sensors

MDPI

Article

Class-Aware Fish Species Recognition Using Deep Learning for an Imbalanced Dataset

Simegnew Yihunie Alaba [1], M M Nabi [1], Chiranjibi Shah [2], Jack Prior [2], Matthew D. Campbell [3], Farron Wallace [4], John E. Ball [1,*] and Robert Moorhead [2]

[1] Department of Electrical and Computer Engineering, James Worth Bagley College of Engineering, Mississippi State University, Starkville, MS 39762, USA
[2] Northern Gulf Institute, Mississippi State University, Starkville, MS 39759, USA
[3] NOAA—National Marine Fisheries Service, Southeast Fisheries Science Center, 3209 Frederic Street, Pascagoula, MS 39567, USA
[4] NOAA Fisheries, 4700 Avenue U, Galveston, TX 77551, USA
* Correspondence: jeball@ece.msstate.edu

Abstract: Fish species recognition is crucial to identifying the abundance of fish species in a specific area, controlling production management, and monitoring the ecosystem, especially identifying the endangered species, which makes accurate fish species recognition essential. In this work, the fish species recognition problem is formulated as an object detection model to handle multiple fish in a single image, which is challenging to classify using a simple classification network. The proposed model consists of MobileNetv3-large and VGG16 backbone networks and an SSD detection head. Moreover, a class-aware loss function is proposed to solve the class imbalance problem of our dataset. The class-aware loss takes the number of instances in each species into account and gives more weight to those species with a smaller number of instances. This loss function can be applied to any classification or object detection task with an imbalanced dataset. The experimental result on the large-scale reef fish dataset, SEAMAPD21, shows that the class-aware loss improves the model over the original loss by up to 79.7%. The experimental result on the Pascal VOC dataset also shows the model outperforms the original SSD object detection model.

Keywords: class-aware loss; deep learning; fish recognition; imbalanced data; object detection; species classification

check for updates

Citation: Alaba, S.Y.; Nabi, M.M.; Shah, C.; Prior, J.; Campbell, M.D.; Wallace, F.; Ball, J.E.; Moorhead, R. Class-Aware Fish Species Recognition Using Deep Learning for an Imbalanced Dataset. *Sensors* **2022**, *22*, 8268. https://doi.org/10.3390/s22218268

Academic Editor: Jiayi Ma

Received: 20 September 2022
Accepted: 26 October 2022
Published: 28 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Fish species classification is an essential component of fisheries management and environmental monitoring. Accurate, reliable, and efficient species recognition of fish is necessary to identify the endangered species, determine the optimal harvest size or time, monitor ecosystems, and develop an intelligent production management system [1,2]. Because of the legal constraints on fishing techniques, precise fish species recognition is essential, especially when their survival is endangered or threatened. Most fisheries use the traditional form of species identification, which demands intensive human labor, consumes time, and can affect the fish's normal behavior. The conventional approaches challenge fishery observers to maintain high-quality data for managing sustainability in the fishery industry, monitoring federal fisheries, assessing fish populations, and identifying different fish species. However, robust deep learning (DL) based fish species identification models would decrease cost and time and improve identification precision.

A machine vision solution can be implemented to replace the manual system. For fish detection, several approaches have been developed, including lidar [3,4], sonar [5], and RGB imaging [6]. RGB imagery is the preferred option to identify fish based on texture, color, and geometry due to ease of operation, cheaper cost, a lightweight system, and no destruction of fish habitat. In the computer vision community, different camera

systems have been implemented to monitor the abundance of fish to aid in assessing the stock and sustainability of marine ecology [7–10]. Different DL techniques, such as object detection and classification, can be implemented to gather information regarding marine life [11,12]. However, the underwater environment has low light conditions, noise, and low-resolution images and videos, making it difficult to distinguish the fish from the background. Additionally, the movement of fish in the water results in images of fish in different shapes and introduces occlusion issues. These issues are challenging for underwater fish species identification and detection.

DL has significantly grown in computer vision applications to solve detection, localization, estimation, and classification problems [13–16]. However, the role is limited in marine ecology and agriculture [17,18] related applications. Different machine learning (ML) and DL methods have been proposed for fish species classification. Huang et al. [19] proposed the hierarchical features of fish and support vector machine (SVM) for fish classification on the fish4knowledge [20] dataset. Jager et al. [21] used AlexNet architecture for feature extraction and multiclass SVM for classification of the LifeCLEF 2015 [22] fish dataset. Zhuang et al. [23] utilized pre-processing and post-processing on modern DL-based models on LifeCLEF 2015 fish data. Most existing DL methods use a simple classification dataset with one fish in each image, which is not always the case in a real environment. In the real system, multiple fish in a single image results in the difficulty of applying a simple classification network. Our dataset, the Southeast Area Monitoring and Assessment Program Dataset (SEAMAPD21) [7], contains multiple fish in a single image, representing a natural habitat. We formulate the fish species identification task as an object detection problem to solve the above problems. Two feature extraction networks, MobileNetv3-large [24] and VGG16 [25], are used as feature extraction networks. The mobileNetv3-large network has fewer parameters than the VGG16 network but has lower mean average precision (mAP), which shows the performance and network speed trade-off. The single-shot multibox detector (SSD) detection head is also used as a regression and classification network. The network generates the class confidence of each species and the location in the image (bounding box information).

Contributions

We formulate the fish species recognition problem as an object detection problem to handle multiple fish recognition in a single image in real time. Additionally, a new loss function is proposed to handle the class imbalance problem to avoid the model's bias toward the dominant class. The proposed model architecture is shown in Figure 1. Our main contributions can be summarized as follows.

1. We formulate the classification problem as an object detection problem to handle multi-class classification and multiple fish in a single image. The model not only classifies the fish species but also localizes each fish in the image, which can work in videos as well.
2. The designed class-balanced term helps to build a class-aware loss function to handle the class imbalance problem. The significance of the loss function is shown in the experiment in Section 5.
3. The model was trained with different backbone networks to show the trade-off between performance and speed between networks. If our need is performance only, we can pick the model with high performance. However, we can choose the lightweight model if we aim to design a fast model.
4. The model was trained on the SEAMAPD21 dataset. The experimental results show the model's promising performance on the challenging dataset.

The rest of the paper is organized as follows. Section 2 presents related work. The use of DL in the fishery industry is summarized in Section 3. Section 4 summarizes the proposed model architecture and sub-networks, such as feature extraction and detection head, dataset augmentation, and loss functions. Results and analyses of the experiment are presented in Section 5. Section 6 concludes the work.

2. Related Work

Fish classification is an extensively studied problem in image segmentation, pattern recognition, and information retrieval. Fish classification uses the resemblance with the representative specimen image to identify and categorize the target fish into species [26]. Some image processing-based techniques for fish recognition have been developed. A balance-guaranteed optimized tree [19] algorithm was developed to reduce the accumulation error in the detection process. Color and shape descriptors were employed to distinguish fish in RGB imagery [27,28]. In past studies, the features used to identify fish primarily relied on hand-crafted feature-generation techniques. If the essential features are missed, the accuracy of fish detection drops drastically. Furthermore, these shallow learning methods do not scale with data. Because of the deep layer structure and massive data support, the performance of DL approaches is higher than the shallow learning methods [29–31].

Instead of the manually created features generally used in classical ML techniques, DL-based vision algorithms distinguish objects by implicitly extracting their distinctive features. Nery et al. [32] presented a fish classification methodology based on a robust feature selection technique. The authors proposed a general set of features and their corresponding weights that the classifier used as prior information to analyze their impacts on the whole classification task. Villon et al. [6] compared DL and histogram of oriented gradients (HOG) and support vector machine (SVM) methods for coral reef fish detection and recognition in underwater videos. For the first method, DL extracts features for detection and classification. For the second method, features are extracted using HOG and fed into SVM for classification. Some studies, such as Li et al. [33], considered fish species classification as object detection. The authors presented fish detection and twelve species recognition of underwater images using Faster-RCNN [34]. However, the model is slow because Faster-RCNN is a two-stage object detection network. Two-stage object detection networks perform better than single-stage networks [35] such as you only look once (YOLO) [36] and SSD [37]. However, they are slower due to extensive computation in the two-stage process region proposal, classification, and regression networks [35,38].

In a real-world application, an unbalanced class distribution in a given dataset, sometimes called a long-tailed data distribution, where a few classes account for most of the data while most classes are under-represented, is common. Models that are trained on such datasets are biased toward the dominant classes. Most existing solutions adopt class rebalancing strategies such as resampling and reweighting based on the number of observations for each class. Sampling techniques, such as synthetic minority over-sampling technique (SMOTE) [39] samples by interpolating from neighboring samples or synthesizing for minor classes [40,41]. These solutions may not work for all datasets with class imbalance problems and may cause models to overfit. Cui et al. [42] proposed a class-balanced loss to solve the class imbalance problem. The authors trained the model on the CIFAR-10 and CIFAR-100 datasets [43] and showed a performance improvement over the original implementation. Li et al. [44] also proposed a balanced group softmax to solve long-tail distribution problems through group-wise training in object detection. This technique divides classes into disjoint groups, and the softmax operation is performed separately on each class. Hence, only classes with a similar number of training samples compete within each group. In this work, we propose class-aware loss to solve the class imbalance problem based on the inverse of the number of samples in each class. We also extend the class-balanced loss into the object detection problem to reweight the classification loss and the localization loss (the class-balanced loss was proposed for the classification problem). Therefore, reweighting the loss based on the inverse number of samples in each class minimizes the effect of model biases toward the dominant class.

3. Deep Learning for Fishery: Motivation

Accurate fish species recognition can provide data for identifying the abundance of fish species in a specific area, assisting with data used for controlling production management and ecosystem monitoring, especially identifying endangered species. Building a

deep learning-based detection system helps maintain high-quality data for sustainable management in the fish industry. This technology reduces time, human labor, and cost to collect and analyze data. Additionally, the deep learning system does not affect the normal behavior of the fish in the habitat during data collection or monitoring. The traditional form of fishery management, including fish species identification, is labor intensive, expensive, and affects the regular activity of the fish in the habitat. Correct data for each species help determine the optimal harvest size and fishing time based on the species distribution in the area. Overall, it is essential to monitor the ecosystem.

Fish species can be distinguished by their shape, color, and size. However, similarity in shape and pattern of some species, relatively low resolution and contrast, and the change in light intensity and fish motion makes accurate fish species identification challenging [45,46]. These challenges complicate solving the problem using classical ML classification techniques (using hand-extracted features). DL methods can learn the unique features without hand-engineered features. Another problem in fish species identification is the number of fish species in each image. In most datasets, each image comprises a single fish, making the classification problem convenient, but finding a single fish in an image with multiple fish is not easy. The dataset of the SEAMAPDP21 [7] consists of many fish species in a single image, making it difficult to use a simple classification network. Therefore, we formulated the fish species identification problem as an object detection problem.

We design an object detection network for fish species recognition based on the SSD [37] object detection network. SSD, which consists of VGG16 [25] as a backbone and a detection head, is designed to do general object detection. In object detection, accuracy and speed are essential. Therefore, the VGG16 [25] backbone network is replaced by MobileNetv3-large [24] to make the model lightweight. The modified model consists of mobileNetv3 as a backbone network and SSD detection head as a classification and regression network, as shown in Figure 1.

4. The Proposed Method and Implementation Details

The proposed model consists of a feature extraction network and a detection head, as shown in Figure 1.

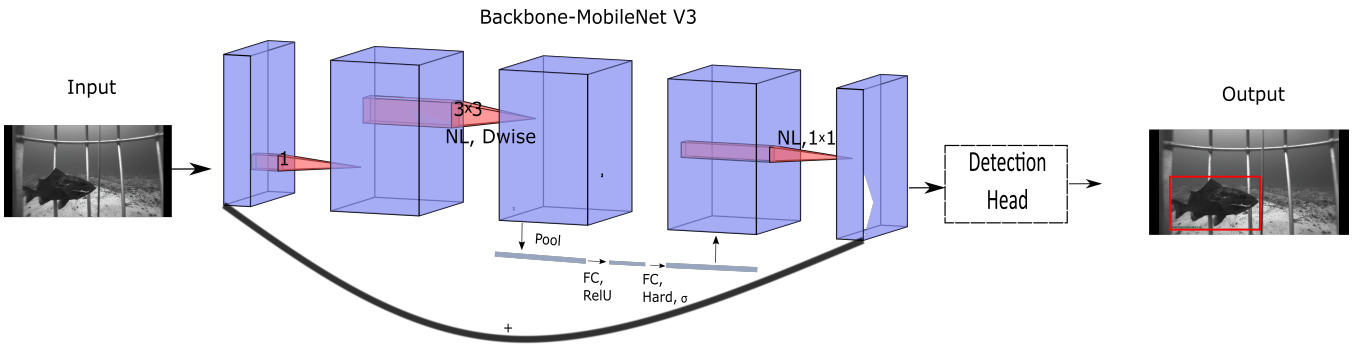


Figure 1. Proposed architecture. The architecture comprises the MobileNetv3-large [24] backbone and SSD [37] detection head. The MobileNetv3 feature extraction network is a lightweight model with good feature extraction capability. The extracted high-level features are input to the SSD detection head for classification and regression tasks. The network outputs fish species type and bounding box information for each image. The model is also trained with the VGG16 [25] backbone network.

The MobileNetv3-large [24] is used as a feature extraction network to extract important features from images and feed them into the detection head for species classification and regression. Once the features are extracted, the SSD [37] detection head uses the high-level feature to predict species classes and regresses bounding boxes for each image. Four boxes with different aspect ratios and scales for each location are predicted, as shown in Figure 2. Each default box has a predicted confidence score and localization offsets for all object categories.

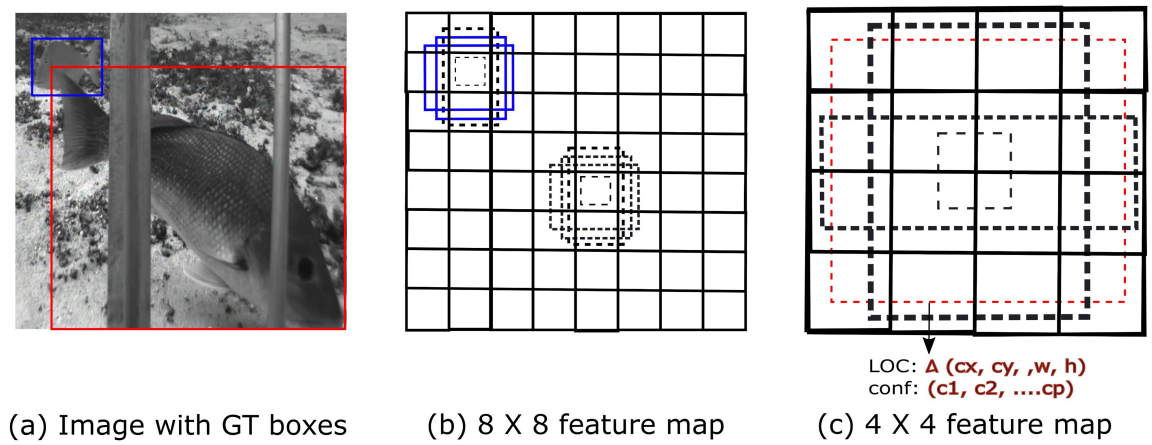


Figure 2. Overview of SSD detection [37]. (**a**) SSD takes an input image and ground truth boxes for each object during training. In a convolutional fashion, at each location, different small sets (e.g., 4) of default boxes of different aspect ratios in several feature maps with different scales (e.g., 8×8 and 4×4 in (**b**,**c**)) are evaluated. For each default box, the shape offsets and the confidences for all object categories (($c_1, c_2, \ldots, c_p$)) are predicted; c_x, c_y, w, and h refer to the x center, y center, width, and height of the bounding box, respectively.

4.1. Feature Extraction Network

Due to the rapid growth of DL, many feature extraction networks, such as VGG [25], ResNet [47], EfficinetNet [48], and MobileNet [49], have been developed. We use MobileNetv3-large, a variety of MobileNet [49], as a feature extraction network because of its lightweight parameter size and good feature extraction capability. The number of parameters of VGG16 [25] is more than 27 times the amount for MobileNetv3-large [24]. The MobileNetv3 network searches the global network structures using platform-aware network architecture search (NAS) by optimizing the network block-wise. Next, the NetAdapt algorithm [50] is used to search the number of filters per layer. This method is essential to fine-tune each layer sequentially, complementing platform-aware NAS. The NAS and NetAdapt techniques can be combined to find an optimized model for specific hardware. The hard-swish [51] activation function used in the MobileNetv3 network also has different advantages, especially during the deployment phase, such as minimizing a potential numerical precision loss usually caused by the sigmoid function. The network also uses depth-wise separable convolution instead of the standard convolution. The depth-wise separable convolution consists of the depth-wise convolution to filter the input channels and point-wise convolution to combine the output of depth-wise convolution. Overall, the depth-wise separable convolution has less computational cost than the standard convolution. These features of the MobileNetv3 network are essential for a real-time deployment of the MobileNetv3-based models on low-power devices. The model is also trained using VGG16 [25] as a backbone network.

4.2. Detection Head

The SSD [37] detection head is used for species prediction and bounding box regression. The SSD detection head makes four object predictions for each location at different scales and aspect ratios [37,52]. This property is essential for detecting small and large objects, which makes SSD preferable as a detection head [52]. A non-maximal suppression (NMS) is also used to suppress redundant predictions based on an intersection over union (IOU) value. The IOU evaluation metric measures the overlap between the ground truth and prediction regions. A higher IOU value indicates a more correct prediction. In most object detection networks, an IOU of 0.5 is commonly used. We also use an IOU of 0.5 to measure a true positive prediction.

4.3. Dataset and Data Augmentation

The large-scale reef fish SEAMAPD21 [7] dataset used for the experiment includes 130 species and 28,328 images; however, it is an imbalanced dataset with some species having many samples, whereas others have few, as shown in Figure 3.

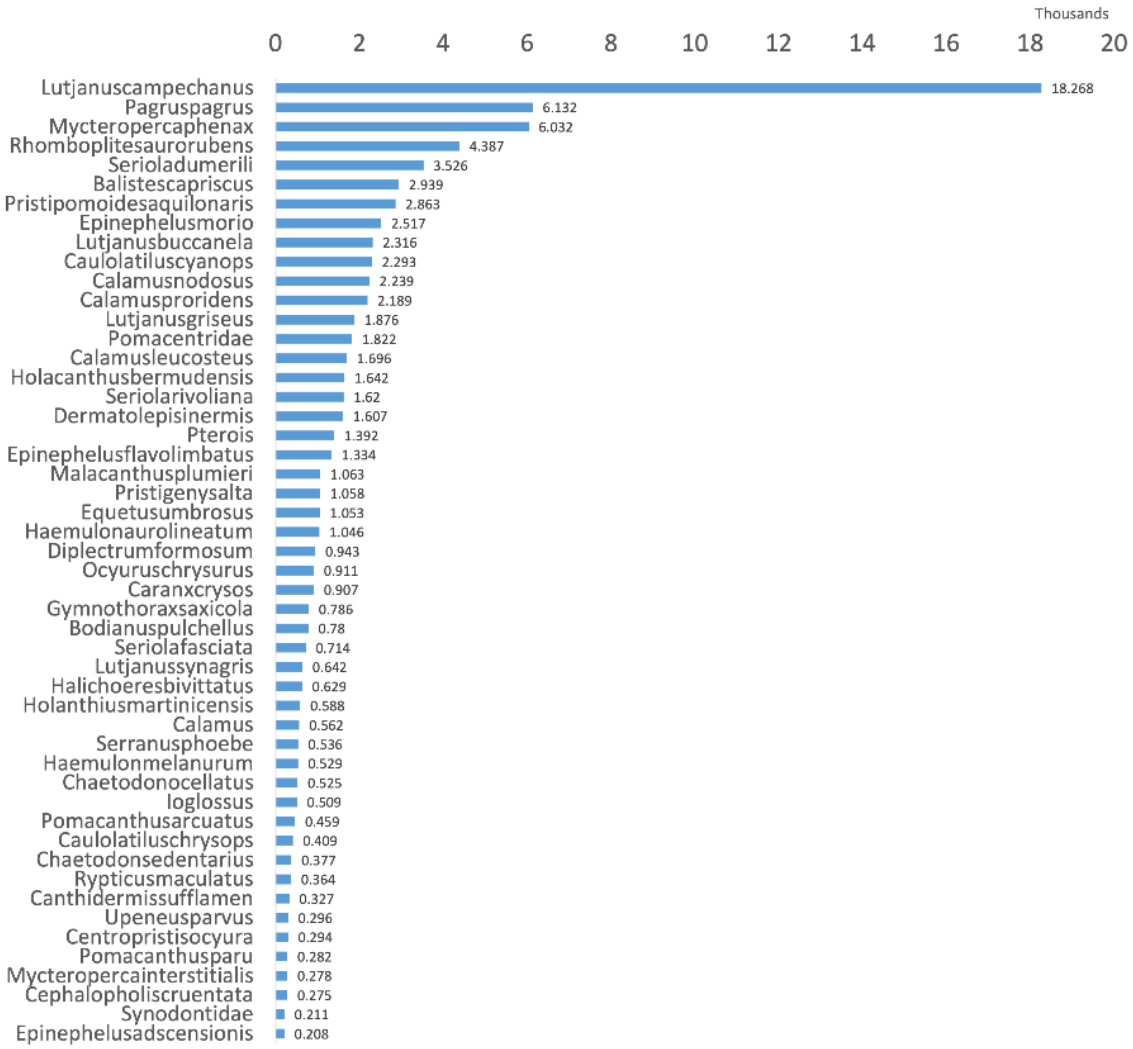


Figure 3. The sample number of occurrences per species distribution in SEAMAPD21 [7] shows a highly imbalanced structure.

The dataset is collected from the Gulf of Mexico continental shelf from Brownsville, TX, USA, to the Dry Tortugas, FL, USA. The target area is also widely diverse in water depth, ranging from 15 m to 200 m [7]. All the sample images are RGB images. When there is a class imbalance, the network prediction is influenced by species with many samples. We modified the cross-entropy loss to handle the class imbalance problem (see Section 4.4). The dataset is fishery-independent (a fishery-independent dataset is a dataset recorded from fish habitat, underwater), so it is challenging to detect fish because of the low resolution and similarity between the background and images (see Ref. [7] for details about the dataset). Sample images from the dataset are shown in Figure 4. Detecting the fish in some images is challenging, even for a human. The dataset is divided into training, validation, and testing in a 70/15/15 ratio.

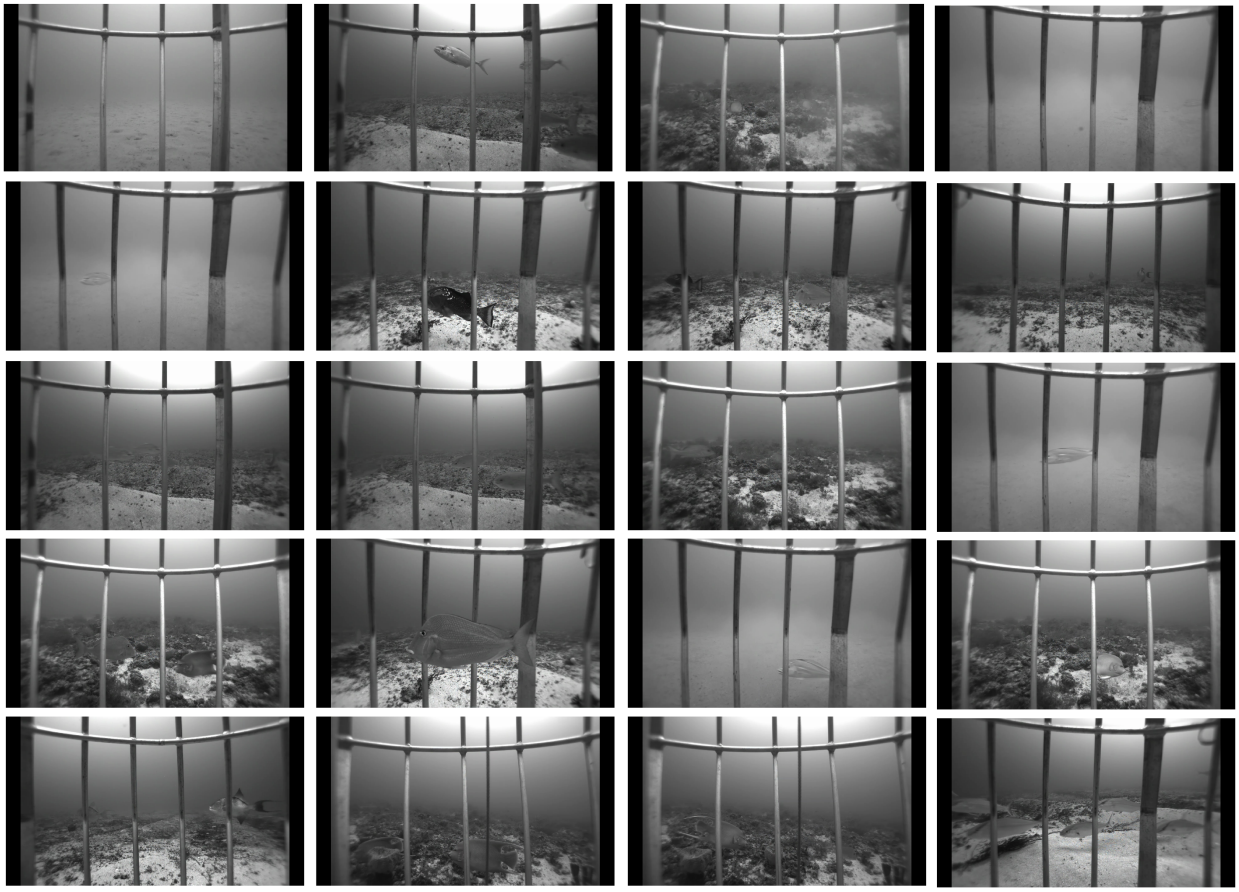


Figure 4. Sample images from the SEAMAPD21 dataset. The images are almost similar to the background, which makes the identification more challenging. Some of the fish images are challenging even for humans to detect. There might be occlusion due to vertical bars or other fish as well.

Data augmentation, which is well-studied in classification problems, is used to increase the dataset slightly using various techniques, such as color transformation, random rotation, rescaling, cropping, zooming, and contrast changes [52]. Different augmentation techniques are available for classification and object detection. We apply random cropping, horizontal flipping with a probability of 0.5, and photometric distortion, including random brightness change, random contrast, and random hue, in the training dataset.

4.4. Loss

The loss function determines the difference between the network-predicted value and the ground truth. If the loss value is large, the predicted value differs from the ground truth, i.e., the prediction is not correct. Therefore, the network trains to learn and updates parameters (weights and biases) to minimize the loss function. Focal loss [53] is a commonly used classification loss for class imbalance, mainly foreground–background class imbalance. However, it does not give better accuracy than cross-entropy in our model because our class imbalance is foreground–foreground class imbalance. Therefore, we use cross-entropy loss as a classification loss and smoothL1 loss [54] as a localization loss. The localization loss, which is similar to SSD [37], can be defined as:

$$
\begin{aligned}
L_{loc(x,p,g)} &= \sum_{i \epsilon pos}^{N} \sum_{m \epsilon (c_x, c_y, w, h)} x_{ij}^{k} SmoothL1(p_i^m - \hat{g}_j^m) \\
\hat{g}_j^{c_x} &= \left(g_j^{c_x} - d_i^{c_x} \right), \\
\hat{g}_j^{c_y} &= \left(g_j^{c_y} - d_i^{c_y} \right), \\
\hat{g}_j^{w} &= \log\left(\frac{g_j^w}{d_i^w} \right), \\
\hat{g}_j^{h} &= \log\left(\frac{g_j^h}{d_i^h} \right),
\end{aligned}
\tag{1}
$$

where N is the number of matched default boxes, d is the default box/anchor, p is the predicted box, g is the ground truth box, w is width, and h is height. The expression x_{ij}^{k} matches the ith default box to the jth ground truth box of category k, and c_x and c_y are centers of the bounding box. Similarly, the classification loss is:

$$
L_{cls} = -\frac{1}{n} \left(\sum_{i=1}^{N} y_i log(\hat{y}_i) \right), \tag{2}
$$

where n is total training samples, y_i is the ground truth, and $\hat{y}_i$ is the predicted output. The total loss is the weighted sum of the localization loss and classification loss.

$$
L(x, c, p, g) = \frac{1}{n} (L_{cls}(x, c) + \alpha L_{loc}(x, p, g)), \tag{3}
$$

where c is class confidence; $\alpha = 1$, which is similar to SSD [37], is used for fair comparison.

We also modified the cross-entropy and smoothL1 loss to handle the class imbalance problem. The proposed loss function considers the number of instances of each species in the training dataset and re-weights both the classification and localization loss to minimize the effect of the dominant class. The proposed class-aware classification and localization losses can be defined as:

$$
\begin{aligned}
L_{clsa} &= \frac{1 - \left(\frac{n_s}{n}\right)}{1 - \left(\frac{n_s}{n}\right)^{\eta}} L_{cls}, \\
L_{loca} &= \frac{1 - \left(\frac{n_s}{n}\right)}{1 - \left(\frac{n_s}{n}\right)^{\eta}} L_{loc},
\end{aligned}
\tag{4}
$$

where L_{clsa} is class-aware classification loss and L_{loca} is class-aware localization loss; n_s is the number of training instances per species, n is the total training samples, and η is a hyper-parameter. We use $\eta = 4$ for this training; $n_s << n$. When n_s is small and the value of η increases, the value of the multiplying term $(\frac{1-\left(\frac{n_s}{n}\right)}{1-\left(\frac{n_s}{n}\right)^{\eta}})$ increases, which maximizes the weight of the less represented class. The hyper-parameter values need to be tuned to get maximum performance. Generally, increasing the η value increases the value of the multiplying term of the loss functions, but it may not increase the overall performance due to its effect on other classes. Therefore, the best value of the hyper-parameter for maximum performance for a given dataset needs to be tuned starting with $\eta = 1$. The class balanced loss proposed by Cui et al. [42] improves the classification problems with class imbalance. Our class-aware loss function can work for classification and object detection with class imbalance datasets. The experimental result on the SEAMAPD21 dataset shows our loss gives a better result than the class balanced loss [42] (see Section 5 for details). The class-balanced (CB) loss for y class with n_y training samples can be written as:

$$
CB_{loss} = \frac{1 - \beta}{1 - \beta^{n_y}} L(x, y), \tag{5}
$$

where $L(x, y)$ can be cross-entropy loss or focal loss, and β is a hyper-parameter. We use $\beta = 2$ similar to the class balanced loss [42] for fair comparison, and this also gives best performance.

The authors considered the class-balanced loss for a classification problem, so they reweighted only the classification loss, such as focal loss. However, for other tasks, such as object detection, the localization loss is also affected by the class imbalance. Therefore, we consider the class-aware and class-balanced loss functions reweighting to both the localization and classification losses. We modified the class balanced loss for object detection as follows:

$$Objective_{loss} = \frac{1}{n}\left(\frac{1-\beta}{1-\beta^{n_y}}L(x,y) + \alpha(\frac{1-\beta}{1-\beta^{n_y}})L_{loc}(x,p,g)\right), \quad (6)$$

where n_y refers to the number of instances per species rather than the number of samples for each class in the class-balanced loss; $\alpha = 1$ is used.

Different regularization techniques have been proposed to optimize network training [55–57]. In [55], a two-stage training method, pretraining and implicit regularization, was proposed to minimize the effect of overfitting. In the pretraining phase, the image representations were extracted. Next, in the implicit regularization stage, the feature boundary is regularized by retraining the network based on the results from the first stage. Training deep learning models, starting with a low learning rate and gradually increasing, becomes the common trend and improves the network's performance [56,57]. In this work, the MultiStepLR scheduler is used to optimize the stochastic gradient descent (SGD) optimizer with an initial warm-up training and an initial learning rate of 10^{-3}. The learning rate gradually decreases with a weight decay of 5×10^{-4}. Additionally, a momentum of 0.9, gamma of 0.1, and batch size of 32 are used. The gamma value shows how much the learning rate drops at each step. All the experiments are done using Tesla v100 GPU running on Centos 7.8 operating system.

5. Results and Discussions

5.1. Quantitative Analysis

The commonly used evaluation metric in object detection is mean average precision (mAP), which we use to measure the model's performance. The network predicts the average precision of each fish species, and the mAP is calculated for the overall class. Tables 1 and 2 show the experimental result of our model on the SEAMAPD21 dataset and Pascal VOC public object detection dataset. The SEAMAPD21 dataset is challenging for the network to learn essential features due to low resolution and background and foreground similarity. Because of these challenges, the mAP of the network in the SEAMAPD21 fish dataset is less than that of the VOC dataset. The proposed class-aware loss function significantly improves the model's performance, as shown in Table 1. When we randomly split the dataset into training, validation, and test data, some species might not have either validation or test data, especially those with fewer than ten images. Due to this issue in the dataset, we report only 82 species for the recognition results in this work. There is an increment of 14.42 mAP when the class-aware loss is used with MobileNetv3-large backbone, which is a 79.70% improvement over the original implementation. The model with a VGG16 backbone outperforms the MobileNetv3-large network even though the total parameters for VGG16 are almost 27 times greater. Increasing the resolution of the input image from 300×300 to 512×512 improves the model's performance, as shown in Tables 1 and 2. Therefore, choosing a smaller or larger backbone is a trade-off between high detection accuracy and low computational power. We also modified the class-balanced loss [42] for object detection (the class-balanced loss was proposed for a classification task), and we trained the model to compare it to our proposed loss function. The model's mAP using the modified class-balanced loss with VGG512 backbone is 51.64%, which is lower than the mAP using the proposed class-aware loss of 52.75%.

Table 1. Experimental result (mAP %) of 82 species on the SEAMAPD21 dataset, where CE stands for cross-entropy loss and CACE stands for class-aware cross-entropy loss; ↑ means increment. VGG300 is the VGG16 backbone with an image size of 300 × 300, and VGG512 is for an image size of 512 × 512.

Backbone	CE Loss mAP	CACE Loss mAP	mAP ↑	Time (FPS)
MobileNetv3-large	18.09	32.51	14.42	105
VGG300	40.00	48.99	8.99	67
VGG512	42.79	52.75	9.96	54

Table 2. The experimental result (mAP %) comparison of the proposed model and the original SSD model on the Pascal VOC dataset. VGG300 is the VGG16 backbone with an image size of 300 × 300, and VGG512 is for an image size of 512 × 512.

Backbone	Original SSD	Proposed Model
VGG300	74.30	78.28
VGG512	76.8	80.61
MobileNetv3-large	-	70.79

Table 2 shows that the model outperforms the original SSD [37] network in 300 × 300 and 512 × 512 resolutions. The Pascal VOC dataset is relatively balanced, so it does not have a class imbalance issue like the SEAMAPD21 dataset. We also trained the model with the MobileNetv3-large backbone, which gave a comparable performance with a smaller number of parameters. When the model is trained with class-aware loss, the AP of many species increases; even those species with less than ten AP got more than 40. The AP of each species is shown in Appendix A Table A1. Some fish species have zero AP, especially in the original cross-entropy loss implementation; however, their performance improves when the proposed loss is used (zero is a rounded value because the value is less than one). The AP of some fish species with more samples decreased when the proposed loss was used. However, the overall AP was improved.

Generally, the number of samples of each fish species, the resolution of each image, the similarity between some fish species, and the similarity of the foreground and background for some fish species contribute to the AP increment or decrement. As a rule of thumb, more samples of each species give better AP than a smaller number of samples if other factors are constant. Higher resolution also improves the detection performance; however, it increases the model parameters. Additionally, the network's performance is reduced when the foreground is similar to the background.

5.2. Qualitative Analysis

We provide the qualitative outputs in Figures 5–7 for VGG300, MobileNetv3, and VGG512 backbones, respectively. Using the VGG16 backbone at different image sizes, 300 × 300 and 512 × 512, gives an excellent qualitative output. The 300 × 300 image size almost gives as good a result as 512 × 512 except for some detections. For example, in row two, in the right image of Figure 5, there are six detections, but there are seven detections for VGG512 in Figure 7, which means one fish is not detected in Figure 5. However, the MobileNetv3 backbone has many missed detections compared to the VGG16 backbone. For example, for rows two and three, in the right images of Figure 6, there are five and one detections, but there are seven and four detections in Figure 7 for the corresponding images.

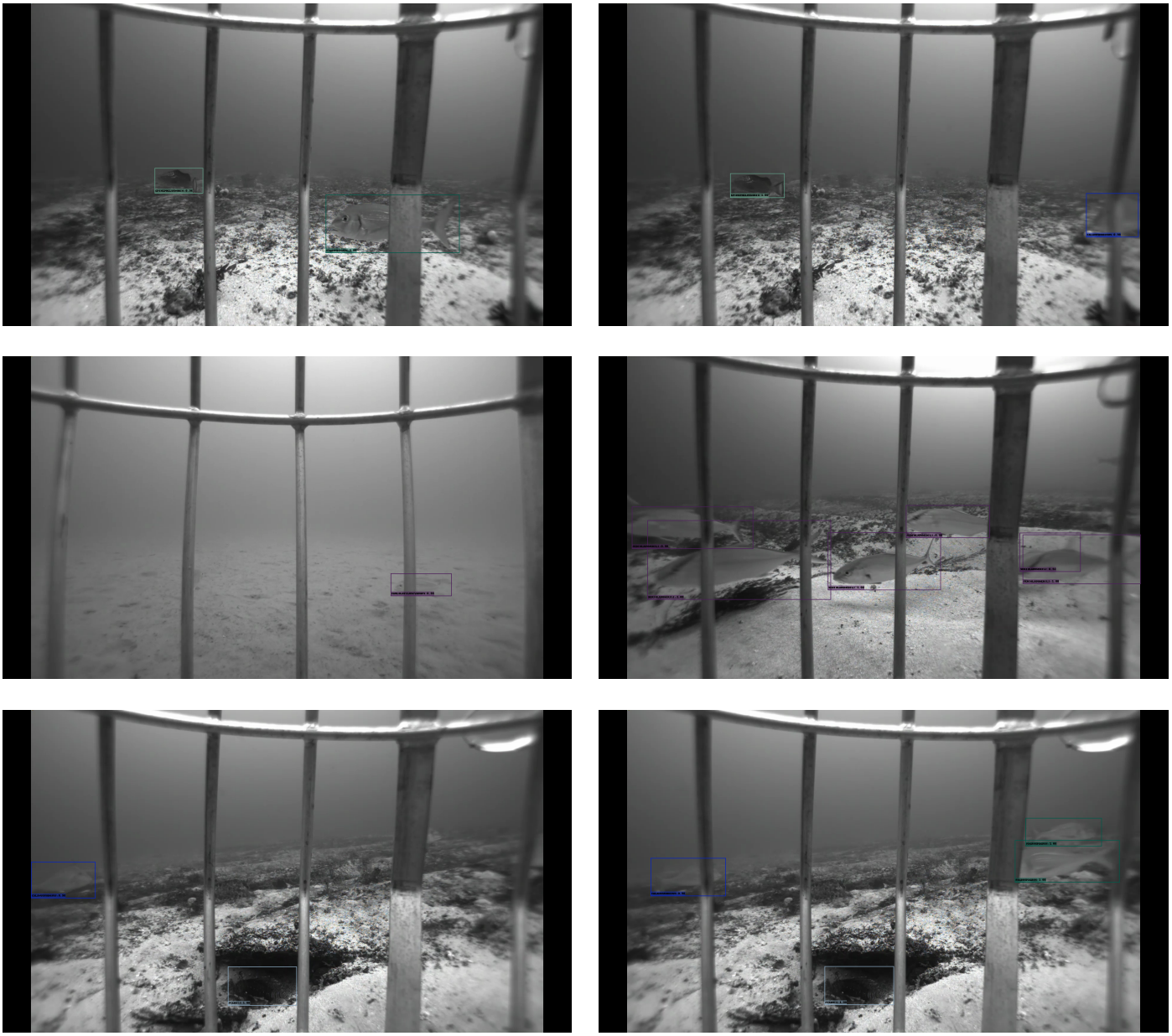


Figure 5. VGG300 backbone qualitative outputs. All fish species are detected in the sample images except the bottom left image. In the bottom left image, one fish is not detected, which is on the middle right side of the image. It is not easy to spot the missed fish, even for humans.

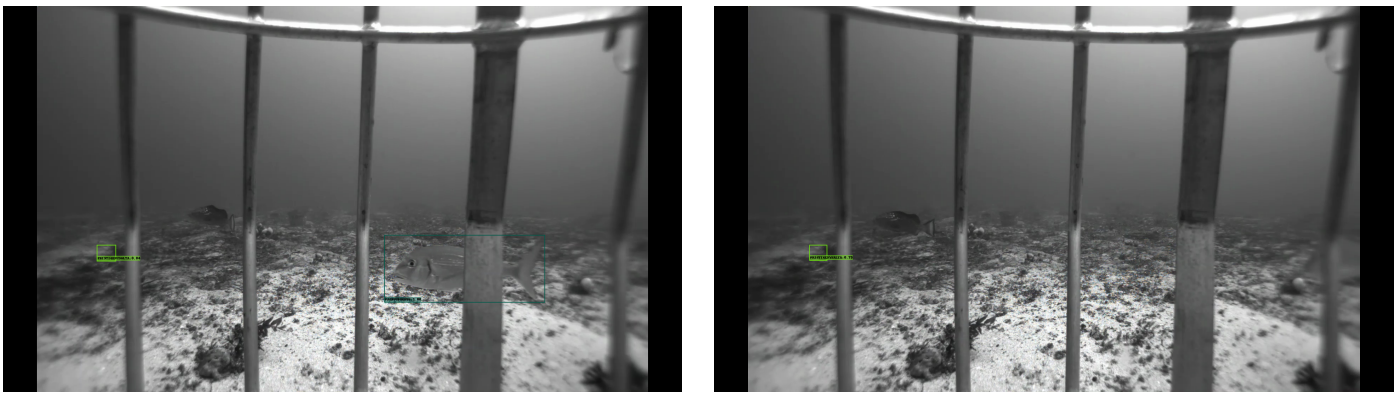


Figure 6. *Cont*.

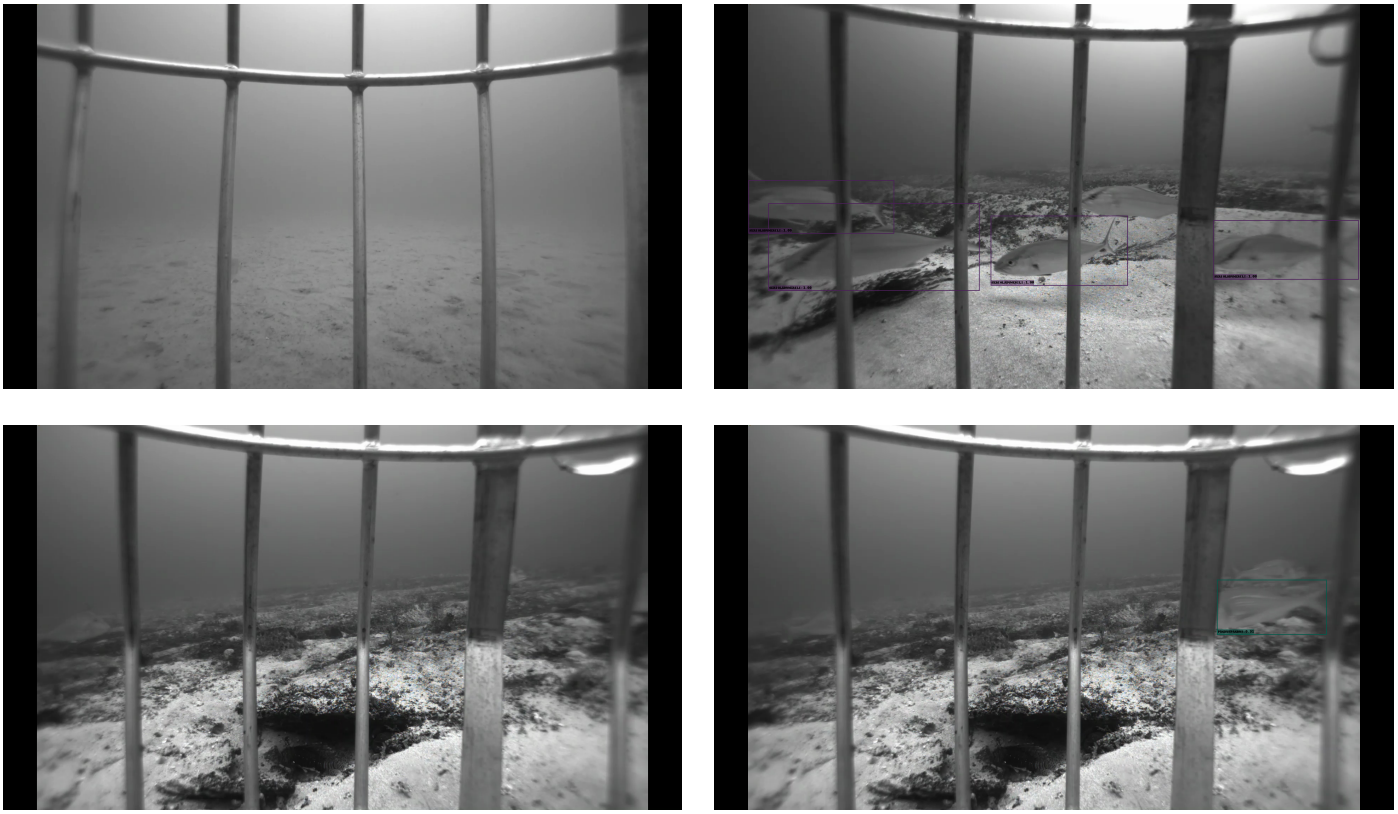


Figure 6. MobileNetv3 backbone qualitative outputs. These sample outputs show the missed detection using the MobileNetv3 backbone, whereas the VGG backbone detects them. One and two fish, respectively, are not detected in the first and second images of the first row. There is the same number of fish missed detection in the second row. However, two and three fish are not detected from the last row of images.

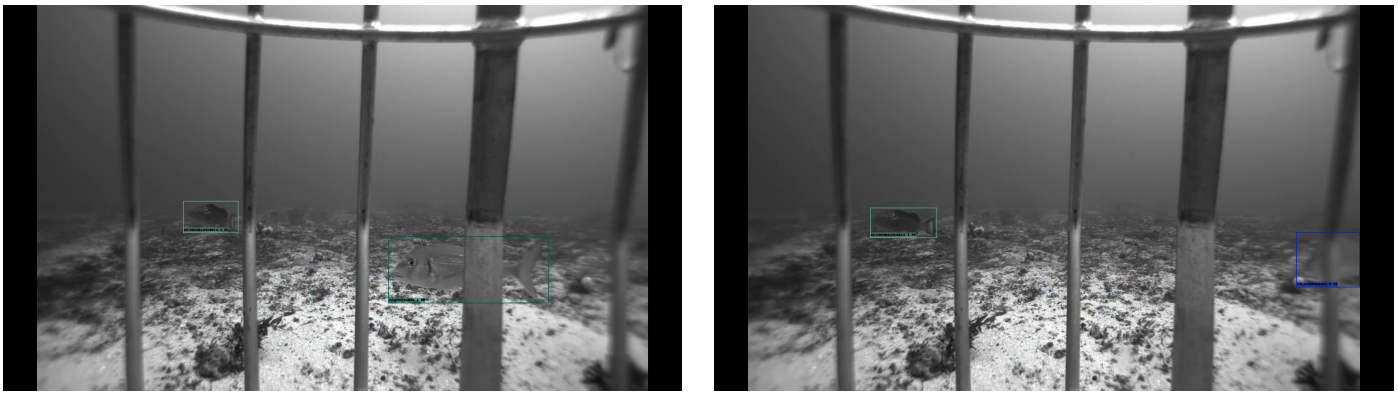


Figure 7. *Cont*.

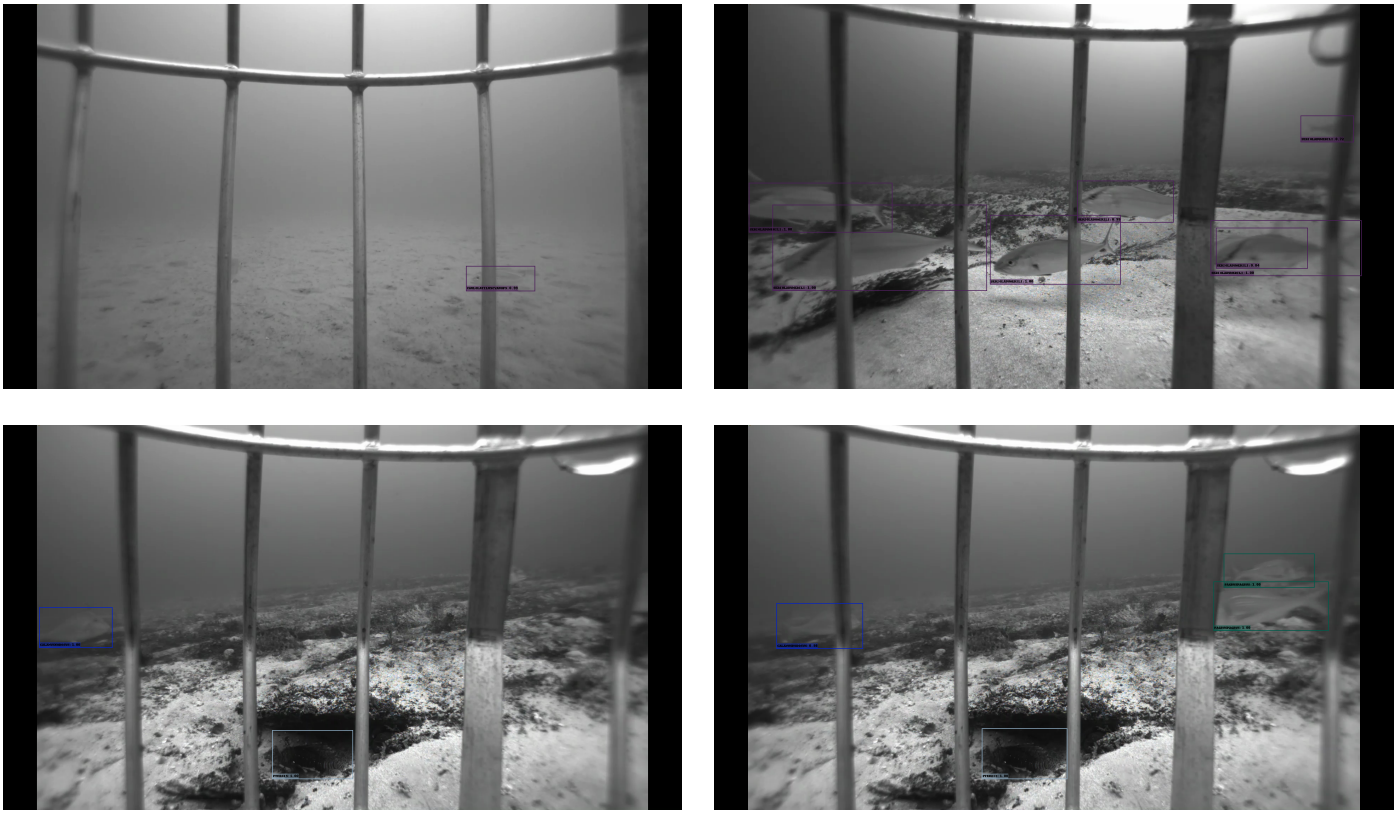


Figure 7. VGG512 backbone qualitative outputs. All fish species in each image are detected with high confidence.

6. Conclusions

In this study, the fish species recognition task is formulated as an object detection task and evaluated on underwater images in the SEAMAPD21 fish species dataset. The MobileNetv3-large and VGG16 backbone networks are used as feature extractors. The experimental result on the SEAMAPD21 dataset shows that a model with large backbone networks can generalize better, but it increases the computational burden of the system. Although the VGG16 backbone network outperforms the MobileNetv3-large backbone network, it is 27 times larger, which increases the computational cost of the overall system, so it may not be suitable for real-time implementation. Therefore, choosing the best backbone network can be a trade-off between performance and processing speed. The SEAMAPD21 dataset is a highly imbalanced dataset, so the dominant class affects the network's detection. The class-aware loss function is proposed to handle the class imbalance. The proposed loss function considers the instance of each species and reweights the loss function at each iteration to minimize the effect of the dominant class. The experimental result shows the effectiveness of the proposed loss function. The proposed loss function can be adapted to any object detection or classification task for an imbalanced dataset.

The existing backbone networks are used for this model. The backbone networks were designed for general-purpose computer vision tasks. To detect small objects, such as fish, and accurately extract features of challenging datasets, especially underwater images, specially designed feature extraction networks are essential. Because of the pooling operation in deep networks, there is information loss at each operation. Information loss affects small objects due to their fewer features at each layer. Due to these issues, there were missed detections in some cases, especially using the MobileNetv3 backbone. Additionally, we did not deploy the model in real-time to test the network's performance. In future work, we will design a suitable backbone network to detect small objects and deploy the model in a low-power device for real-time implementation.

Author Contributions: Conceptualization, S.Y.A.; methodology, S.Y.A.; software, S.Y.A.; validation, J.E.B. and R.M.; formal analysis, S.Y.A.; investigation, S.Y.A.; resources, J.E.B. and R.M.; data curation, J.P., M.D.C. and S.Y.A.; writing—original draft preparation, S.Y.A., M.M.N. and C.S.; writing—review and editing, J.E.B., R.M., J.P., M.D.C. and F.W.; visualization, S.Y.A., C.S. and M.M.N.; supervision, J.E.B., R.M. and F.W.; project administration, J.E.B., R.M. and F.W.; funding acquisition, R.M. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by awards NA16OAR4320199 and NA21OAR4320190 to the Northern Gulf Institute at Mississippi State University from NOAA's Office of Oceanic and Atmospheric Research, U.S. Department of Commerce.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset is available at https://github.com/SEFSC/SEAMAPD21 (accessed on 18 May 2022). The given link is updated every time. Therefore, it may have more datasets than we have used for this experiment.

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

Abbreviations

The following abbreviations are used in this manuscript:

CB	Class Balanced
DL	Deep Learning
HOG	Histogram of Oriented Gradients
IOU	Intersection Over Union
ML	Machine Learning
mAP	Mean Average Precision
NMS	Non-maximal Suppression
RCNN	Region-based Convolutional Neural Network
SEAMAPD	Southeast Area Monitoring and Assessment Program Dataset
SSD	Single Shot MultiBox Detector
SGD	Stochastic Gradient Descen
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
YOLO	You Only Look Once

Appendix A

Table A1. AP of the original cross-entropy and proposed class-aware losses for each fish species. VGG300 is for the VGG16 backbone with a resolution of 300 × 300, and VGG512 is for a resolution of 512 × 512. The MobileNet and VGG are backbone networks, and the detection head is SSD for both cases.

	AP (%) Using Cross-Entropy Loss			**AP (%) Using Class-Aware Loss**		
Species	**MobileNet**	**VGG300**	**VGG512**	**MobileNet**	**VGG300**	**VGG512**
ACANTHURUS	0	54	19.95	20.5	54.55	59.32
ALECTISCILIARIS	0	36.36	37.69	72.73	36.36	29.9
ANOMURA	0	18.18	29.2	0	61.36	72.73
ARCHOSARGUSPROBATOCEPHALUS	0	41.91	60.05	15.02	45.45	54.13
BALISTESCAPRISCUS	61.74	74.08	68.71	67.53	73.42	74.31
BALISTESVETULA	0	6.06	22.46	20.7	37.26	42.92
BODIANUSPULCHELLUS	30.68	53.26	51.26	38.07	55.63	59.21
BODIANUSRUFUS	0	18.18	36.36	0	20.41	41.53
CALAMUSBAJONADO	0	0	87.58	0	78.6	89.6

Table A1. *Cont.*

	AP (%) Using Cross-Entropy Loss			AP (%) Using Class-Aware Loss		
Species	**MobileNet**	**VGG300**	**VGG512**	**MobileNet**	**VGG300**	**VGG512**
CALAMUSLEUCOSTEUS	48.07	60.48	68.47	54.86	69.24	73.16
CALAMUSNODOSUS	12.64	4.28	9.38	7.57	6.89	10.3
CALAMUSPRORIDENS	14.42	17.57	13.02	15.34	18.64	12.75
CALAMUS	25.68	44.81	50.35	26.73	48.4	58.22
CANTHIDERMISSUFFLAMEN	1.5	37.5	24.24	2.6	47.73	48.42
CARANXBARTHOLOMAEI	42.86	40.49	53.41	46.62	50.64	54.35
CARANXCRYSOS	14.74	47.34	33.2	19.13	48.94	45.56
CARANXRUBER	0	54.55	100	0	84.85	54.55
CARCHARHINUSFALCIFORMIS	0	54.55	54.55	100	54.55	54.55
CARCHARHINUSPEREZI	45.45	100	100	100	100	100
CARCHARHINUSPLUMBEUS	0	0	36.36	0	6.06	36.36
CAULOLATILUSCHRYSOPS	39.23	41.65	21.28	41.85	42.43	39.22
CAULOLATILUSCYANOPS	13.55	8.78	6.07	10.88	9.32	10.86
CEPHALOPHOLISCRUENTATA	18.18	64.09	59.52	31.1	70.63	65.14
CHAETODONSEDENTARIUS	0	9.68	19.75	9.09	9.4	16.52
CHROMISENCHRYSURUS	0	0	0	1.23	0	42.08
CHROMISINSOLATUS	0	0	0	100	100	11.76
DERMATOLEPISINERMIS	18.84	0	6.73	17.44	6.43	13.01
DIODONTIDAE	0	0	0	100	100	100
DIPLECTRUMFORMOSUM	18.85	36.3	40.95	27.72	44.46	60.04
EPINEPHELUSADSCENSIONIS	12.55	65.23	50.03	21.44	63.68	75.8
EPINEPHELUSFLAVOLIMBATUS	87.49	90.14	72.13	90.85	90.76	90.41
EPINEPHELUSMORIO	30.42	35.95	25.69	33.69	37.5	35.33
EPINEPHELUSNIGRITUS	96.1	100	90.68	100	100	100
EPINEPHELUS	0	63.64	53.9	36.36	63.64	63.64
EQUETUSLANCEOLATUS	0	100	44.74	100	100	100
EQUETUSUMBROSUS	27.61	70.76	55.33	39.5	73.26	73.64
GONIOPLECTRUSHISPANUS	0	0	70.59	81.82	85.71	84.03
HAEMULONAUROLINEATUM	45.51	61.5	62.24	55.57	68.3	75.68
HAEMULONFLAVOLINEATUM	0	19.17	42.47	0	24.11	67.85
HAEMULONMACROSTOMUM	0	100	100	50	100	100
HAEMULONMELANURUM	45.51	75.69	79.05	53.03	77.29	84.94
HAEMULONPLUMIERI	11.86	40.49	57.93	39.48	47.68	64.31
HALICHOERESBATHYPHILUS	0	26.3	31.64	9.27	26.55	52.36
HOLACANTHUSBERMUDENSIS	5.84	15.53	25.58	11.02	18.65	23.38
HOLOCENTRUS	0	73.27	48.6	0	84.22	93.78
HYPOPLECTRUSUNICOLOR	0	7.77	27.27	0	6.06	33.64
KYPHOSUS	0	24.79	22.73	24.76	18.58	27.91
LACHNOLAIMUSMAXIMUS	0	2.02	9.09	9.8	12.73	9.32
LIOPROPOMAEUKRINES	0	0	0	3.03	20.78	21.56
LUTJANUSANALIS	0	18.18	33.33	41.12	28.02	21.21
LUTJANUSAPODUS	0	54.55	95.16	0	44.44	54.55
LUTJANUSBUCCANELA	62.15	76.25	73.02	62.02	77.09	81.72
LUTJANUSCAMPECHANUS	34.06	46.59	40.51	37.58	49.26	50.18
LUTJANUSGRISEUS	12.71	31.59	46.06	16.3	40.1	54.56
LUTJANUSSYNAGRIS	6.31	3.72	13.2	11.81	13.16	9.49
LUTJANUSVIVANUS	48.45	20.45	56	70.66	36.56	45.45
MULLOIDICHTHYSMARTINICUS	0	36.36	72.73	0	72.73	72.73
MURAENARETIFERA	49	80.44	70.29	70.94	89.05	71.73
MYCTEROPERCABONACI	90.91	86.76	90.91	45.45	90.91	90.91
MYCTEROPERCAINTERSTIALIS	0	63.64	40.91	0	70.91	39.09
MYCTEROPERCAINTERSTITIALIS	54.74	58.71	63.3	68.82	63.42	69.87
MYCTEROPERCAPHENAX	23.26	38.32	31.5	26.26	40.52	41.48
MYCTEROPERCA	0	45.45	45.45	14.55	54.65	45.86

Table A1. *Cont.*

	AP (%) Using Cross-Entropy Loss			AP (%) Using Class-Aware Loss		
Species	**MobileNet**	**VGG300**	**VGG512**	**MobileNet**	**VGG300**	**VGG512**
OCYURUSCHRYSURUS	18.41	37.17	41.56	21.24	42.16	55.55
PAGRUSPAGRUS	21.43	26.4	27.27	21.21	28.55	31.68
PARANTHIASFURCIFER	0	45.45	53.5	54.55	56.21	15.36
POMACANTHUSARCUATUS	16.87	15.82	10.08	18.92	17.43	18.81
POMACANTHUSPARU	25.13	55.48	59.32	27.86	59.82	71.83
POMACANTHUS	0	63.64	18.18	0	0	63.64
POMACENTRIDAE	0	0	9.09	0	10.99	25.77
PRISTIGENYSALTA	2.3	11.91	15.44	0	12.33	12.86
PRISTIPOMOIDESAQUILONARIS	12.19	15.8	18.18	3.58	15.44	18.17
PTEROIS	27.13	20.97	19.62	32.83	23.57	22.9
RACHYCENTRONCANADUM	3.41	27.27	27.27	45.45	0	27.72
RHOMBOPLITESAURORUBENS	38.39	57	55.54	44.73	60.6	67.29
RYPTICUSMACULATUS	23.23	50.95	38.63	50.79	70.94	73.96
SERIOLADUMERILI	49.16	59.14	52.09	51.62	61.19	59.48
SERIOLAFASCIATA	46.85	57.9	42.41	45.62	61.89	54.66
SERIOLARIVOLIANA	48.01	59.23	59.6	51.55	60.93	66.46
SPHYRAENABARRACUDA	0	4 5.45	45.45	0	45.45	45.45
UPENEUSPARVUS	0	28.21	10.95	1.07	17.65	37.36
XANTHICHTHYSRINGENS	0	45.45	52.27	23.21	100	100

References

1. Chang, C.; Fang, W.; Jao, R.C.; Shyu, C.; Liao, I.C. Development of an intelligent feeding controller for indoor intensive culturing of eel. *Aquac. Eng.* **2005**, *32*, 343–353. [CrossRef]
2. Cabreira, A.G.; Tripode, M.; Madirolas, A. Artificial neural networks for fish-species identification. *ICES J. Mar. Sci.* **2009**, *66*, 1119–1129.
3. Churnside, J.H.; Wells, R.; Boswell, K.M.; Quinlan, J.A.; Marchbanks, R.D.; McCarty, B.J.; Sutton, T.T. Surveying the distribution and abundance of flying fishes and other epipelagics in the northern Gulf of Mexico using airborne lidar. *Bull. Mar. Sci.* **2017**, *93*, 591–609. [CrossRef]
4. Jalali, M.A.; Ierodiaconou, D.; Monk, J.; Gorfine, H.; Rattray, A. Predictive mapping of abalone fishing grounds using remotely-sensed LiDAR and commercial catch data. *Fish. Res.* **2015**, *169*, 26–36. [CrossRef]
5. Boswell, K.M.; Wilson, M.P.; Cowan, J.H., Jr. A semiautomated approach to estimating fish size, abundance, and behavior from dual-frequency identification sonar (DIDSON) data. *N. Am. J. Fish. Manag.* **2008**, *28*, 799–807. [CrossRef]
6. Villon, S.; Chaumont, M.; Subsol, G.; Villéger, S.; Claverie, T.; Mouillot, D. Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between Deep Learning and HOG+ SVM methods. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, ACIVS 2016, Lecce, Italy, 24–27 October 2016; pp. 160–171.
7. Boulais, O.; Alaba, S.Y.; Ball, J.E.; Campbell, M.; Iftekhar, A.T.; Moorehead, R.; Primrose, J.; Prior, J.; Wallace, F.; Yu, H.; et al. SEAMAPD21: A large-scale reef fish dataset for fine-grained categorization. In Proceedings of the FGVC8: The Eight Workshop on Fine-Grained Visual Categorization, Online, 25 June 2021.
8. Gilby, B.L.; Olds, A.D.; Connolly, R.M.; Yabsely, N.A.; Maxwell, P.S.; Tibbetts, I.R.; Schoeman, D.S.; Schlacher, T.A. Umbrellas can work under water: Using threatened species as indicator and management surrogates can improve coastal conservation. *Estuar. Coast. Shelf Sci.* **2017**, *199*, 132–140. [CrossRef]
9. Langlois, T.; Goetze, J.; Bond, T.; Monk, J.; Abesamis, R.A.; Asher, J.; Barrett, N.; Bernard, A.T.; Bouchet, P.J.; Birt, M.J.; et al. A field and video annotation guide for Baited Remote Underwater stereo-video surveys of Demersal Fish Assemblages. *Methods Ecol. Evol.* **2020**, *11*, 1401–1409. [CrossRef]
10. Whitmarsh, S.K.; Fairweather, P.G.; Huveneers, C. What is big bruvver up to? methods and uses of Baited Underwater Video. *Rev. Fish Biol. Fish.* **2016**, *27*, 53–73. [CrossRef]
11. Perry, D.; Staveley, T.A.; Gullström, M. Habitat connectivity of fish in temperate shallow-water seascapes. *Front. Mar. Sci.* **2017**, *4*, 440. [CrossRef]
12. Stål, J.; Paulsen, S.; Pihl, L.; Rönnbäck, P.; Söderqvist, T.; Wennhage, H. Coastal Habitat support to fish and fisheries in Sweden: Integrating ecosystem functions into fisheries management. *Ocean Coast. Manag.* **2008**, *51*, 594–600. [CrossRef]
13. Nabi, M.M.; Senyurek, V.; Gurbuz, A.C.; Mehmet, K. Deep Learning-Based Soil Moisture Retrieval in CONUS Using CYGNSS Delay–Doppler Maps. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 6867–6881. [CrossRef]
14. Zhao, M.; Chang, C.H.; Xie, W.; Xie, Z.; Hu, J. Cloud shape classification system based on multi-channel cnn and improved fdm. *IEEE Access* **2020**, 44111–44124. [CrossRef]

15. You, L.; Jiang, H.; Hu, J.; Chang, C.H.; Chen, L.; Cui, X.; Zhao, M. GPU-accelerated Faster Mean Shift with euclidean distance metrics. In Proceedings of the 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Alamitos, CA, USA, 27 June–1 July 2022; pp. 211–216.
16. Jin, B.; Cruz, L.; Gonçalves, N. Pseudo RGB-D Face Recognition. *IEEE Sens. J.* **2022**. [CrossRef]
17. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis detection and three-dimensional localization of cut off point for vision-based banana robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [CrossRef]
18. Tang, Y.; Zhou, H.; Wang, H.; Zhang, Y. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. *Expert Syst. Appl.* **2022**, *211*, 118573. [CrossRef]
19. Huang, P.X.; Boom, B.J.; Fisher, R.B. Underwater live fish recognition using a balance-guaranteed optimized tree. In Proceedings of the Asian Conference on Computer Vision—ACCV 2012, Daejeon, Korea, 5–9 November 2012; pp. 422–433. [CrossRef]
20. Fisher, R.B.; Chen-Burger, Y.H.; Giordano, D.; Hardman, L.; Lin, F.P. *Fish4knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 104, p. 319.
21. Jäger, J.; Rodner, E.; Denzler, J.; Wolff, V.; Fricke-Neuderth, K. SeaCLEF 2016: Object Proposal Classification for Fish Detection in Underwater Videos. In Proceedings of the Working Notes of CLEF 2016, Évora, Portugal, 5–8 September 2016; pp. 481–489.
22. Joly, A.; Goëau, H.; Glotin, H.; Spampinato, C.; Bonnet, P.; Vellinga, W.P.; Planqué, R.; Rauber, A.; Palazzo, S.; Fisher, B.; et al. LifeCLEF 2015: Multimedia life species identification challenges. In Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages, Toulouse, France, 8–11 September 2015; pp. 462–483.
23. Zhuang, P.; Xing, L.; Liu, Y.; Guo, S.; Qiao, Y. Marine Animal Detection and Recognition with Advanced Deep Learning Models. In Proceedings of the CLEF 2017, Dublin, Ireland, 11–14 September 2017.
24. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
26. Ogunlana, S.; Olabode, O.; Oluwadare, S.; Iwasokun, G. Fish classification using support vector machine. *Afr. J. Comput. ICT* **2015**, *8*, 75–82.
27. Strachan, N. Recognition of fish species by colour and shape. *Image Vis. Comput.* **1993**, *11*, 2–10. [CrossRef]
28. White, D.J.; Svellingen, C.; Strachan, N.J. Automated measurement of species and length of fish by computer vision. *Fish. Res.* **2006**, *80*, 203–210. [CrossRef]
29. Salman, A.; Jalal, A.; Shafait, F.; Mian, A.; Shortis, M.; Seager, J.; Harvey, E. Fish species classification in unconstrained underwater environments based on deep learning. *Limnol. Oceanogr. Methods* **2016**, *14*, 570–585. [CrossRef]
30. Labao, A.B.; Naval, P.C. Simultaneous localization and segmentation of fish objects using multi-task CNN and dense CRF. In Proceedings of the Asian Conference on Intelligent Information and Database Systems, Yogyakarta, Indonesia, 8–11 April 2019; pp. 600–612.
31. Rekha, B.; Srinivasan, G.; Reddy, S.K.; Kakwani, D.; Bhattad, N. Fish detection and classification using convolutional neural networks. In Proceedings of the International Conference on Computational Vision and Bio Inspired Computing, Coimbatore, India, 25–26 September 2019; pp. 1221–1231.
32. Nery, M.S.; Machado, A.; Campos, M.F.M.; Pádua, F.L.; Carceroni, R.; Queiroz-Neto, J.P. Determining the appropriate feature set for fish classification tasks. In Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05), IEEE, Natal, Rio Grande do Norte, Brazil, 9–12 October 2005; pp. 173–180.
33. Li, X.; Shang, M.; Qin, H.; Chen, L. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–5.
34. 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*. [CrossRef] [PubMed]
35. Alaba, S.; Ball, J. Deep Learning-based Image 3D Object Detection for Autonomous Driving: Review. *TechRxiv* **2022**. [CrossRef]
36. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
37. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
38. Alaba, S.; Gurbuz, A.; Ball, J. A Comprehensive Survey of Deep Learning Multisensor Fusion-based 3D Object Detection for Autonomous Driving: Methods, Challenges, Open Issues, and Future Directions. *TechRxiv* **2022**. [CrossRef]
39. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
40. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 July 2008; pp. 1322–1328.
41. Zou, Y.; Yu, Z.; Kumar, B.; Wang, J. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 289–305.
42. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9268–9277.

43. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report TR-2009; University of Toronto: Toronto, ON, Canada, 2009.
44. Li, Y.; Wang, T.; Kang, B.; Tang, S.; Wang, C.; Li, J.; Feng, J. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10991–11000.
45. Dos Santos, A.A.; Gonçalves, W.N. Improving Pantanal fish species recognition through taxonomic ranks in convolutional neural networks. *Ecol. Inform.* **2019**, *53*, 100977. [CrossRef]
46. Hu, J.; Li, D.; Duan, Q.; Han, Y.; Chen, G.; Si, X. Fish species classification by color, texture and multi-class support vector machine using computer vision. *Comput. Electron. Agric.* **2012**, *88*, 133–140. [CrossRef]
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
48. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
49. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
50. Yang, T.J.; Howard, A.; Chen, B.; Zhang, X.; Go, A.; Sandler, M.; Sze, V.; Adam, H. Netadapt: Platform-aware neural network adaptation for mobile applications. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 285–300.
51. Avenash, R.; Viswanath, P. Semantic Segmentation of Satellite Images using a Modified CNN with Hard-Swish Activation Function. In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Prague, Czech Republic, 25–27 February 2019; pp. 413–420.
52. Alaba, S.Y.; Ball, J.E. WCNN3D: Wavelet Convolutional Neural Network-Based 3D Object Detection for Autonomous Driving. *Sensors* **2022**, *22*, 7010. [CrossRef]
53. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
54. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
55. Zheng, Q.; Yang, M.; Yang, J.; Zhang, Q.; Zhang, X. Improvement of generalization ability of deep CNN via implicit regularization in two-stage training process. *IEEE Access* **2018**, *6*, 15844–15869. [CrossRef]
56. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv* **2017**, arXiv:1706.02677.
57. Congcong Li High Quality, Fast, Modular Reference Implementation of SSD in PyTorch. 2018. Available online: https://github.com/lufficc/SSD (accessed on 20 September 2022).