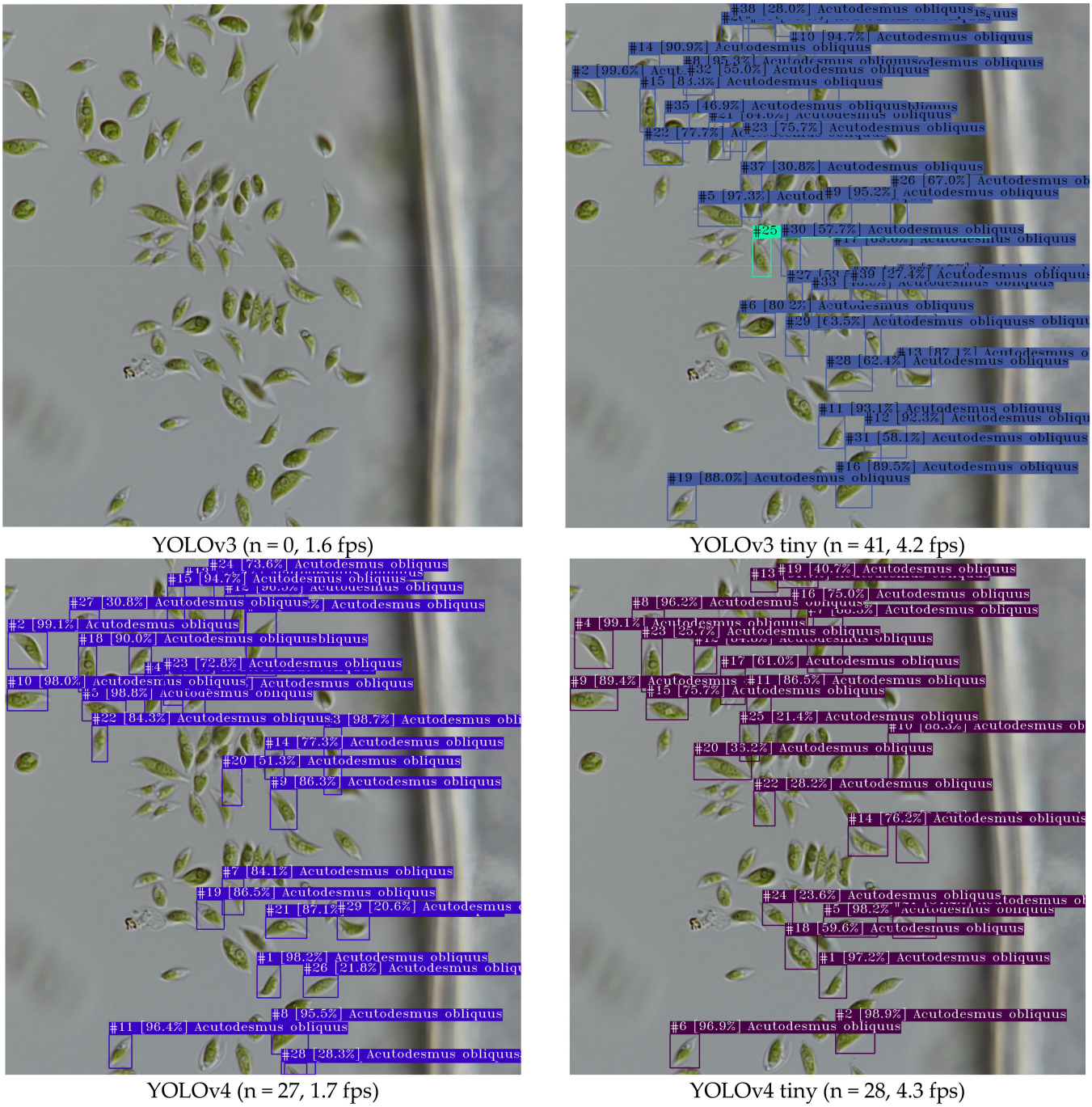

Figure 5. Comparison of the algal detection model performance with densely distributed relatively small-size algal cell objects.

The relative ratios of the number of detected target cells in each model to YOLO v4 model were compared for quantitative analysis about the effect of relative object size on the model performance (Figure 7). The target objects were classified to three groups by relative object sizes (i.e., <15%, 15%~45% and <45%) which were determined from the average length of the bounding box detected by YOLO v4 model for 30 algal genera. The three models showed similar detection performances for various relative object size, while YOLO v3 showed noticeably low detection efficiency for the small size (<15%) objects than the other three models.

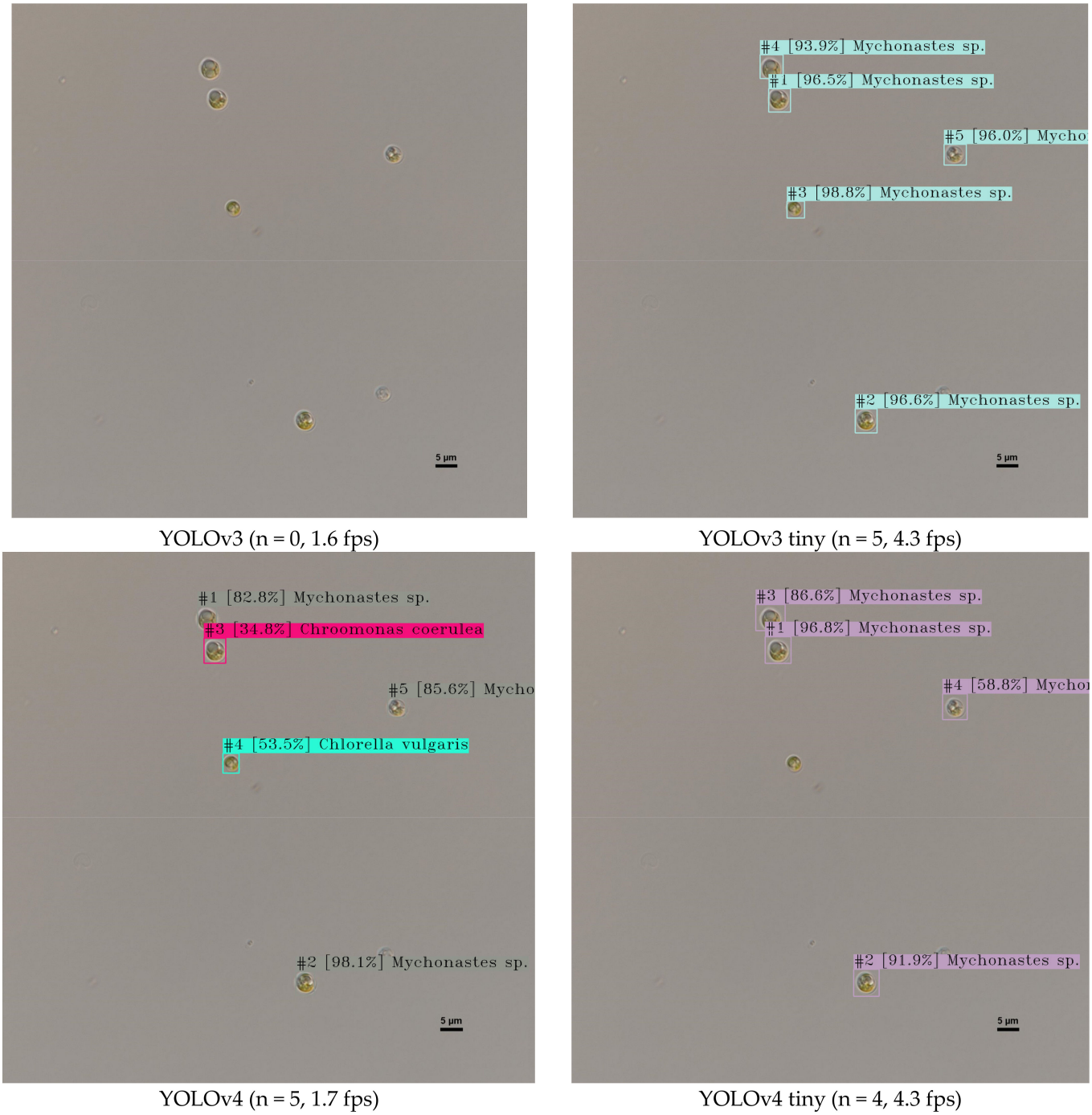

Figure 6. Comparison of the algal detection model performance with sparsely distributed relatively small-size algal cell objects.

For further understanding about the effect of relative object size, the model performances were tested using the image including an object with 10%, 30%, 50%, 70%, and 90% of the size of the entire image. The images of two genera *Closterium* sp. and *Pseudopediastrum boryanum* were used for the test, where the relative object sizes were 61.7% and 35.2% of the entire images, respectively (Figure 8).

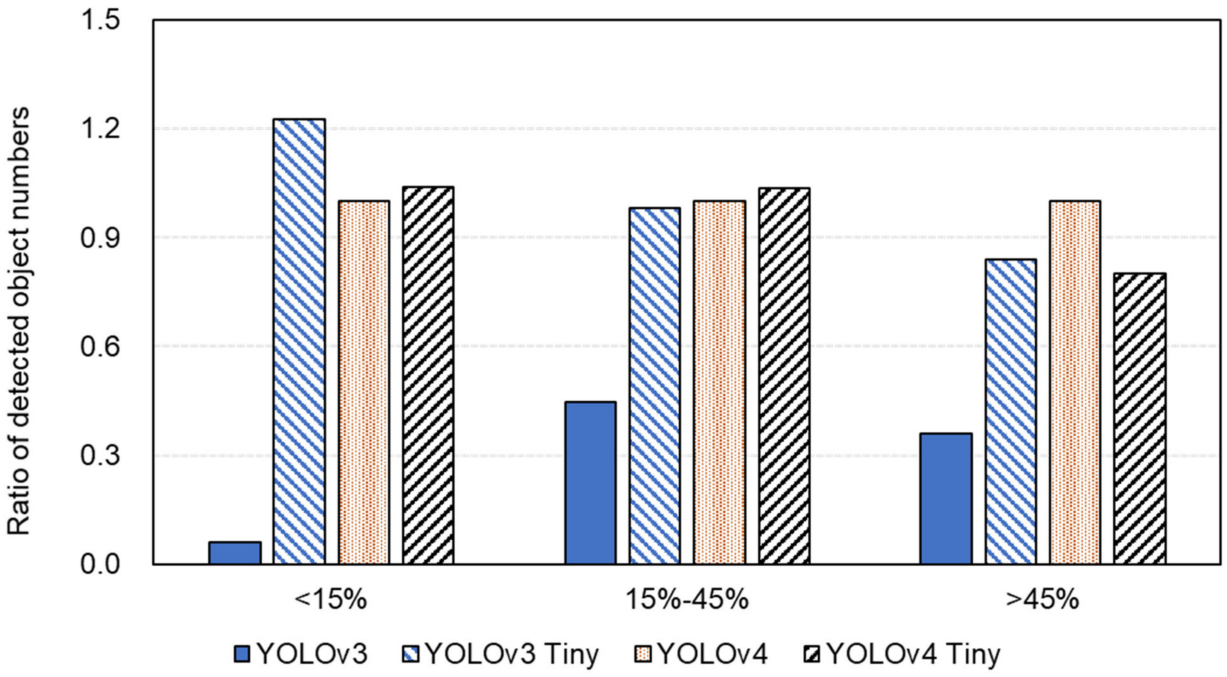

Figure 7. Comparison of the algal detection model performance with different relative object size.

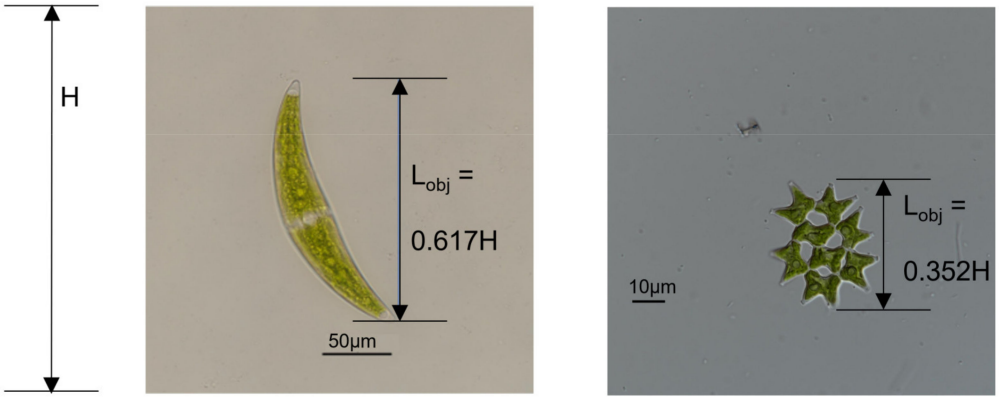

Figure 8. The original images used to test the effect of relative object size on model performance. (**a**) *Closterium* sp. (**b**) *Pseudopediastrum boryanum*.

The object detection accuracy of the YOLO model for various object sizes was tested with three sample images used for comparing the model performance (Table 6).

For the image of *Closterium* sp., where the relative object size was 61.7% of the entire image, the three YOLO models (i.e., v3-tiny, v4, and v4-tiny) stably detected the reduced image with relative object sizes of 10%, 30%, and 50% of the entire image size, but did not detect the enlarged images with 70% and 90% relative object size (Table 6). YOLO v3 did not detect objects in this sample image.

The model also showed a similar stable performance with relative object sizes of 10%, 30%, and 50% of the entire image for the second example image of *Pseudopediastrum boryanum* (Table 6). In this image, YOLO v3 did not detect enlarged or reduced images. The two tiny version models detected the object with a relative object size from 10 to 70% but failed to detect the image with a relative object size of 90%.

The results indicated that YOLO v4 performed better for images with various relative object sizes. The tiny versions had a more stable performance than the standard versions for both v3 and v4 models.

Table 6. The performance comparison of YOLO models for various relative object sizes.

Genera	Relative Size (%)	Probability of Class Prediction (%)			
		YOLO v3	YOLO v3-Tiny	YOLO v4	YOLO v4-Tiny
Closterium sp.	61.7 *	ND	98.3	99.6	90.6
	10	ND	51.1	68.9	81.2
	30	ND	99.9	98.6	98.1
	50	ND	41.9	97.3	55.5
	70	ND	FD	FD	ND
	90	ND	ND	FD	ND
Pseudopediastrum boryanum	35.2 *	81.5	100.0	99.3	98.9
	10	FD	96.3	99.9	92.5
	30	ND	100.0	99.7	99.8
	50	ND	100.0	99.0	100.0
	70	ND	99.9	FD	98.7
	90	ND	FD	FD	AD

*: original image; FD: falsely detected; ND: not detected.

4. Conclusions

In this study, algal cell detection models were developed using four YOLO model algorithms: YOLO v3, YOLO v3-tiny, YOLO v4, and YOLO v4-tiny.

The results showed that YOLO v4 had a better performance than YOLO v3, both in general and for the detection of small-sized objects. The tiny version models worked with noticeably higher model accuracies and ten times faster object detection time compared to previous studies. The performance of the tiny version models could be attributed to the fact that fewer algal genera with less complicated morphological shapes were screened. This resulted in extracting less information from convolutional layers during the object detection process for the classification of algal cell images.

Further analyses of images with small objects and various relative object sizes also indicated that tiny version models have a better performance than standard version models, especially for small size object.

In general, the type of dominant algal genera growing over a period of time at one site is often limited; thus, tiny version models can provide sufficient performance for field use in water quality monitoring. The results of this study verified the practical advantages of tiny version models for algal cell detection with a limited number of genera for classification.

The performance of the object detection model is affected by the various characteristics of the input image dataset. Thus, the result of this study may not generally be applicable to other academic/industrial fields or datasets with a larger number of classes. However, the results of this study provide a useful perspective to improve the practical applicability of the object detection model in future studies.

Author Contributions: J.P.: conceptualization, investigation, writing original draft preparation. J.B.: methodology, data curation. J.K.: visualization, investigation. K.Y.: software, validation. K.K.: conceptualization, writing review and editing, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by Korea Ministry of Environment (MOE) (2020003030006).

Data Availability Statement: Due to confidentiality agreements, supporting data can only be made available to bona fide researchers, subject to a non-disclosure agreement.

Acknowledgments: This work was supported by Korea Environment Industry & Technology Institute (KEITI) through Aquatic Ecosystem Conservation Research Program, funded by the Korea Ministry of Environment (MOE) (2020003030006).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Codd, G.A.; Morrison, L.F.; Metcalf, J.S. Cyanobacterial toxins: Risk management for health protection. *Toxicol. Appl. Pharmacol.* **2005**, *203*, 264–272. [CrossRef] [PubMed]
2. Paerl, H.W.; Otten, T.G. Harmful cyanobacterial blooms: Causes, consequences, and controls. *Microb. Ecol.* **2013**, *65*, 995–1010. [CrossRef] [PubMed]
3. Otten, T.G.; Crosswell, J.R.; Mackey, S.; Dreher, T.W. Application of molecular tools for microbial source tracking and public health risk assessment of a *Microcystis* bloom traversing 300 km of the Klamath River. *Harmful Algae* **2015**, *46*, 71–81. [CrossRef]
4. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
5. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 818–833.
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
8. Sultana, F.; Sufian, A.; Dutta, P. A review of object detection models based on convolutional neural network. In *Intelligent Computing: Image Processing Based Applications*; Springer: Singapore, 2020; Volume 1157, pp. 1–16.
9. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 580–587.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
11. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1440–1448.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
13. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
14. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
15. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car detection using unmanned aerial vehicles: Comparison between faster R-CNN and yolov3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; pp. 1–6.
16. Wu, D.; Lv, S.; Jiang, M.; Song, H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* **2020**, *178*, 105742. [CrossRef]
17. Sonmez, M.E.; Eczacıoglu, N.; Gumuş, N.E.; Aslan, M.F.; Sabanci, K.; Aşikkutlu, B. Convolutional neural network-Support vector machine based approach for classification of cyanobacteria and chlorophyta microalgae groups. *Algal Res.* **2021**, *61*, 102568. [CrossRef]
18. Medina, E.; Petraglia, M.R.; Gomes, J.G.R.; Petraglia, A. Comparison of CNN and MLP classifiers for algae detection in underwater pipelines. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–6.
19. Park, J.; Baek, J.; You, K.; Nam, S.W.; Kim, J. Microalgae Detection Using a Deep Learning Object Detection Algorithm, YOLOv3. *J. Korean Soc. Environ. Eng.* **2021**, *37*, 275–285.
20. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
21. Zhao, K.; Ren, X. Small aircraft detection in remote sensing images based on YOLOv3. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Kazimierz Dolny, Poland, 21–23 November 2019; p. 012056.
22. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
23. Bochkovskiy, A.D. Open Source Neural Networks in Python. 2020. Available online: https://github.com/AlexeyAB/darknet (accessed on 19 January 2021).
24. Remon, J.D. Open Source Neural Networks in C, 2013–2016. Available online: https://pjreddie.com/darknet/ (accessed on 15 July 2020).
25. Jiang, Z.; Zhao, L.; Li, S.; Jia, Y. Real-time object detection method based on improved YOLOv4-tiny. *arXiv* **2020**, arXiv:2011.04244.
26. Ozenne, B.; Subtil, F.; Maucort-Boulch, D. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. *J. Clin. Epidemiol.* **2015**, *68*, 855–859. [CrossRef] [PubMed]