Journal of *Marine Science and Engineering*

MDPI

Article

Artificial Neural Network for Glider Detection in a Marine Environment by Improving a CNN Vision Encoder

Jungwoo Lee [1], Ji-Hyun Park [1], Jeong-Hwan Hwang [1], Kyoungseok Noh [1], Youngho Choi [1] and Jinho Suh [2,*]

[1] Smart Mobility Research Center, Korea Institute of Robotics and Technology Convergence (KIRO), Pohang 37666, Republic of Korea; ricow@kiro.re.kr (J.L.); jhpark87@kiro.re.kr (J.-H.P.); hwangjh@kiro.re.kr (J.-H.H.); ksno@kiro.re.kr (K.N.); rockboy@kiro.re.kr (Y.C.)
[2] Major of Mechanical System Engineering, Pukyong National University, Busan 48513, Republic of Korea
* Correspondence: suhgang@pknu.ac.kr

Abstract: Despite major economic and technological advances, much of the ocean remains unexplored, which has led to the use of remotely operated vehicles (ROVs) and gliders for surveying. ROVs and underwater gliders are essential for ocean data collection. Gliders, which control their own buoyancy, are particularly effective unmanned platforms for long-term observations. The traditional method of recovering the glider on a small boat is a risky operation and depends on the skill of the workers. Therefore, a safer, more efficient, and automated system is needed to recover them. In this study, we propose a lightweight artificial neural network for underwater glider detection that is efficient for learning and inference. In order to have a smaller parameter size and faster inference, a convolutional neural network (CNN) vision encoder in an artificial neural network splits an image of a glider into a number of elongated patches that overlap to better preserve the spatial information of the pixels in the horizontal and vertical directions. Global max-pooling, which computes the maximum over all the spatial locations of an input feature, was used to activate the most salient feature vectors at the end of the encoder. As a result of the inference of the glider detection models on the test dataset, the average precision (AP), which indicates the probability that an object is located within the predicted bounding box, shows that the proposed model achieves AP = 99.7%, while the EfficientDet-D2 model for comparison of detection performance achieves AP = 69.2% at an intersection over union (IOU) threshold of 0.5. Similarly, the proposed model achieves an AP of 78.9% and the EfficientDet-D2 model achieves an AP of 50.5% for an IOU threshold of 0.75. These results show that accurate prediction is possible within a wide range of recall for glider position inference in a real ocean environment.

Keywords: artificial neural network; CNN vision encoder; underwater glider; low-rank adaptation

Citation: Lee, J.; Park, J.-H.; Hwang, J.-H.; Noh, K.; Choi, Y.; Suh, J. Artificial Neural Network for Glider Detection in a Marine Environment by Improving a CNN Vision Encoder. *J. Mar. Sci. Eng.* **2024**, *12*, 1106. https://doi.org/10.3390/jmse12071106

Academic Editors: Jingchun Zhou, Wenqi Ren, Qiuping Jiang and Yan-Tsung Peng

Received: 11 June 2024
Revised: 27 June 2024
Accepted: 28 June 2024
Published: 29 June 2024

1. Introduction

A rich source of natural resources, the oceans cover a significant portion of the Earth's surface. With advances in economics and technology, ocean exploration has gained widespread attention. Remotely operated vehicles (ROVs) and underwater gliders play a critical role in ocean data collection. ROVs can operate in deep-sea environments that are difficult for humans to access and use a variety of sensors and tools to collect data. Underwater gliders are among the most effective unmanned underwater platforms for ocean observation. These gliders are designed to operate by changing the buoyancy of the surrounding water. They have advantages such as low energy consumption, long navigation range, and extended endurance. Meanwhile, gliders will explore the oceans autonomously and transmit data continuously for long periods of time.

By switching between a negative and positive lift, the gliders follow a slow sawtooth up-and-down trajectory using very little energy. This allows them to travel farther and stay open longer. As a result, they can conduct ocean sampling missions lasting hours to weeks

or months, over thousands of kilometers. At the end of long missions, the glider must be recovered because it is drifting at sea and has no surface propulsion for its own return to the mothership.

The conventional method of lifting the glider onto a small boat relies on human skill and basic tools such as nets, cranes, belts, and hooks. First, the mother ship navigates close to the floating position of the glider. Workers then approach the glider in a small boat. They drop a net or hook to secure the glider and then pull it on board. The boat with the workers and the glider on board is then hoisted back to the mother ship by a crane. The experience and skill of the workers are critical to the success of this recovery operation. Adverse weather conditions can frequently complicate the operation, slowing it down and increasing the risk of injury to the workers. These traditional methods remain largely manual, despite their risks and limitations. There is a need for a safer, more efficient, and automated system of recovery.

In this work, we propose an artificial neural network to detect gliders in a maritime environment and recover them using an ROV platform, which can operate in rough seas and is capable of both human-operated and semi-autonomous control. The ROV platform uses GPS tracking to navigate from a distance to the visible location of the glider, guided by the camera sensors. To navigate long distances, the ROV relies on GPS to get within a few meters of the glider. For short-range navigation, the ROV uses a camera sensor to detect the glider within its visible range and then moves to within a meter of the glider. To facilitate this process, an artificial neural network detects the glider in the camera images, provides position coordinates, and helps control the ROV's movements based on this information.

The main contributions of this work are summarized as follows:

- We propose a lightweight, end-to-end artificial neural network model for detecting gliders with fast inference speed. To compress the weight of the model, a low-rank approximation is applied to the CNN vision encoder.
- A CNN vision encoder that divides an image of a glider at sea into several elongated variable-width patches and extracts feature vectors for each direction. These patches are overlapped to better preserve the spatial information of the pixels in the horizontal and vertical directions.

2. Related Works

Deep learning models use various image segmentation methods to perform image processing tasks. The feature pyramid network (FPN) [1] is a method that effectively uses the features extracted from different resolutions of an image to perform image segmentation tasks. To achieve accurate image segmentation results, the FPN effectively utilizes important information from multiple resolutions of the image. The FPN extracts feature maps from the input image at different resolutions, and the high-resolution feature map information is fed into low-resolution feature maps to generate feature maps with rich information. The vision transformer (ViT) [2] is a method that uses a transformer-based neural network architecture to segment images. It consists of patching, which divides the input image into smaller patches; position encoding, which adds position information to each patch; a transformer encoder, which processes the sequence of patches to learn the image context; and a segmentation head, which uses the learned features to predict the category of each pixel. The ViT can better understand the complex structure of images by learning longer-term dependencies than CNN-based models.

Methods for detecting and tracking objects like gliders in images include traditional image processing techniques and deep learning models. Conventional image processing involves object patch matching, which involves extracting a patch containing the slider from an image and then matching it to similar areas in another image to locate it. These algorithms include Tracking, Learning and Detection (TLD) [3], Kernel Correlation Filters (KCFs) [4], and Channel and Spatial Reliability (CSRT) [5]. However, this method has the drawback that it often fails to match due to variations in brightness and color in different

marine environments and changes in the glider's pose. Also, as the image size increases, the complexity of the matching operation increases.

In order to identify and track the glider more efficiently, deep learning-based object detection uses advanced models. These models are typically implemented as convolutional neural networks (CNNs) and detect objects using images. One of the early successful models is the region-based convolutional neural network (R-CNN) [6], which uses a selective search to generate regions of interest (ROIs) and performs classification and location within those ROIs. An improvement on R-CNN, Fast R-CNN [7] shares the feature extraction stage and uses RPNs to efficiently generate ROIs. As a further refinement, Faster R-CNN integrates RPNs into the network, enabling end-to-end training and faster performance. The single shot multibox detector (SSD) [8] uses a single convolutional network to predict bounding boxes and class labels for each object in an image. RetinaNet [9] is a one-stage dense object detector that uses focal loss, a feature pyramid network, and dense anchor boxes to achieve high accuracy and efficiency. CenterNet [10] is also a one-stage object detector that predicts the key points of objects and regresses their sizes and offsets, which allows for fast and accurate object detection without the need to generate bounding boxes. EfficientDet [11] is a family of neural network architectures that combine depth, width, and resolution scaling to outperform other CNN models. You Only Look Once (YOLO) [12] takes a different approach by simultaneously detecting and classifying objects using a single neural network.

3. Methods

3.1. CNN Vision Encoder

The CNN vision encoder proposed in this work N-splits the input image into horizontal and vertical patches of different widths and generates a feature vector by convolving each image block. The convolution layer applies a depth-separable convolution, passes through a multi-layer perception (MLP) that non-linearizes the output of the convolution, and performs global max pooling to compress the weights of the vision encoder model.

A splitting overlapping image patches in deep learning models, such as CNNs and vision transformers, reduces information loss at the patch boundaries, maintains image continuity, and enables richer and more robust feature learning. Overlapping patches help prevent the loss of important information at the patch boundaries and better detect continuous edges and textures that span across patch boundaries more effectively. This approach mitigates the boundary effects and improves the model's spatial awareness, leading to smoother predictions and better generalization, despite the increased computational cost.

In this work, as shown in Figure 1 below, by dividing the image horizontally and vertically so that some of the patches overlap, the spatial characteristics of the pixels are preserved, and the structural information of the image is transferred to the convolutional layer. The nested CNN vision encoder processes each patch of the image and generates feature vectors.

Depthwise separable convolution is an efficient technique that can significantly reduce the size of the model and the computational cost when compared to the standard convolution operations. It applies a 1×1 filter to each input channel independently and extracts the spatial information across the channels, and then combines all the channels using a 1×1 filter. In a normal convolution operation, each input channel is convolved with a 1×1 filter, which results in the application of the same filter to all the channels. In contrast, depthwise separable convolution applies independent filters to each channel and extracts only spatial information across channels. This results in a significant reduction in the number of parameters in the model.

Low-rank approximation (LRA) [13] has emerged as a valuable technique for compressing a MLP with dense layers in neural networks. LRA can dramatically reduce the number of parameters in dense layers, leading to a substantial decrease in the model size. This translates to reduced storage requirements, faster model training, and more efficient model deployment, particularly for resource-constrained environments such as mobile

devices or embedded systems. By reducing the number of parameters, LRA also leads to a significant reduction in computational complexity. This translates to faster model inference speeds, enabling real-time applications and reducing the overall computational burden. Despite parameter reduction, LRA can often maintain or even slightly improve model accuracy. This is attributed to the ability of LRA to identify and retain the most important features in the data while discarding redundant or less significant information.

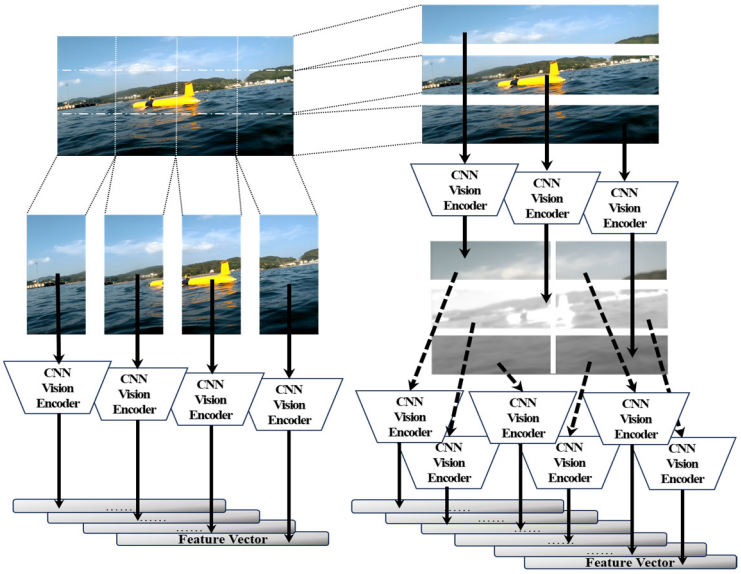

Figure 1. The process of dividing an image into horizontal and vertical directions in a CNN vision encoder.

LRA can be applied to a wide range of dense layer architectures, including fully connected layers in feedforward neural networks, recurrent layers in long short-term memory (LSTM) networks, and fully connected layers in convolutional neural networks (CNNs). Xception [14] is a deep CNN architecture that proposes efficient low-rank approximation techniques for deep CNNs. It introduces depth-wise separable convolutions and a novel depth multiplier mechanism to control the depth and width of the network. Xception also employs low-rank approximation to compress the weight matrices, enabling efficient training and inference of deep CNNs. MobileNet [15] is a lightweight CNN architecture designed for efficient operation on mobile and embedded devices. It utilizes depth-wise separable convolutions and low-rank approximation to significantly reduce the number of parameters and computational cost without sacrificing accuracy.

Global max-pooling, a technique commonly applied in convolutional neural networks (CNNs), offers several advantages that enhance the model performance. Global max-pooling selects the most prominent activations from the input feature map to generate its output. This enables the model to better capture and learn the dominant features within an image. Particularly in object recognition tasks, global max-pooling can improve the model accuracy. Unlike max-pooling, global max-pooling retains spatial information by considering all pixels within the feature map. This helps preserve the structural information of the image, making it valuable for tasks like object detection. In a previous study, Network in Network (NIN) [16] employed global max-pooling for image classification. SqueezeNet [17] is an ultra-lightweight CNN architecture that achieves the accuracy of AlexNet [18] on ImageNet with fewer parameters, making efficient use of convolutional layers and global max-pooling layers.

The diagram of the proposed CNN vision encoder is shown in Figure 2. An image patch of variable width, divided horizontally and vertically, is generated as a feature vector in latent space by depthwise separable convolution, which significantly reduces computational parameters with a separate filter for each channel and a 1×1 convolution

kernel for all channels. The global max-pooling layer compresses the vision encoder output after passing through the dense layer block by low-rank adaptation.

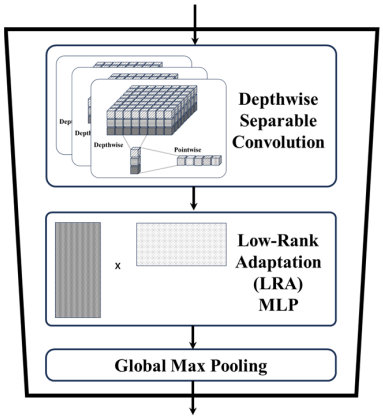

Figure 2. A diagram of the proposed CNN vision encoder.

3.2. Artificial Neural Network Model for Glider Detection

In the proposed artificial neural network model for glider detection, as shown in Figure 3, the image is divided into 2 × 2, 4 × 3, and 9 × 5 with horizontal and vertical, respectively, considering the resolution of the camera mounted on the ROV, and each image patch is converted into a feature vector by a CNN vision encoder.

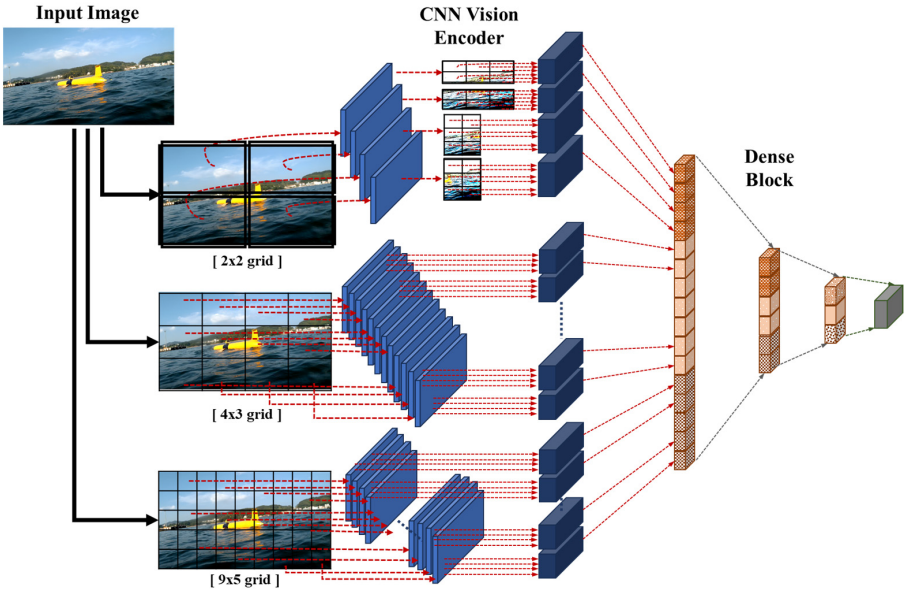

Figure 3. The proposed artificial neural network for glider detection.

Feature vectors in latent space, which are the output of nested CNN vision encoders, are concatenated according to the order of segmented positions, and the output dense layers are mapped to the bounding box and center position to estimate the glider position.

4. Experiments and Results

4.1. Collecting Data in the Marine Environment

For data collection in the marine environment [19], a mothership control system is set up in a dock at Yongho Pier in Busan, South Korea, as shown in Figure 4. The ROV is steered towards the glider to acquire video data from the camera. The sensor information and images are collected by moving the ROV platform closer to the glider as it floats on

the sea surface [20]. Floating gliders have different postures, such as rolling or submerged, which are affected by waves. Using the ROV-mounted PoE camera, the internal camera, and the additional GoPro camera mounted on top of the ROV, the ROV captured video from different approaches. The tracking API of OpenCV is used to mark the position of the float in the recorded image with a bounding box and then automatically track it. The bounding box of the glider object is set wide enough to include the center of the glider for tracking purposes.

Figure 4. Experiment for the collection of image data of gliders at sea.

Automated object bounding box detection methods quickly generate tagging information for large amounts of image data. However, compared to manually setting bounding boxes by humans, the accuracy of the location information is lower, and incorrect detection information is sometimes mixed, which reduces the quality of the dataset. Datasets with an imprecise ground truth are a factor that hinders the improvement of the model's object detection performance, but we used them appropriately to balance cost and resources. The training dataset for the detection of gliders floating on the water surface has generated approximately 2,250,000 samples in a variety of environments.

4.2. Evaluation of the Inference Performance of Artificial Neural Network for Glider Detection

The intersection over union (IOU), a common metric for evaluating detection models, is used to evaluate the inference performance of the deep learning-based glider detection model. The IOU used to evaluate the inference performance of the glider detection model is the ratio of the overlap between two ground truth and predicted bounding boxes. If the value is greater than or equal to the threshold of a given value, it is considered true positive (TP), and if it is less than the threshold, it is considered false positive (FP). TP means that the prediction is actually correct and FP means that the prediction is actually wrong. If it is not possible to detect the ground truth, it is false negative (FN). For general classification problems, the thresholds of IOU are 50%, 75%, and 95%.

However, in the case of glider position tracking detection, glider detection and center position detection are more important than the need to accurately match the shape of the glider object, so the IOU threshold does not need to be high.

Furthermore, since the ground truth bounding box is imprecise when the training dataset is generated, and often only a specific part of the object is tagged instead of the whole object, the IOU threshold to measure the inference performance of the glider detection model is set to 50% and 75% for evaluation.

Precision, also known as positive predictive value (PPV), measures the proportion of true positives, or correctly predicted positive cases, among all positive predictions. It reflects the accuracy of the model's predictions and is an answer to the question of how many of the cases the model predicted would be positive were actually positive. A high PPV is an indication that the model is good at identifying true positives and does not over-predict false positives.

Recall is also known as sensitivity or a hit rate and is a measure of the proportion of true positives among all true positives. It reflects the model's ability to identify positive cases and answers the question of how many true positives the model has correctly identified. A high recall rate is an indication that the model is good at identifying the majority of true positives and that there are no misses.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths} \tag{2}$$

The calculation formulas for precision and recall are described in Equations (1) and (2), and the confusion matrix that explains the relationship between them is shown in Figure 5.

		Predicted Value: Positive	Predicted Value: Negative	
Actual Value	Positive	**TP** (True Positive)	**FN** (False Negative)	**Recall** $\frac{TP}{TP+FN}$
	Negative	**FP** (False Positive)	**TN** (True Negative)	**Specificity** $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FP}$	**Negative Predictive Value** $\frac{TN}{TN+FN}$	**Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$

Figure 5. Confusion matrix.

There is a trade-off between precision and recall. That is, an improvement in one metric often comes at the expense of the other. For example, a model that is very strict in what it predicts will have high precision but low recall. This is because the model will correctly identify most positive cases but miss some. Conversely, a model that is more lenient in its predictions will have high recall but low precision because it will correctly identify most positive cases, but it will also predict many false positives.

The average precision (AP) calculates the prediction results, sorts them in descending order by the confidence score, which indicates the probability that an object is within the prediction bounding box, and creates a precision–recall (P–R) curve, which is the average of the area under the curve. In the COCO metric, the AP is obtained by 101-point interpolation, adding the precision for each 0.01-unit section of the recall and averaging. For this metric, the IOU ranges from 0.5 to 0.95 and is incremented by 0.05. In particular, an AP with an IOU of 0.50 is referred to as the PASCAL VOC metric, and an AP with an IOU of 0.75 is referred to as the strict metric. Equation (3) is the definition of the AP calculation of the COCO metric.

$$AP = \frac{1}{101} \sum_{r \in (0.0, 0.01, \ldots 1)} \underset{r:r \geq r}{max}\, \rho(r) \tag{3}$$

To evaluate the precision and recall of the glider detection, it is necessary to set an IOU threshold t that classifies each detection as a TP or FP. The AP is a metric for the evaluation

of precision and recall at different confidence levels. Thus, it is necessary to count the number of classifications of a TP and FP at different confidence levels. Different precision and recall values can be obtained by choosing a more restrictive threshold value for the AP. When the precision and recall values are calculated using a more restrictive AP threshold, the number of FP detections significantly reduces the recall values.

We compare it with the inference results obtained by EfficientDet. For this comparison, the EfficientDet model was transfer trained using the glider detection dataset. The proposed deep learning glider detection model, SingleDet, uses two modes with a different number of internal layers and weight values, while EfficientDet uses three modes, D0, D1, and D2, depending on the size of the input image and the backbone network. EfficientDet-D0 reportedly achieved 34.3 AP and EfficientDet-D2 achieved 43.5 AP on the COCO dataset.

To verify the inference performance of the model, the test dataset collected from the ocean environment is kept separate from the training dataset. It is not used to train the model, but only to measure the inference performance of the model.

The AP values for the IOU range of the EfficientDet models and the proposed SingleDet models are compared in Table 1 below. In the result, EfficientDet models record an average AP of about 0.67 for an IOU of 0.5 for a test dataset consisting of images collected directly from the ocean environment for glider detection, which decreases exponentially as the IOU increases. In contrast, the SingleDet model records an AP of about 0.78 or higher up to the strict metric with an IOU of 0.75. On average, the SingleDet model performs 50% better than the AP of the EfficientDet model. A graph comparing the AP of different models used in the evaluation by IOU value is shown in Figure 6.

Table 1. Comparison of AP between EfficientDet and SingleDet models in the IOU range.

Model	IOU										mAP
	0.50	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**	
EfficientDet-D0	0.670	0.635	0.593	0.550	0.515	0.454	0.371	0.266	0.112	0.083	0.425
EfficientDet-D1	0.656	0.629	0.594	0.555	0.507	0.437	0.351	0.240	0.094	0.053	0.412
EfficientDet-D2	0.692	0.659	0.628	0.601	0.556	0.505	0.426	0.318	0.134	0.029	0.455
SingleDet-01	0.997	0.993	0.976	0.938	0.879	0.789	0.663	0.517	0.288	0.217	0.726
SingleDet-02	0.989	0.978	0.955	0.916	0.865	0.786	0.655	0.459	0.258	0.092	0.695

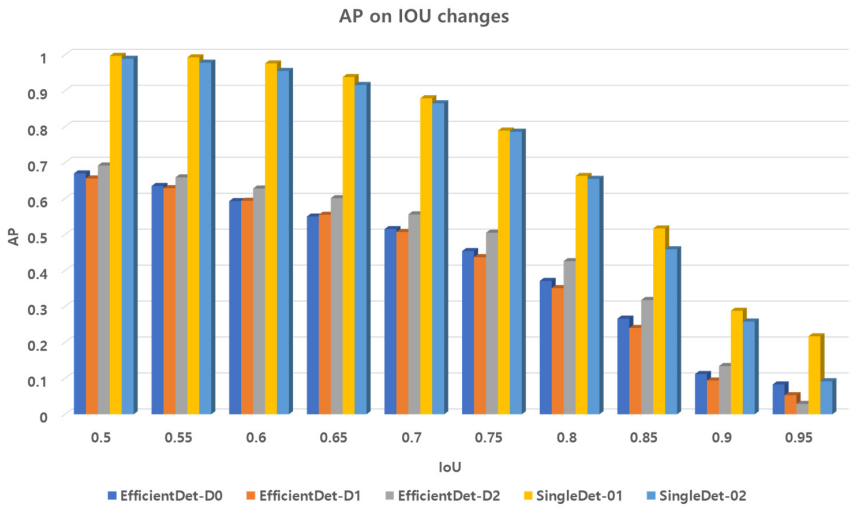

Figure 6. A graphical representation comparing the AP of models by IOU.

The graphical representation of the P–R curve for the inference of the SingleDet glider detection models evaluated for the PASCAL VOC metric at IOU of 0.50 and the strict metric at IOU of 0.75 are as Figure 7. In the P–R curve, the precision of the SingleDet model group tends to be maintained constant even when the recall is increased, which means that the model increases the detection rate while retaining the precision.

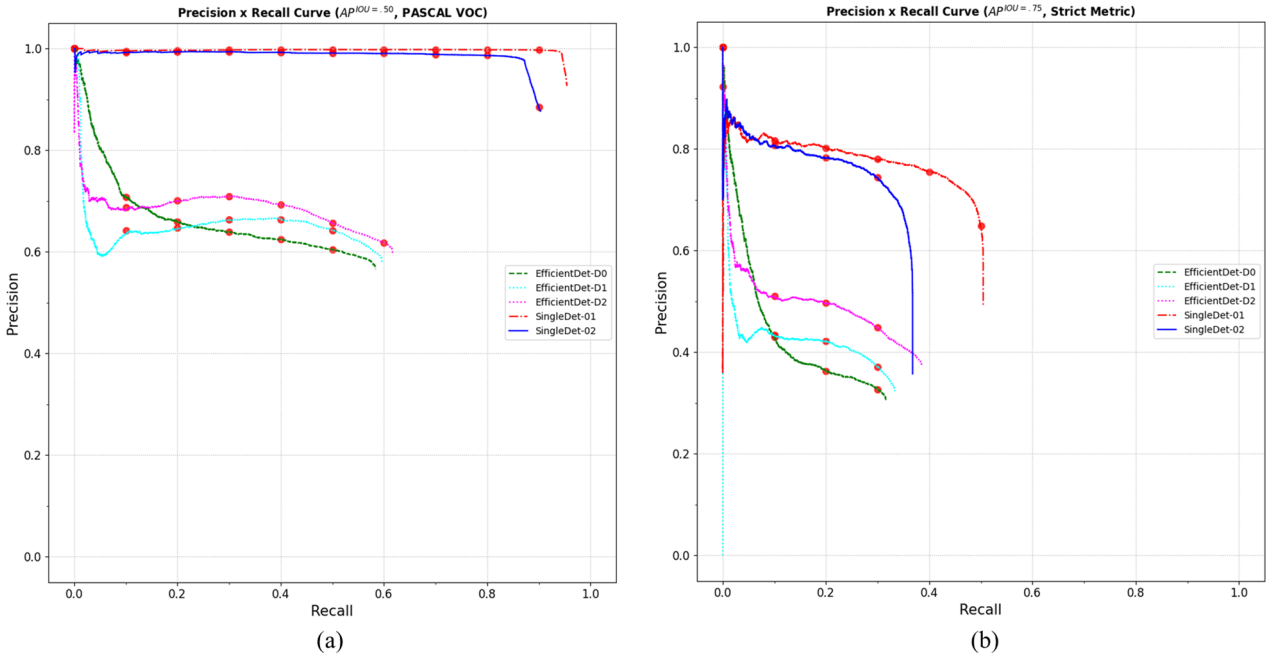

Figure 7. Graph of the precision–recall curve ((**a**) $IOU_{threshold} = 0.50$, (**b**) $IOU_{threshold} = 0.75$).

For the SingleDet model group, using a less restrictive IOU threshold (t = 0.5) leads to higher recalls with the highest precision. That is, the inference of the Singledet-01 model can predict approximately 99.7% of the total ground truth without missing any detections.

The SingleDet-01 model achieves AP = 99.7% while the EfficientDet-D2 model achieves AP = 69.2% at an IOU threshold of t = 0.5. Similarly, for an IOU threshold of t = 0.75, the SingleDet-01 model achieves an AP of 78.9% and the EfficientDet-D2 model achieves an AP of 50.5%. When we use the IOU threshold of t = 0.75, the inference of the glider detection model becomes more sensitive to different confidence levels. This can be explained by the degree to which the curve goes up and down.

A comparison of the inference performance of the SingleDet and EfficientDet glider detection models is shown in Table 2. The computer specifications consist of an Intel i9-9980 CPU and an Nvidia RTX A3000 GPU used to perform the model inference. The results of the comparison show that the SingleDet set of models has more than 1.5 times the average precision of the EfficientDet set and uses about half the number of weight parameters, resulting in a faster inference speed.

Table 2. Comparison of SingleDet and EfficientDet models' AP, parameter size, and prediction time.

Model	Average Prediction (%)			#Parameters		Predict Time
	AP_{50}	AP_{75}	AP_{90}	Params	Ratio	
SingleDet-01	99.7	78.9	28.8	2.5 M	1.0×	177 ms
SingleDet-02	98.9	78.6	25.8	2.9 M	1.2×	314 ms
EfficientDet-D0	67.0	45.4	11.2	3.9 M	1.6×	183 ms
EfficientDet-D1	65.6	43.7	9.4	6.6 M	2.6×	534 ms
EfficientDet-D2	69.2	50.5	13.4	8.1 M	3.3×	732 ms

The results of glider detection using different artificial neural network detection models by operating the ROV platform in the marine environment are shown in Figure 8 below.

Figure 8. Results of the inference of artificial neural network models for the detection of gliders in the marine environment ((**a**) EfficientDet-D0, (**b**) EfficientDet-D1, (**c**) EfficientDet-D2, (**d**) SingleDet-01, (**e**) SingleDet-02).

5. Conclusions

Underwater gliders use long missions to explore the oceans. Unlike other underwater vehicles, they do not have a propulsion system. As a result, recovery of these gliders at the end of their missions is currently a manual process that can be dangerous depending on weather conditions.

In this study, we improve the CNN vision encoder of an artificial neural network model for detecting a glider floating on the water from camera images during the stage of performing glider recovery operations using the ROV platform, when the ROV and the glider are in close proximity.

The proposed artificial neural network glider detection model divides the input image horizontally and vertically into a number of variable-width patches and extracts feature maps of different latent space scales by depthwise separable convolution. The output of the glider detection model is information about the center position and bounding box of the detected object. The proposed SingleDet-01 model showed an average precision of 99.7% when comparing the inference performance of the other detection models on the IOU thresholds set to 0.5. It also had a tendency to maintain precision as recall increased, so the actual detection rate is judged to be high over a wide range.

For future works, this artificial neural network model will be used to assist the ROV platform in glider recovery. The time required for each unit task and the success rate of the operation will be evaluated compared to when a human operator performs the recovery operation with manual control alone. Our goal is to provide feedback on the performance of the work support in the real marine environment and to advance an artificial neural network model to a level where it can be used in practice.

Author Contributions: Conceptualization, J.L. and Y.C.; methodology, J.L. and J.-H.H.; software, J.L., J.-H.H. and K.N.; validation, J.L., Y.C. and J.S.; formal analysis, J.L. and K.N.; investigation, J.L. and J.-H.P.; resources, J.L.; data curation, J.L.; writing—original draft preparation, J.L. and J.-H.H.; writing—review and editing, J.L., Y.C. and J.S.; visualization, J.L., J.-H.H. and K.N.; supervision, J.S.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Korea Institute of Marine Science & Technology Promotion (KIMST), funded by the Ministry of Ocean and Fisheries (2020048213, Development of the core technology and establishment of the operation center for underwater gliders).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: All authors have read and agreed to the published version of the manuscript. The authors would like to thank all researchers of the smart mobility research center in the Korea Institute of Robotics and Technology Convergence, and all reviewers for very helpful comments and suggestions.

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

References

1. Lin, T.-Y.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
2. Dosovitskiy, A. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
3. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422. [CrossRef] [PubMed]
4. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *arXiv* **2014**, arXiv:1404.7584. [CrossRef] [PubMed]
5. Lukežič, A.; Vojíř, T.; Čehovin, L.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. *arXiv* **2016**, arXiv:1611.08461. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef] [PubMed]
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9905, pp. 21–37. [CrossRef]
9. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
10. 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]
11. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *arXiv* **2019**, arXiv:1911.09070.
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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
13. Iyer, C.; Gittens, A.; Carothers, C.; Drineas, P. Iterative Randomized Algorithms for Low Rank Approximation of Tera-scale Matrices with Small Spectral Gaps. In Proceedings of the IEEE/ACM 9th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems (scalA), Dallas, TX, USA, 12 November 2018; pp. 33–40. [CrossRef]
14. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [CrossRef]
15. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNet: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
16. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
17. Iandola, F.N.; Han, S.; Dally, M.W.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
18. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
19. Huynh, T.; Tran, M.-T.; Lee, M.; Kim, Y.-B.; Lee, J.; Suh, J.-H. Development of Recovery System for Underwater Glider. *J. Mar. Sci. Eng.* **2022**, *10*, 1448. [CrossRef]
20. Kumar, R.P.; Dasgupta, A.; Kumar, C.S. Robust Tracking Control of Underwater Vehicles using Time-Delay Control in Discrete-Time Domain. In Proceedings of the OCEANS 2006—Asia Pacific, Singapore, 16–19 May 2006; pp. 1–5. [CrossRef]