IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 February 2023, accepted 19 March 2023, date of publication 28 March 2023, date of current version 6 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3262649

RESEARCH ARTICLE

Semantic Segmentation of Fish and Underwater Environments Using Deep Convolutional Neural Networks and Learned Active Contours

MIGUEL CHICCHON[1], HECTOR BEDON[2], CARLOS R. DEL-BLANCO[3], AND IVAN SIPIRAN[4]

[1]Escuela de Posgrado, Pontificia Universidad Católica del Perú, Lima 15088, Peru
[2]School of Technical Sciences and Engineering, Madrid Open University (UDIMA), 28040 Madrid, Spain
[3]Information Processing and Telecommunications Center, Grupo de Tratamiento de Imágenes (GTI), ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[4]Department of Computer Science, University of Chile, Santiago 8331150, Chile

Corresponding author: Miguel Chicchon (mchicchon@pucp.edu.pe)

This work was supported in part by the National Program for Innovation in Fisheries and Aquaculture of Peru, in part by the University of Lima, and in part by the Open University of Madrid.

ABSTRACT The conservation of marine resources requires constant monitoring of the underwater environment by researchers. For this purpose, visual automated monitoring systems are of great interest, especially those that can describe the environment using semantic segmentation based on deep learning. Although they have been successfully used in several applications, such as biomedical ones, obtaining optimal results in underwater environments is still a challenge due to the heterogeneity of water and lighting conditions, and the scarcity of labeled datasets. Even more, the existing deep learning techniques oriented to semantic segmentation only provide low resolution results, lacking the enough spatial details for a high performance monitoring. To address these challenges, a combined loss function based on the active contour theory and level set methods is proposed to refine the spatial segmentation resolution and quality. To evaluate the method, a new underwater dataset with pixel annotations for three classes (fish, seafloor, and water) was created using images from publicly accessible datasets like SUIM, RockFish, and DeepFish. The performance of architectures of convolutional neural networks (CNNs), such as UNet and DeepLabV3+, trained with different loss functions (cross entropy, dice, and active contours) was compared, finding that the proposed combined loss function improved the segmentation results by around 3%, both in the metric Intercept Over Union (IoU) as in Hausdorff Distance (HD).

INDEX TERMS Active contour, computer vision, convolutional neural network, deep learning, semantic segmentation, underwater images.

I. INTRODUCTION

Sustaining and conserving the seas, oceans, and marine resources is one of the Sustainable Development Goals of the United Nations (SDGs). The strategies for achieving this goal are to protect and manage coastal and marine ecosystems with the purpose of adopting the required measures to preserve the oceans' health and to facilitate artisanal fishermen's access to marine resources [1].

To that end, automated marine ecosystem monitoring is a key tool that takes advantage of the increasing availability of video collections of marine life [2]. The goal of a monitoring system is to perform a scene comprehension and react appropriately to the surrounding events and environment conditions [3]. One of the key system modules is the segmentation that generates labels for each pixel describing the object to which it belongs. The location and comprehension of the objects in the scene provided by the segmentation serve as the foundation for a more in-depth analysis [4].

In the last years, different advanced neural network architectures from the field of deep learning have been

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

successfully incorporated in numerous application fields [5], [6], [7]. In natural image segmentation challenges, these architectures have proven to be much better than previous systems based on handcrafted features and shallow classifiers [8], [9]. It is remarkable the recent progress in the field of medical imaging, where the most promising algorithms belong to the family of active contour methods [10], [11] and deep neural networks [12], [13], [14]. However, image or video segmentation in underwater environments presents additional significant challenges due to the absorption, scattering, attenuation of light rays, effects of suspended particles in the water [15], and the potential dynamic camera condition [16].

This paper proposes to combine convolutional neural networks (CNN) and learned active contour to improve the quality of the segmentation in underwater imagery. More specifically, the main contributions of this work are as follows: (1) a comparison of the performance of different segmentation algorithms based on deep learning architectures for the task of fish segmentation in underwater environments; (2) an strategy to improve the segmentation performance in underwater environments by combining different loss functions (cross-entropy, dice and active contours); and (3) the creation of a new underwater dataset whose images have been obtaining from existing databases (SUIM, RockFish, and DeepFish), but whose annotations have been fully relabeled for the purpose of semantic segmentation considering three classes (water background, seafloor/obstacle, and fish). This dataset along with the best obtained models are available at https://github.com/miguelCh2912/Fish-and-Underwater-Environments.

The remainder of this article is structured as follows. Section II describes the state of the art in image segmentation in underwater environments, including active contour methods and deep learning models. Section III presents the proposed segmentation system, describing the pre-processing, the neural network architectures for semantic segmentation, and the loss functions used to combine them. Section IV describes the experiments, including quantitative/qualitative results and an ablation study. Finally, Section V presents the conclusions and future work.

II. RELATED WORKS

Recently, there has been an increasing interest in using image segmentation for estimating the volume and quantity of fish on the seabed. Traditional segmentation methods based on image processing techniques [15] have been widely outperformed by deep learning techniques for many marine applications. In [17], the neural network mask R-CNN is used to detect various fish types. In [18], authors compare various Convolutional Neural Network (CNN) approaches for classifying and semantically segmenting coral reefs in underwater images. King et al. [19] present a three-dimensional reconstruction of coral reefs combining CNN and Siamese network architectures. Some works are oriented to the problem of the data scarcity for training the underlying deep learning models. In [20], a semantic segmentation approach for coral reefs is proposed taking into account the sparsity of the annotated labels. Alonso et al. [21] describe a method for working with few labeled data that utilizes adaptive superpixel segmentation propagation to increase sparse annotations.

To address the lack of annotated data, several works have focused on the creation of large databases. An underwater large-scale dataset for the semantic segmentation, called SUIM, is presented in [22]. They also propose an encoder-decoder model (SUIM-Net) to balance the performance and computational efficiency. In [23], a dataset of images of real and simulated environments is presented, and explores different strategies of segmentation, fine-tuning, and image restoration. Complementarily, [24] presents the DeepFish benchmark for classification, counting, location, and segmentation tasks, allowing the training of multitasking models.

Other works address the enhancement of underwater imagery to compensate for the challenges of this type of imagery. In [25], a CNN-based approach to enhance underwater imagery is proposed for obstacle detection by combining monocular semantic image segmentation with sparse stereo point clouds. Also, Generative Adversarial Networks (GANs) have been successfully used to improve the quality and resolution of underwater images [3], [26], [27].

Despite deep neural networks' success in semantic segmentation tasks, some limitations remain, such as the loss of spatial details due to the low resolution of the segmentation results. Some methods make use of a post-processing stage to refine the segmentation results, such as Conditional Random Fields (CRFs) [28], at the expense of increasing the requirements of memory and computational cost. Recently, numerous approaches based on variational methods and active contours have been proposed to address the limitations in the results of segmentation by deep neural networks. The first approaches proposed the use of active contour methods as a post-processing step. Later approaches integrated the active contours as part of the network architecture, performing an end-to-end training. For example, Hatamizadeh et al. in [29] first used active contours without edges (ACWE) along with the Chan-Vese method as a post-processing stage in DALS, and later, in [30], they embedded their approach in an end-to-end learning neural network in DCAC. Lastly, in [31], they exchanged the Chan-Vese method with the Localizing Region-Based Active Contours (LRACWE) in TDAC. Similarly, Zang et al. [32] evaluated an end-to-end approach using the Split Bregman method for active contours (SBACWE) in DACN.

Other works addressing the loss of spatial details have focused on the design and combination of new loss functions for guiding the training of the underlying neural network architectures. Kim et al. [33] proposed a loss function based on level set theory for multiple classes, decomposing the ground truth into a binary image for each class. Kim and Ye

[34] presented a loss function based on the Mumford-Shah functional for semi-supervised and unsupervised segmentation. An elastic loss function is used in [35] and [36] that is inspired by Euler's Elastic model and the mean curvature of objects. Ma et al. [37] integrated global geometric information of objects in a loss function using Geodesic Active Contours (GAC). And, Le et al. [38] combined a CNN based level set approach with recurrent networks, named Recurrent Level Set (RLS).

III. METHODS

The methodology of the segmentation process is shown in Figure 1, involving two steps. The first one, the preprocessing step, involves the acquisition and labeling of images, followed by the database division into training, validation, and testing partitions, with a proportion of 70%, 15%, and 15%, respectively. The training partition is further subject to data augmentation to alleviate the overfitting. The second step, the training one, performs the supervised training of deep learning models based on the U-Net and DeepLabV3+ architectures, both characterized by an encoder-decoder scheme. Different pretraining weights for the encoder stage are used to improve the training convergence. The optimization process for the training is guided by the proposed combination of loss functions. This is carried out by a weighted loss function that combines, in turn, the cross-entropy, dice, and active contour losses. Finally, the segmentation system is evaluated using the quality metrics of Intersection Over Union (IoU) and Hausdorff Distance (HD).

A. IMAGE DATA PREPROCESSING

1) IMAGE DATASET

The dataset used in the experiment is composed of 1824 images obtained from images of underwater environments. The images come from different existing datasets, but the associated labels do not fit the proposed segmentation task. Subsequently, they have been specifically relabeled at the pixel level into 3 classes: water background (BW), seafloor/obstacles (SO), and fish (F). A total of 918 images of several sizes were chosen from the Submarine Imagery (SUIM) dataset [22], initially labeled in 8 semantic classes (ship-wrecks/ruins, plants/aquatic flora, human divers, water body background, robots and instruments, reefs and other invertebrates, fish and other vertebrates, and seafloor/rocks), which have been grouped in the three considered classes. The other 150 images were selected and labeled from DeepFish dataset [24], consisting of underwater images from 20 habitats in tropical Australia. These images have little background variability, as they are video sequences from static cameras. Other 280 images were obtained from the RockFish [39] dataset and labeled. These images contain moving and stationary fish in a complex rocky seabed background, acquired by a remote-controlled submarine vehicle. Over 200 images were obtained from videos captured by a HD camera mounted on Chasing Gladius Mini underwater drones. They contain footage from beaches and fish farms in Peru. The remaining images have been obtained from public repositories on the internet containing seabed scenes from different parts of the world.

All the images were resized at 256 × 320 pixels and divided into three subsets with 1280/272/272 images for the training/validation/testing, respectively. Table 1 shows more detail.

TABLE 1. Number of images according to the origin and split of the dataset.

Dataset Origin	Training	Validation	Testing
SUIM	646	138	135
DeepFish	114	28	23
RockFish	194	43	43
Our	143	26	31
Internet	183	37	40

2) IMAGE ANNOTATION

The software tool MatLab Image Labeler has been used for per-pixel annotation of the underwater images. This software offers a graphical user interface that enables the labeling of ground truth data in a collection of images. Then, a MatLab script generates *.bmp files with pixels that are different colors depending on the classes considered: water background (black), seafloor/obstacle (red), and fish (yellow).

3) IMAGE DATA AUGMENTATION

Data augmentation improves the robustness and generalization abilities of deep learning models. The following strategies are used to increase the number of samples in the training partition: random crop and horizontal flip; shift, scale, and rotation geometric transformations; random variations of brightness and contrast to ensure operation in a variety of lighting settings; shifts in HSV and RGB color representations. Furthermore, due to the frequent presence of suspended particles in the water, the functions of blur, Gaussian noise, and gamma noise were also applied. Details can be found in the supplementary materials.

B. DEEP MODEL ARCHITECTURE FOR SEMANTIC SEGMENTATION

In this study, we compared the DeepLabV3+ and U-Net architectures within the proposed workflow. This is due to the good results obtained in similar environments [4], [22], [40].

1) U-NET

The U-Net architecture [41] is composed of two paths, namely, the contraction path (encoder) and the expansion path (decoder), and it incorporates the concept of skip connections between the encoder and decoder layers to recover lost spatial features. The encoder path begins with a pairwise convolution

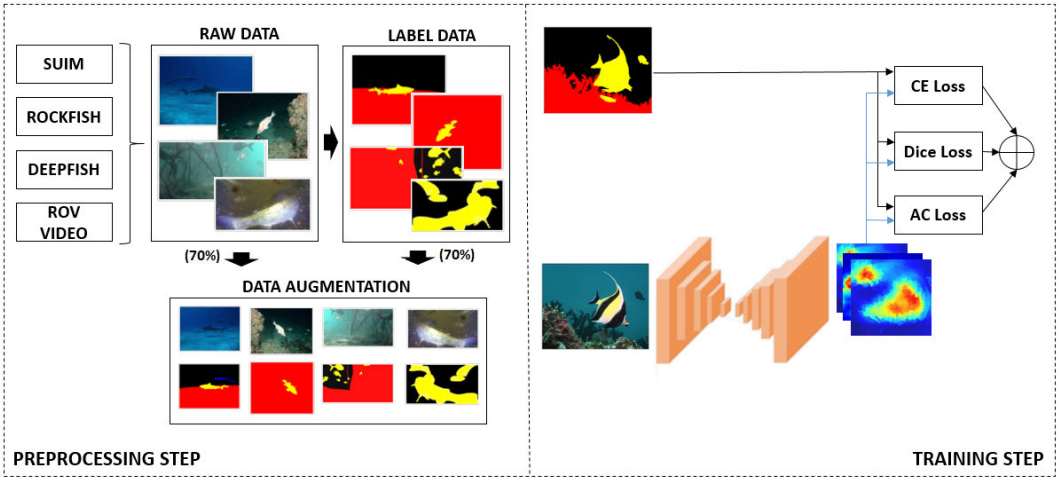

FIGURE 1. Overview of the proposed method.

layer of size 3 × 3, followed by batch normalization and ReLU activation, and finally by 2 × 2 maximum pooling operations. The decoder path begins with a 2 × 2 transposed convolution operation that reduces the number of feature channels, followed by two 3 × 3 convolutions with batch normalization and ReLU activation. The convolutional layers of each encoder block and the corresponding decoder blocks present the following number of feature maps (64, 128, 256, 512, 1024).

In our experiments, we used a modification of the U-Net architecture. The first modification is related to the use of pre-trained networks in the encoder path and the second modification with the use of attention mechanisms in the decoder path. Figure 2 shows how the attention mechanisms are incorporated into the decoder by adding a scSE block following a combination of convolution, batch normalization, and ReLU activation. The scSE blocks joint spatial and channel squeeze and excitation [42]. By incorporating global spatial information, the scSE blocks recalibrate the channels. Additionally, the sSE blocks generate a spatial attention map, indicating the areas on which the network should focus its attention to help the segmentation. The convolutional layers of each decoder block present the following number of feature maps (256, 128, 64, 32, 16), refer to Figure 2.

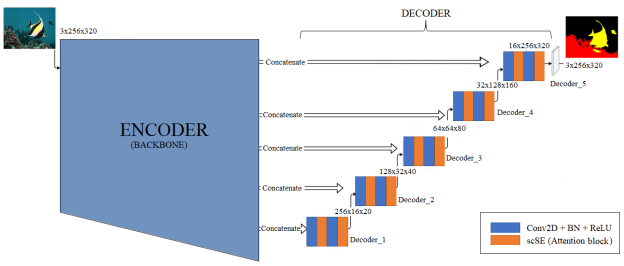

FIGURE 2. U-Net architecture with scSE attention block.

2) DEEPLABV3+

The DeepLabv3+ architecture [43] is composed of encoder and decoder paths. Initially, contextual information is encoded to later retrieve the object's bounds. The encoder is based on DeepLabv3 [44], and it uses Atrous Convolution to extract deep CNN computed features at any resolution. The output step is defined in this architecture as the ratio of the spatial resolution of the input image to the final output (before global grouping). The Atrous Spatial Pyramid Pooling module enables the analysis of convolutional features at various scales by combining Atrous convolutions of various ratios with image-level features. The decoder performs a bilinear sampling by a factor of 4 on the encoder features and then concatenates the result with the low-level features of the backbone with the same spatial resolution. After that, the number of channels is reduced by performing a 1 × 1 convolution on the low-level features. Finally, following concatenation, 3 × 3 convolutions are applied to refine the features, and a simple bilinear upward sampling by a factor of 4 is performed. In our experiments, the dilation rates for the ASPP modulate were (6, 12, 18) with a downsampling factor of 16 between the input and the output, as illustrated in Figure 3.

C. DEEP MODEL ARCHITECTURE FOR BACKBONE

ImageNet [45] is a database containing over a million object categories that is used in the well-known ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [46]. As a result of this type of challenge, large databases are generated with pre-trained networks that later serve as the initial stages or network backbone for more complex tasks such as object detection or segmentation. For the coder path of U-Net and DeepLabv3+, different pre-trained networks in the ImageNet database were used, such as the backbone with five stages in all cases. The architecture used in this article is summarized below.

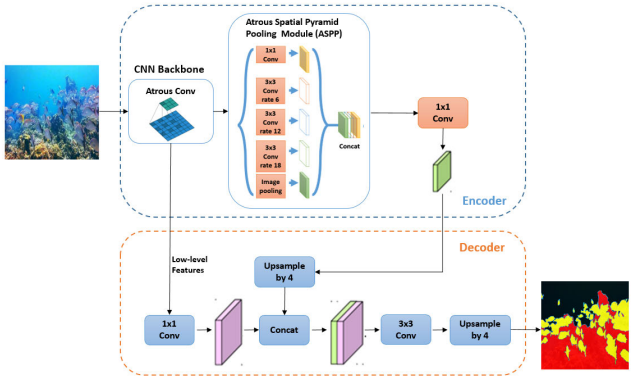

FIGURE 3. DeepLabV3+ architecture [43].

1) MOBILENET V2

Optimized architecture for mobile devices [47]. It consists of an inverted residual structure connected by shortcuts between the narrow bottleneck layers. The middle expansion layer uses narrow-depth convolutions to eliminate output features as a source of nonlinearity and maintain representativeness. The MobileNet v2 architecture consists of a convolution layer with 32 filters and 19 residual bottleneck layers. Additionally, ReLU6 is used to provide nonlinearity in combination with batch normalization. This network demonstrated a strong correlation between inference time and prediction accuracy.

2) EFFICIENTNET-B

Family of models scaled from a baseline network known as EfficientNet-B0 [48]. This network was produced through a multi-objective neural architecture search that optimizes both accuracy and efficiency. It is mainly composed of the mobile inverted bottleneck MBConv, which includes squeeze-and-excitation optimization. All other models are scaled by carefully balancing the depth, width, and resolution of the network using a compound scaling method. This method relates a user-specified coefficient that controls how many more resources are available, and constants that specify how to allocate these additional resources to the width, depth, and resolution of the network. The network architectures used in this study were EfficientNet-B0 and EfficientNet-B7.

3) RESNEST

Split-Attention Network (ResNeSt) [49] is a ResNet variant and consist of stacking several Split-Attention blocks. A split-Attention block consist of a feature map group and split attention operations. Each block incorporates a channel-wise attention strategy to capture the interdependencies of the featuremap. Furthermore, the combination with a multi-path network layout approach allows learning diverse representations. The network architecture used in this research is ResNeSt-269e, since in experiments it has obtained a higher precision than EfficientNet-B7 with 32% less latency [49].

D. LOSS FUNCTIONS

1) CROSS-ENTROPY LOSS

The standard multi-class cross-entropy loss function is given by:

$$L_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\sum_{p=1}^{P} T_{ncp}\ln\left(Y_{ncp}\right). \tag{1}$$

where P, C, and N are the number of pixels, the number of classes, and the mini-batch size, respectively. Also, T_{ncp} is a binary indicator that indicates whether class label c is the correct classification for pixel p, and Y_{ncp} is the corresponding predicted probability obtained from the softmax function's evaluation of the logits values, F_{cp}, by the softmax function; see (2) and Figure 4.

$$Y_{cp} = \frac{e^{F_{cp}}}{\sum_{c=1}^{C} e^{F_{cp}}}. \tag{2}$$

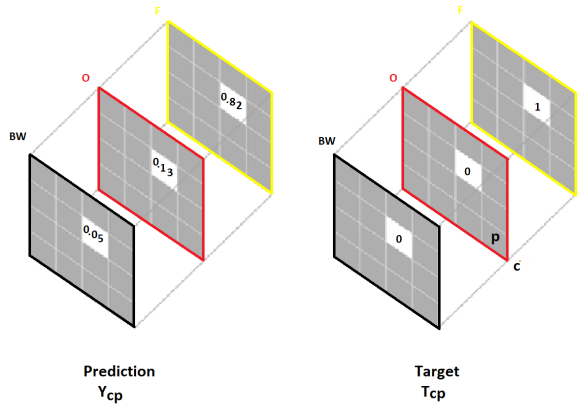

FIGURE 4. Relationship between predicted and labeled GT pixels.

2) DICE LOSS

The Dice metric, which is defined in the following section, can be expressed as a loss function. The smooth form of the proposals is described in [50]:

$$L_{Dice} = \frac{1}{N}\sum_{n=1}^{N}\left(1 - \frac{2\sum_{c=1}^{C}\sum_{p=1}^{P} T_{ncp}Y_{ncp}}{\sum_{c=1}^{C}\sum_{p=1}^{P}\left(\left\|T_{ncp}\right\|_{pp} + \left\|Y_{ncp}\right\|_{pp}\right)}\right). \tag{3}$$

where $\|*\|_{pp}$ represents the norm pp, finding in the literature the use of norm 1 and 2. It is common to consider using the whole batch to reduce the effect of the absence of a class in some images, according to:

$$L_{Dice} = 1 - \frac{2\sum_{n=1}^{N}\sum_{c=1}^{C}\sum_{p=1}^{P} T_{ncp}Y_{ncp}}{\sum_{n=1}^{N}\sum_{c=1}^{C}\sum_{p=1}^{P}\left(\left\|T_{ncp}\right\|_{pp} + \left\|Y_{ncp}\right\|_{pp}\right)}. \tag{4}$$

3) LEARNING ACTIVE CONTOUR

Active contour models consist of enveloping a curve, subject to the constraints of a given image. The criterion for modifying the curve shape can be based on various features, such as the intensity gradient, color, texture, or other image features. The active contour models continuously update the curve shape and must stop on the boundary of the object [51].

Based on [33], [52], we use a loss function inspired by the theory of active contour models without edges, specifically the representation through the level set method of the energy functional proposed by Chan-Vese [51] and defined by:

$$\begin{aligned} &F_{\xi}\left(c_1, c_2, \phi\right) \\ &= \mu \, Length\left(\phi\right) \\ &\quad + \lambda_1 \int_{\Omega} \left|u_0\left(x, y\right)\right) - c_1\right|^2 H_{\xi}\left(\phi\left(x, y\right)\right) dxdy \\ &\quad + \lambda_2 \int_{\Omega} \left|u_0\left(x, y\right)\right) - c_2\right|^2 \left(1 - H_{\xi}\left(\phi\left(x, y\right)\right)\right) dxdy. \end{aligned} \tag{5}$$

where u_0 is the image to be segmented with the domain Ω. The parameters μ, λ_1, and λ_2 are weighting coefficients for each term; c_1 is the internal grayscale average of the evolving curve, and c_2 is the external grayscale average of the evolving curve, defined by (7). ϕ is the level set function, and H_{ξ} is the Approximated Heaviside function, defined by:

$$H_{\xi}\left(\phi\right) = \frac{1}{2}\left[1 + \frac{2}{\pi} \arctan\left(\frac{\phi}{\xi}\right)\right]. \tag{6}$$

In addition, the terms c_1 and c_2 are defined as:

$$\begin{aligned} c_1\left(\phi\right) &= \frac{\int_{\Omega} u_0\left(x, y\right) H_{\xi}\left(\phi\left(x, y\right)\right) dxdy}{\int_{\Omega} H_{\xi}\left(\phi\left(x, y\right)\right) dxdy}, \\ c_2\left(\phi\right) &= \frac{\int_{\Omega} u_0\left(x, y\right) \left(1 - H_{\xi}\left(\phi\left(x, y\right)\right)\right) dxdy}{\int_{\Omega} \left(1 - H_{\xi}\left(\phi\left(x, y\right)\right)\right) dxdy}. \end{aligned} \tag{7}$$

The first term corresponds to the length of the evolving curve and is sensitive to the object's size, so in this research μ is set to zero because the input images have multiple size objects [33]. Then, the multi-class semantic segmentation is calculated based on reconstructing the dense binary ground truth for each class separately. Considering λ_1 and λ_2 equal to one, the Chan-Vese loss function for deep learning is formulated as follows:

$$\begin{aligned} L_{CV} = \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} \Bigg(&\sum_{p=1}^{P} \left|T_{ncp} - c_{ncp,1}\right|^2 H_{\xi}\left(\phi_{ncp}\right) \\ &+ \sum_{p=1}^{P} \left|T_{ncp} - c_{ncp,2}\right|^2 \left(1 - H_{\xi}\left(\phi_{ncp}\right)\right) \Bigg). \end{aligned} \tag{8}$$

where the level set function H_{ξ} is a shifted dense probability map that is estimated from $\xi_{ncp} = Y_{ncp} - 0.5 \in [-0.5, 0.5]$ and the average intensity of the binary ground truth map T_{ncp} for the inside and outside contours are:

$$\begin{aligned} c_{ncp,1}\left(\phi_{ncp}\right) &= \frac{\sum_{p=1}^{P} T_{ncp} H_{\xi}\left(\phi_{ncp}\right)}{\sum_{p=1}^{P} H_{\xi}\left(\phi_{ncp}\right)}, \\ c_{ncp,2}\left(\phi_{ncp}\right) &= \frac{\sum_{p=1}^{P} T_{ncp}\left(1 - H_{\xi}\left(\phi_{ncp}\right)\right)}{\sum_{p=1}^{P} \left(1 - H_{\xi}\left(\phi_{ncp}\right)\right)}. \end{aligned} \tag{9}$$

To help the network learn more discriminative features of objects and handle the problem of class imbalance, we combine the Chan-Vese loss function with the Dice loss and cross-entropy loss as follows:

$$L = \alpha L_{CE} + \beta L_{Dice} + \gamma L_{CV}. \tag{10}$$

where $\alpha, \beta, \gamma > 0$ denote the weights used to achieve a trade-off between the loss terms. We have used an empirical approach similar to that used by Kervadec et al. [53] to obtain the appropriate weights. In our experiment, β is set equal to 1 as a starting point, and then the approximate relationship between L_{Dice} and the other loss functions (L_{CE} and L_{CV}) is established to find the initial values of α and γ. Then, as a function of experimentation, balancing is performed.

		Predicted class	
		Positive	Negative
True class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

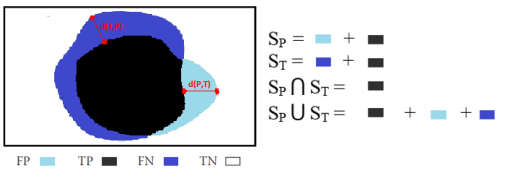

FIGURE 5. Confusion matrix and notation.

IV. EXPERIMENTS AND RESULTS

A. EVALUATION METRIC

Generally, segmentation in computer vision is evaluated using region-based quantitative metrics and statistical analysis [54]. Metrics based on relevance enable an analysis of how many of the image's segmented pixels were correctly detected or not compared to the data labeled ground truth (GT). The relationship between the confusion matrix concept and image segmentation is illustrated in Figure 5. True positives (TP) are pixels with correct labels, whereas false negatives (FN) are pixels with incorrect labels. The pixels that do not belong but are incorrectly labeled as belonging to that class as false positives (FP), and the number of pixels that do not belong but are correctly labeled as true negatives (TN).

We can mention the following well-known relevance metrics:

1) IOU

False positives are penalized using the intersection over union (IoU) or Jaccard's coefficient of similarity [55]. For

each class, this is expressed by the relationship shown in (11). mIoU is the average of the IoU of all classes in all images in the dataset; see (12). When the class sizes of the images are unbalanced, the frequency-weighted IoU metric is used to minimize the effect of the small class error on the overall score; see (13).

$$IoU = \frac{|S_T \cap S_P|}{|S_T \cup S_P|} = \frac{TP}{TP + FP + FN}. \quad (11)$$

$$mIoU = \frac{1}{C} \sum_{c=1}^{C} \frac{p_{ii}}{t_i + \sum_j p_{ij} - p_{ii}}. \quad (12)$$

$$mfwIoU = \frac{1}{\sum_k t_k} \sum_{c=1}^{C} \frac{t_i p_{ii}}{t_i + \sum_j p_{ij} - p_{ii}}. \quad (13)$$

where p_{ij} is the number of predicted pixels of class i that belong to class j, and t_i is the total number of pixels of class i in the segmentation of the label or ground truth. Figure 5 shows the case of two classes. The number of elements in the intersection set grows as the similarity grows, whereas the cardinality of the union set decreases. As a result of (11), when the similarity is maximized, the value of IoU equals one. Additionally, IoU will be zero in the absence of similarity.

2) DICE

Dice's coefficient of similarity is defined as the relationship between twice the intersection and the total pixels predicted and labeled GT, and its definition is given by (14). This metric determines the accuracy of the segmentation limits by evaluating the number of correctly labeled pixels [49]. Dice is positively correlated with IoU but has a lower penalty for incorrect results [20].

$$Dice = \frac{2\,|S_T \cap S_P|}{|S_T| + |S_P|} = \frac{2 * TP}{2 * TP + FP + FN}. \quad (14)$$

3) HD

The Hausdorff distance (HD) metric measures the longest Euclidean distance, d(T,P), between the ground truth contour (T) and the predicted contour (P), and its definition is given by (15) [56], [57]. The HD95 metric is slightly more stable to small outliers and computes the 95th percentile of the Hausdorff distance (0 mm for a perfect segmentation). Therefore, the HD95 metric is used in this research.

$$HD(T, P) = max \left\{ \sup_{t \in T} \inf_{p \in P} d(T, P), \sup_{p \in P} \inf_{t \in T} d(T, P) \right\}. \quad (15)$$

B. EXPERIMENTAL SETTING

The experiment used the Pytorch deep learning framework. The following are the characteristics and configurations of the hardware used: Intel Core i9-9820X CPU @ 3.30 GHz with 20 threads, 96 GB of RAM, Nvidia GeForce RT2800 Ti GPU with 11 GB of RAM, 1TB of solid state drive, NVIDIA driver version 470.74, CUDA version 11.4, CUDNN 7.6.5 neural network acceleration library, Linux Ubuntu 20.04 LTS operating system, Python version 3.8.10, and Pytorch version 1.9.1 with some packages [58].

A 5-layers U-Net, SegNet, and DenseUNet (DenseNet-121 backend [59]) networks are used as a baseline for end-to-end training for the experiments. It is also experimented with models based on a U-Net network with MobileNetV2, EfficientNet-B0, EfficientNet-B7, ResNet-50, and ResNeSt-296e, such as feature extraction models or encoder layers; and decoder layers with scSE blocks. The rest of the models are DeepLabV3+ networks with a coding structure ASPP of 6, 12, and 18 holes and the following feature extraction models: MobileNetV2, EfficientNet-B0, and EfficientNet-B7.

Each model runs for a maximum of 600 epochs. Adam is chosen as an optimizer with beta 1 and beta 2 by default (0.9 and 0.9999). To avoid overadjusting, we use a ReduceLROnPlateau scheduler with a reduction factor of 0.1 and patience of 40, and early stopping with patience of 60. Based on the GPU memory limitations, batch sizes of 4, 8, and 16 were used. Additionally, the value of 0.0001 is selected as the learning rate obtained from a quick search.

The objective of the next experiments is to demonstrate the efficacy of the proposed loss function and the selection of the best models by quantitative and qualitative comparison.

TABLE 2. Mean IoU performance (mean and standard deviation) of our proposed combined loss function compared to the separate loss functions.

Model	L_{CE}	L_D	L_{CV}	$L_{CE+D+CV}$
DLV3+ MnV2	$83.91_{(0.13)}$	$84.11_{(0.31)}$	$\mathbf{84.72}_{(0.28)}$	$84.69_{(0.09)}$
DLV3+ Eff-B0	$83.88_{(0.46)}$	$84.36_{(0.48)}$	$84.64_{(0.33)}$	$\mathbf{86.04}_{(0.25)}$
U-Net-s MnV2	$84.69_{(0.07)}$	$84.92_{(0.31)}$	$85.40_{(0.12)}$	$\mathbf{85.65}_{(0.18)}$
U-Net-s Eff-B0	$84.91_{(0.44)}$	$85.51_{(0.27)}$	$85.81_{(0.32)}$	$\mathbf{86.10}_{(0.18)}$
SegNet	$74.65_{(0.34)}$	$74.14_{(0.30)}$	$74.13_{(0.46)}$	$\mathbf{77.08}_{(0.10)}$
U-Net	$82.04_{(0.66)}$	$82.04_{(0.15)}$	$82.73_{(0.53)}$	$\mathbf{83.37}_{(0.35)}$
DenseUNet	$80.89_{(0.51)}$	$80.01_{(0.27)}$	$80.29_{(0.24)}$	$\mathbf{81.57}_{(0.18)}$
DLV3+ Eff-B7	$85.91_{(0.13)}$	$85.91_{(0.13)}$	$86.01_{(0.33)}$	$\mathbf{86.04}_{(0.25)}$
U-Net-s Eff-B7	$85.79_{(0.32)}$	$86.15_{(0.37)}$	$86.57_{(0.38)}$	$\mathbf{86.62}_{(0.31)}$
U-Net-s Rn50	$85.87_{(0.31)}$	$85.82_{(0.38)}$	$85.76_{(0.18)}$	$\mathbf{86.20}_{(0.27)}$
U-Net-s Rns269e	$86.66_{(0.48)}$	$87.07_{(0.54)}$	$87.31_{(0.39)}$	$\mathbf{87.45}_{(0.17)}$

TABLE 3. Mean Dice performance (mean and standard deviation) of our proposed combined loss function compared to the separate loss functions.

Model	L_{CE}	L_D	L_{CV}	$L_{CE+D+CV}$
DLV3+ MnV2	$91.06_{(0.08)}$	$91.19_{(0.19)}$	$\mathbf{91.58}_{(0.17)}$	$91.55_{(0.10)}$
DLV3+ Eff-B0	$91.07_{(0.27)}$	$91.36_{(0.29)}$	$91.52_{(0.20)}$	$\mathbf{91.67}_{(0.16)}$
U-Net-s MnV2	$91.55_{(0.04)}$	$91.70_{(0.18)}$	$91.99_{(0.07)}$	$\mathbf{92.14}_{(0.11)}$
U-Net-s Eff-B0	$91.70_{(0.26)}$	$92.06_{(0.16)}$	$92.25_{(0.19)}$	$\mathbf{92.41}_{(0.10)}$
SegNet	$84.80_{(0.24)}$	$84.40_{(0.24)}$	$84.42_{(0.40)}$	$\mathbf{86.54}_{(0.80)}$
U-Net	$89.96_{(0.31)}$	$89.86_{(0.08)}$	$90.33_{(0.33)}$	$\mathbf{90.73}_{(0.21)}$
DenseUNet	$89.11_{(0.34)}$	$88.60_{(0.21)}$	$88.83_{(0.15)}$	$\mathbf{89.62}_{(0.12)}$
DLV3+ Eff-B7	$91.90_{(0.05)}$	$92.32_{(0.08)}$	$92.37_{(0.19)}$	$\mathbf{92.39}_{(0.14)}$
U-Net-s Eff-B7	$92.25_{(0.19)}$	$92.46_{(0.21)}$	$92.71_{(0.22)}$	$\mathbf{92.73}_{(0.18)}$
U-Net-s Rn50	$92.27_{(0.18)}$	$92.24_{(0.23)}$	$92.19_{(0.11)}$	$\mathbf{92.47}_{(0.10)}$
U-Net-s Rns269e	$92.75_{(0.28)}$	$92.99_{(0.31)}$	$93.13_{(0.23)}$	$\mathbf{93.22}_{(0.10)}$

TABLE 4. Mean HD performance (mean and standard deviation) of our proposed combined loss function compared to the separate loss functions.

Model	L_{CE}	L_D	L_{CV}	$L_{CE+D+CV}$
DLV3+ MnV2	$55.47_{(0.38)}$	$56.80_{(0.86)}$	$55.23_{(0.79)}$	$\mathbf{54.56}_{(1.01)}$
DLV3+ Eff-B0	$53.87_{(0.83)}$	$55.02_{(1.71)}$	$54.05_{(1.52)}$	$\mathbf{53.64}_{(1.16)}$
U-Net-s MnV2	$57.45_{(1.49)}$	$55.93_{(1.05)}$	$54.43_{(1.79)}$	$\mathbf{53.54}_{(1.53)}$
U-Net-s Eff-B0	$55.46_{(1.51)}$	$56.18_{(1.71)}$	$55.37_{(1.34)}$	$\mathbf{53.88}_{(1.06)}$
SegNet	$80.04_{(1.51)}$	$83.07_{(1.84)}$	$83.19_{(1.94)}$	$\mathbf{78.03}_{(0.84)}$
U-Net	$66.04_{(1.73)}$	$67.12_{(1.03)}$	$66.00_{(1.67)}$	$\mathbf{64.42}_{(0.73)}$
DenseUNet	$67.27_{(1.30)}$	$70.42_{(1.60)}$	$64.82_{(1.06)}$	$\mathbf{62.79}_{(0.42)}$
DLV3+ Eff-B7	$50.62_{(0.79)}$	$50.65_{(1.26)}$	$\mathbf{50.14}_{(0.90)}$	$50.63_{(0.86)}$
U-Net-s Eff-B7	$52.03_{(1.16)}$	$52.88_{(0.91)}$	$52.10_{(0.76)}$	$\mathbf{51.90}_{(1.38)}$
U-Net-s Rn50	$54.16_{(1.77)}$	$55.88_{(1.49)}$	$55.69_{(1.33)}$	$\mathbf{52.25}_{(0.44)}$
U-Net-s Rns269e	$52.90_{(1.40)}$	$51.99_{(1.14)}$	$51.77_{(1.9)}$	$\mathbf{51.19}_{(1.37)}$

TABLE 5. Mean HD95 performance (mean and standard deviation) of our proposed combined loss function compared to the separate loss functions.

Model	L_{CE}	L_D	L_{CV}	$L_{CE+D+CV}$
DLV3+ MnV2	$30.92_{(0.43)}$	$30.64_{(1.14)}$	$29.47_{(0.61)}$	$\mathbf{28.59}_{(0.65)}$
DLV3+ Eff-B0	$28.42_{(1.01)}$	$28.78_{(1.57)}$	$27.86_{(1.29)}$	$\mathbf{27.65}_{(0.98)}$
U-Net-s MnV2	$31.17_{(0.30)}$	$30.12_{(0.85)}$	$27.74_{(0.81)}$	$\mathbf{27.63}_{(0.55)}$
U-Net-s Eff-B0	$28.04_{(1.13)}$	$29.16_{(1.61)}$	$27.75_{(0.64)}$	$\mathbf{26.53}_{(0.81)}$
SegNet	$52.10_{(1.67)}$	$54.55_{(0.50)}$	$53.79_{(1.15)}$	$\mathbf{49.24}_{(0.84)}$
U-Net	$37.35_{(1.67)}$	$37.57_{(1.23)}$	$36.45_{(0.94)}$	$\mathbf{35.64}_{(0.64)}$
DenseUNet	$39.43_{(0.92)}$	$41.32_{(1.21)}$	$35.88_{(0.63)}$	$\mathbf{34.65}_{(058)}$
DLV3+ Eff-B7	$25.27_{(0.69)}$	$25.08_{(0.15)}$	$24.96_{(0.69)}$	$\mathbf{24.76}_{(0.68)}$
U-Net-s Eff-B7	$26.05_{(0.52)}$	$26.74_{(0.33)}$	$25.91_{(0.39)}$	$\mathbf{25.68}_{(0.90)}$
U-Net-s Rn50	$27.05_{(0.78)}$	$29.25_{(1.01)}$	$29.42_{(0.69)}$	$\mathbf{26.39}_{(0.39)}$
U-Net-s Rns269e	$24.87_{(1.17)}$	$25.14_{(0.63)}$	$23.71_{(0.73)}$	$\mathbf{23.40}_{(0.68)}$

C. QUANTITATIVE AND QUALITATIVE RESULTS

This section shows the results of a quantitative comparison of all the models trained with the loss functions: L_{CE} (cross-entropy), L_D (Dice), L_{CV} (Chan-Vese), and $L_{CE+D+CV}$ (combined). The metrics used are the average class values of IoU (Table 2), Dice (Table 3), HD (Table 4), and HD95 (Table 5). The metrics HD and HD95 are expressed in mm and calculated on the basis of the average of the results of each image in the test split, whereas the metrics IoU and Dice are calculated based on the confusion matrix of the entire test split and are expressed in percentage. In the case of the combined loss function, the best results were obtained for values of (10): $\alpha = 0.1 - 0.5$, $\beta = 1$, $\gamma = 5 - 10$.

The values in bold in each table refer to the best results for each model based on the loss function used, and the values in red correspond to the best result in the table. With some exceptions, models trained with the combined loss function produce the best results in all metrics. Table 2 and Table 3 shows that only in the case of the DeepLabV3+ model with MobileNetV2, slightly better mIoU and mDice values are obtained for the loss function L_{CV}. Table 4 indicates that the DeepLabV3+ model with EfficientNet-B7 achieves a better mHD value for the loss function L_{CV}, which is also the best value obtained in the table. In general, the U-Net-scSE model with ResNeSt-269e trained with the combined loss function achieves the best results in all metrics except in mHD95, where it obtains the second-best value. As expected, the baseline models perform worst in all metrics. Furthermore, it is visible that all models with pre-trained encoders achieve successful results when trained only using L_{CV}.

Table 6 shows the quantitative results of the segmentation by class of the models obtained using the combined loss function and based on IoU and HD95. The up arrow (↑) indicates that it is better when it tends to 100%, and the down arrow (↓) when it tends to 0. By grouping the models according to their size, we can classify the first four as lightweight and the last four as heavyweight. U-Net-scSE with ResNeSt-296e performs best in all classes for both metrics for all models. For lightweight models, U-Net-scSE with EfficientNet-B0 achieves better performance for both metrics in all classes except for the BW class, where U-Net-scSE and DeepLabV3+ models with MobileNetV2 obtain a superior performance on IoU and HD95 metrics, respectively.

Figure 6 shows the semantic segmentation results of different lightweight models for the built dataset using the combined loss function, where 6(a) is the original RGB image, 6(b) is the result of manual semantic segmentation or ground truth, 6(c) is the result of semantic segmentation of the DeepLabV3+ with MobileNetV2 model, 6(d) is the result of semantic segmentation of the DeepLabV3+ with EfficientNet-B0 model, 6(e) is the result of semantic segmentation of the U-Net-scSE with MobileNetV2 model, and 6(f) is the result of semantic segmentation of the U-Net-scSE with EfficientNet-B0 model. The red region represents the background water (BW), the black region is the seafloor or obstacle (SO), and the yellow is the fish region. The first row corresponds to an image from the RockFish dataset, and the best result is obtained using U-Net-scSE and EfficientNet-B0. The second row corresponds to an image from the DeepFish dataset, and the best result comes from U-Net-scSE with MobileNet. The third row corresponds to an image from the SUIM dataset, and the best result is obtained using U-Net-scSE with EfficientNet-B0. The fourth and fifth rows correspond to images captured by an underwater drone, and it is observed that in the case of the fourth row, U-Net-scSE with EfficientNet-B0 obtains the best mIoU value, but DeepLabV3+ with EfficientNet-B0 obtains by far a better value of mHD95. In the fifth row, U-Net-scSE with EfficientNet-B0 performs better.

Figure 7 shows the semantic segmentation results of different heavyweight models for the built dataset using the combined loss function, where 6(a) is the original RGB image; 6(b) is the result of manual semantic segmentation or ground truth; 6(c) is the result of semantic segmentation of the U-Net model; 6(d) is the result of semantic segmentation of the DeepLabV3+ with EfficientNet-B7 model; 6(e) is the result of semantic segmentation of the U-Net-scSE with EfficientNet-B7 model; 6(f) is the result of semantic

TABLE 6. Quantitative segmentation results (mean and standard deviation) with intersection-over-union (IoU) and 95% Hausdorff distance (HD95) for all classes on the proposed dataset.

Model	IoU(↑)				HD95(↓)			
	BW	SO	F	**mIoU**	BW	SO	F	**mHD95**
DLV3+ MnV2	$85.71_{(0.28)}$	$92.83_{(0.13)}$	$75.52_{(0.22)}$	$84.69_{(0.09)}$	$\mathbf{28.33}_{(1.11)}$	$20.48_{(0.32)}$	$37.66_{(1.63)}$	$28.59_{(0.65)}$
DLV3+ Eff-B0	$85.05_{(0.41)}$	$92.87_{(0.22)}$	$76.63_{(0.31)}$	$84.85_{(0.27)}$	$29.96_{(1.77)}$	$20.63_{(0.78)}$	$33.64_{(1.27)}$	$27.65_{(0.98)}$
U-Net-s MnV2	$\mathbf{86.23}_{(0.18)}$	$93.26_{(0.11)}$	$77.46_{(0.38)}$	$85.65_{(0.18)}$	$29.26_{(0.74)}$	$20.31_{(0.90)}$	$34.82_{(0.58)}$	$27.63_{(0.55)}$
U-Net-s Eff-B0	$86.12_{(0.39)}$	$\mathbf{93.41}_{(0.15)}$	$\mathbf{78.53}_{(0.23)}$	$\mathbf{86.02}_{(0.19)}$	$29.35_{(1.71)}$	$\mathbf{19.35}_{(0.87)}$	$\mathbf{33.07}_{(1.70)}$	$\mathbf{26.61}_{(0.92)}$
SegNet	$81.00_{(0.72)}$	$89.04_{(0.51)}$	$61.18_{(0.80)}$	$77.08_{(0.10)}$	$43.49_{(1.16)}$	$36.93_{(1.86)}$	$67.20_{(0.90)}$	$49.24_{(0.84)}$
U-Net	$84.82_{(0.28)}$	$92.07_{(0.27)}$	$73.21_{(0.50)}$	$83.37_{(0.35)}$	$34.96_{(0.52)}$	$27.08_{(0.54)}$	$45.34_{(1.39)}$	$35.64_{(0.63)}$
DenseUNet	$83.38_{(0.07)}$	$92.62_{(0.14)}$	$70.71_{(0.54)}$	$81.57_{(0.18)}$	$33.01_{(0.88)}$	$25.12_{(1.10)}$	$45.01_{(0.60)}$	$34.65_{(0.58)}$
DLV3+ Eff-B7	$85.65_{(0.74)}$	$93.48_{(0.21)}$	$79.01_{(0.26)}$	$86.04_{(0.25)}$	$28.34_{(0.76)}$	$17.65_{(1.00)}$	$31.07_{(1.30)}$	$24.76_{(0.68)}$
U-Net-s Eff-B7	$86.22_{(0.62)}$	$93.67_{(0.27)}$	$79.98_{(0.43)}$	$86.62_{(0.31)}$	$30.90_{(1.59)}$	$18.33_{(0.97)}$	$30.72_{(1.03)}$	$25.68_{(0.90)}$
U-Net-s Rn50	$86.41_{(0.33)}$	$93.66_{(0.08)}$	$78.51_{(0.43)}$	$86.20_{(0.27)}$	$27.71_{(1.82)}$	$19.43_{(0.22)}$	$33.42_{(0.99)}$	$26.39_{(0.39)}$
U-Net-s Rns269e	$\mathbf{87.31}_{(0.34)}$	$\mathbf{94.21}_{(0.12)}$	$\mathbf{80.84}_{(0.49)}$	$\mathbf{87.45}_{(0.17)}$	$\mathbf{27.27}_{(1.29)}$	$\mathbf{17.35}_{(1.09)}$	$\mathbf{28.33}_{(1.08)}$	$\mathbf{23.40}_{(0.68)}$

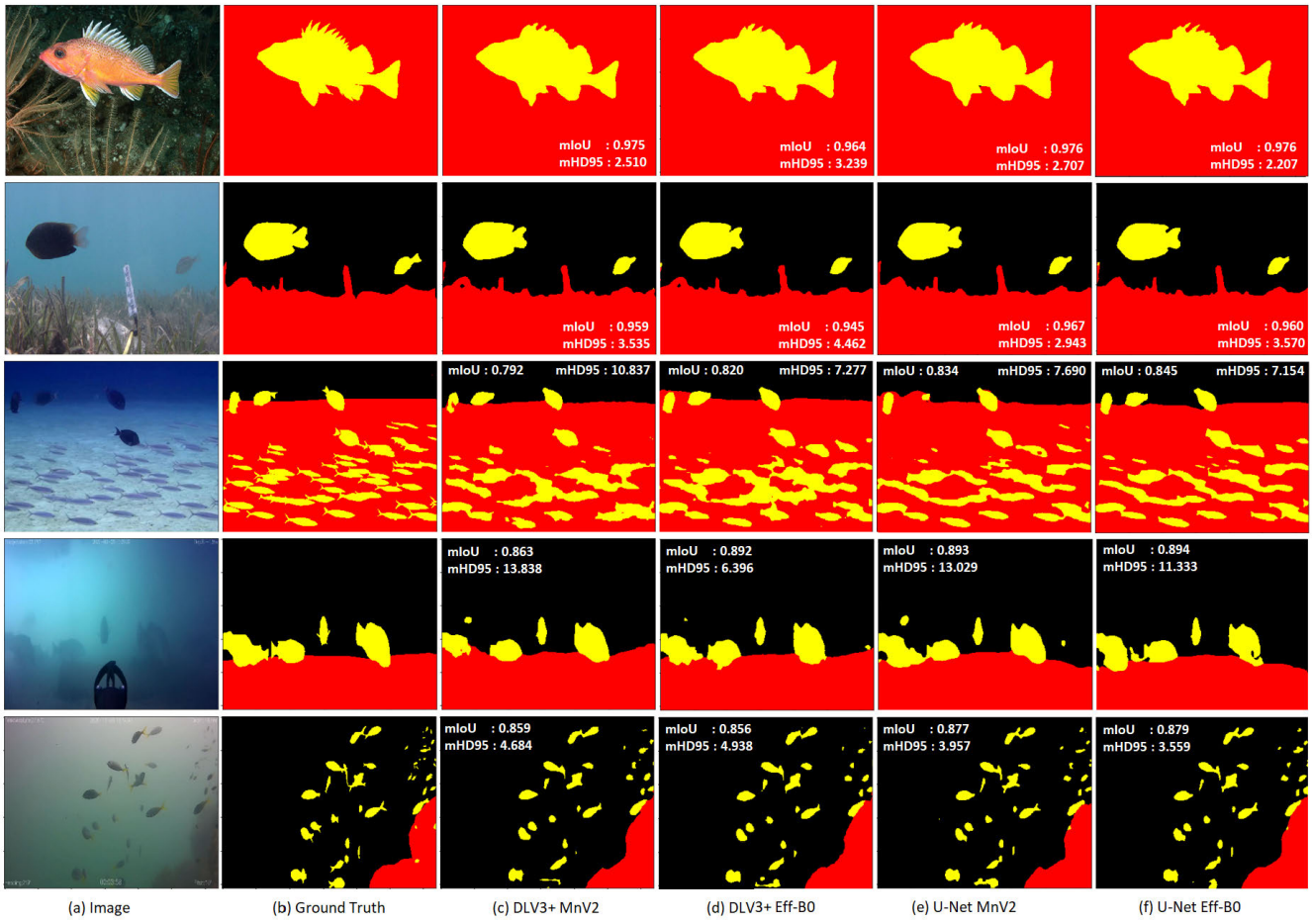

FIGURE 6. Qualitative comparison results of the semantic segmentation with light models.

segmentation of the U-Net-scSE with ResNeSt-296e model. The first row corresponds to the same image as the first row in Figure 6, and U-Net-scSE with ResNeSt-269e gets the best overall result. The same model obtains the best results in the rest of the rows except the fourth, which corresponds to an image captured in a fish farm with many fish and suspended particles close to the drone camera, where the U-Net model obtains a better result. The DeepLabV3+ with EfficientNet-B7 model performs poorly in the fourth row as well, which is confirmed by the high value of 70.3 mm obtained in the mHD95 metric.

Table 7 shows a comparison of the best models trained on our dataset. The results showed that the training time per epoch of the small networks based on MobileNetV2 and EfficientNet-B0 is shorter, whereas large networks require a longer training time. The largest networks based on EfficientNet-B7 and ResNeSt-269e are trained with a batch size of 4 because they require more memory. The

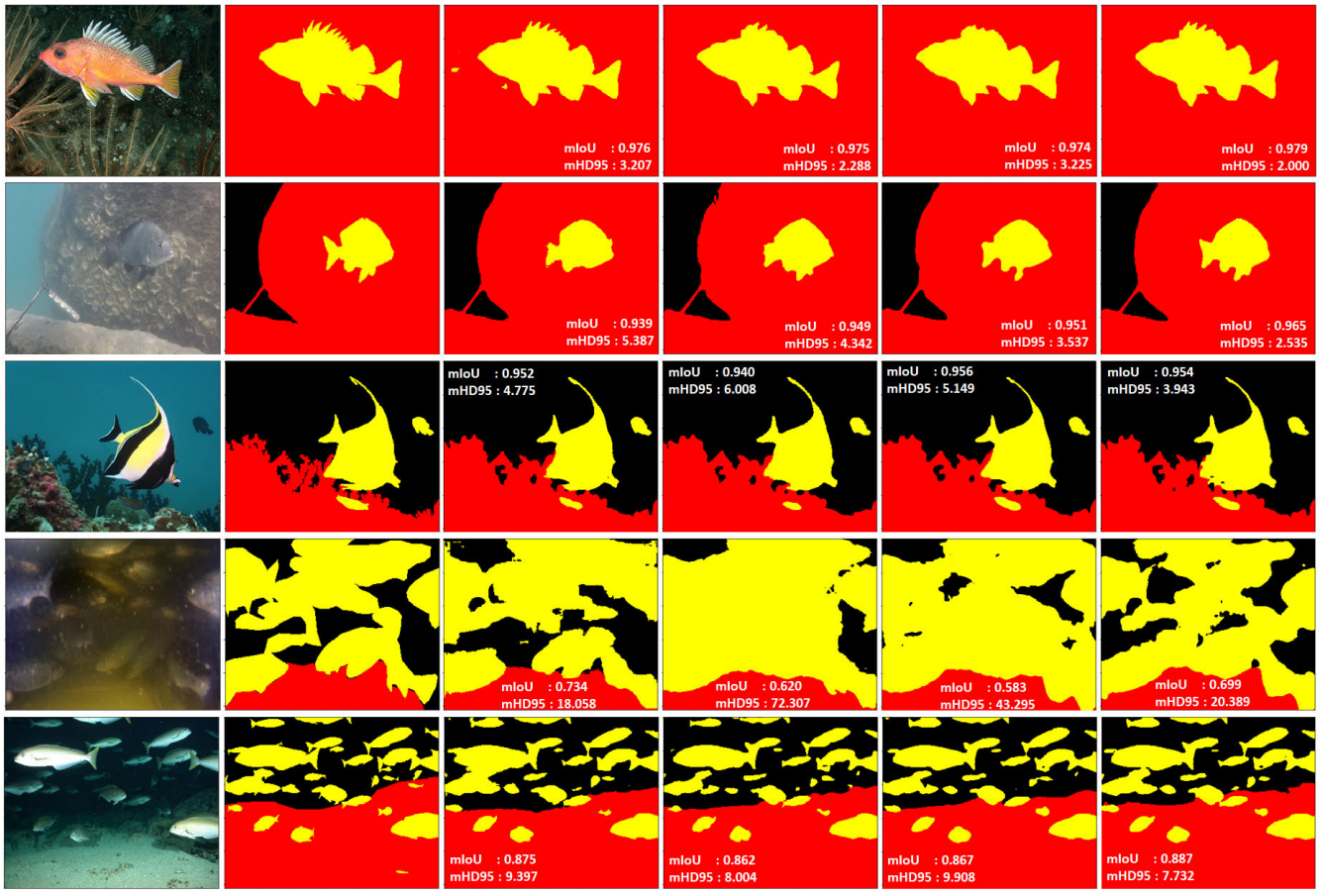

FIGURE 7. Qualitative comparison results of the semantic segmentation with heavy models.

TABLE 7. Evaluation parameters.

Model	Batch Size	Training time (s/epoch)	Inference Time (ms)	# of parameters (M)	Memory size (MB)
DLV3+ MnV2	16	16	11.24	4.38	17.80
DLV3+ Eff-B0	16	20	15.22	4.50	20.00
U-Net-s MnV2	16	16	13.91	6.90	27.90
U-Net-s Eff-B0	16	21	16.89	5.89	25.60
SegNet	16	24	13.11	12.54	50.20
U-Net	16	40	13.68	31.04	124.30
DenseUNet	8	38	28.68	12.37	50.10
DLV3+ Eff-B7	4	170	38.56	63.46	262.10
U-Net-s Eff-B7	4	80	36.15	65.58	270.60
U-Net-s Rn50	16	25	15.23	33.82	136.00
U-Net-s Rns269e	4	150	62.03	119.21	479.00

DeepLabV3+ model with MobileNetV2 as an encoder only requires 17.8 MB of storage space and can perform inference in just 11.24 ms, whereas U-Net with scSE blocks and ResNeSt-269e as the encoder requires 26 times more storage space and is 5 times slower.

D. ABLATION STUDY

To validate the contribution of each component of the combined loss function, ablation experiments are performed using the U-Net-scSE and DeepLabV3+ models with EfficientNet B0 as the backbone. The quantitative results of the experiments are summarized in Table 8.

1) EFFECTIVENESS OF L_{CE} LOSS

Without using an L_{CE} loss, the performance drops slightly on the worked dataset. The results for the DeepLabV3+ model show an average decrease of 0.14% in mIoU and an average increase of 0.44 mm in mHD95. In the case of the U-Net-scSE model, the results show an average decrease of 0.3% in mIoU and an average increase of 0.5 mm in mHD95.

2) EFFECTIVENESS OF L_D LOSS

Without using an L_D loss, performance drops to a greater extent compared to LCE, especially in the IoU metric. The results for the DeepLabV3+ model show an average decrease of 0.37% in mIoU and an average increase of 0.18 mm in mHD95. In the case of the U-Net-scSE model, the results show an average decrease of 0.49% in mIoU and an average increase of 0.87 mm in mHD95.

3) EFFECTIVENESS OF L_{CV} LOSS

Without using an L_{CV} loss, performance decreases to a greater extent in both metrics. This highlights the impor-

TABLE 8. Result of ablation study (mean and standard deviation).

Model	Loss	mIoU (↑)	mHD95 (↓)
DeepLabV3+ Efficientnet-B0	$L_{CE+D+CV}$	**84.85**$_{(0.27)}$	**27.65**$_{(0.98)}$
	L_{CE+CV}	84.54$_{(0.11)}$	27.84$_{(0.62)}$
	L_{D+CV}	84.73$_{(0.05)}$	28.10$_{(0.94)}$
	L_{CE+D}	84.35$_{(0.21)}$	29.03$_{(1.26)}$
U-Net-scSE Efficientnet-B0	$L_{CE+D+CV}$	**86.10**$_{(0.18)}$	**26.53**$_{(0.81)}$
	L_{CE+CV}	85.68$_{(0.23)}$	27.40$_{(0.76)}$
	L_{D+CV}	85.84$_{(0.29)}$	27.03$_{(1.31)}$
	L_{CE+D}	85.08$_{(0.31)}$	27.95$_{(1.10)}$

tance of Chan-Vese based active contour loss and indicates that this component allows for more accurate segmentation results. The results for the DeepLabV3+ model show an average decrease of 0.59% in mIoU and an average increase of 1.38 mm in mHD95. In the case of the U-Net-scSE model, the results show an average decrease of 1.18% in mIoU and an average increase of 1.42 mm in mHD95.

In the worked dataset, the Dice coefficient-based loss function achieved better results compared to the cross-entropy loss function, especially in the mIoU metric. However, there were cases in which L_D did not achieve convergence, specifically when using networks without pre-training such as SegNet. For this reason, we chose to combine the terms L_D and L_{CE}.

E. COMPARISON WITH STATE-OF-THE-ART LEARNED ACTIVE CONTOURS FOR SEMANTIC SEGMENTATION OF FISH

We also compared the L_{CV} loss function with other studies incorporating the use of active contours in deep learning. The experiments were performed on the same dataset but only on the fish class. A comparison was made between the end-to-end approaches of [30], [31], [32] and the post-processing approach of [29]. In both approaches, a CNN and an ACM method are combined, but only [30], [31], [32] include the ACM's parameters as part of the end-to-end approach, theoretically improving the segmentation results. However, they have difficulties handling multi-class problems because of their high complexity. In comparison, the proposed method can manage multi-class segmentation naturally, owing to the design of a loss function based on active contours and level sets. Moreover, it may be easily combined with other loss functions. In the experiments, an exhaustive search was carried out to determine the parameters that generate the best segmentation results for each method. The details of the parameter search procedure can be found in the supplementary materials.

Table 9 shows the quantitative segmentation results of different methods on the worked dataset using DeepLabV3+, whereas Table 10 shows the results for U-Net-scSE, both architectures with EfficientNet-B0 as the backbone. Using the L_{CV} loss function in combination with L_{CE} or L_D outperforms the benchmark methods mainly on the mIoU metric and in most cases on mHD95. In particular, when combining L_{CV} with L_{CE} to train DeepLabV3+, the mIoU and mHD95 metrics improve on average by 1.09% and 0.62 mm, respectively. Whereas for U-Net-scSE, the mIoU and mHD95 metrics improve on average by 1.4% and 0.56 mm, respectively. Similarly, when combining L_{CV} with L_D to train DeepLabV3+, the mIoU and mHD95 metrics improve on average by 0.7% and 1.39 mm, respectively. Whereas for U-Net-scSE, the mIoU and mHD95 metrics improve on average by 0.14% and 1.23 mm, respectively.

TABLE 9. Result of comparison with state-of-the-art learned active contours (mean and standard deviation). (DeepLabV3+ with Efficientnet-B0).

Loss	mIoU (↑)	mHD95 (↓)	Epochs
L_{BCE}	75.93$_{(0.47)}$	20.42$_{(0.75)}$	189
L_{BCE} + ACWE	75.96$_{(0.49)}$	19.90$_{(0.67)}$	189
L_{BCE} + LRACWE [29]	76.08$_{(0.51)}$	**19.44**$_{(1.12)}$	189
L_{BCE} + SBACWE	75.99$_{(0.50)}$	20.29$_{(1.05)}$	189
$L_{(BCE+ACWE)}$ [30]	76.25$_{(0.24)}$	20.12$_{(0.95)}$	218
$L_{(BCE+LRACWE)}$ [31]	76.31$_{(0.28)}$	19.52$_{(0.92)}$	230
$L_{(BCE+SBACWE)}$ [32]	76.36$_{(0.26)}$	19.66$_{(0.78)}$	180
$L_{(BCE+CV)}$	**76.76**$_{(0.38)}$	19.80$_{(0.80)}$	262
L_D	76.20$_{(0.46)}$	21.37$_{(1.53)}$	327
L_D + ACWE	0.76$_{(0.45)}$	21.11$_{(1.76)}$	327
L_D + LRACWE [29]	76.22$_{(0.46)}$	21.36$_{1.50)}$	327
L_D + SBACWE	76.24$_{(0.47)}$	22.37$_{(1.41)}$	327
$L_{(D+ACWE)}$ [30]	76.46$_{(0.18)}$	20.36$_{(0.88)}$	309
$L_{(D+LRACWE)}$ [31]	76.72$_{(0.20)}$	**19,95**$_{(0.87)}$	347
$L_{(D+SBACWE)}$ [32]	76.56$_{(0.40)}$	21.29$_{(0.60)}$	336
$L_{(D+CV)}$	**76.73**$_{(0.32)}$	19.98$_{(1.04)}$	311

TABLE 10. Result of comparison with state-of-the-art learned active contours (mean and standard deviation). (U-Net-scSE with Efficientnet-B0).

Loss	mIoU (↑)	mHD95 (↓)	Epochs
L_{BCE}	77.38$_{(0.42)}$	19.76$_{(1.28)}$	143
L_{BCE} + ACWE	77.46$_{(0.43)}$	19.21$_{(1.15)}$	143
L_{BCE} + LRACWE [29]	77.46$_{(0.42)}$	19.31$_{(1.28)}$	143
L_{BCE} + SBACWE	77.50$_{(0.42)}$	19.64$_{(0.98)}$	143
$L_{(BCE+ACWE)}$ [30]	77.51$_{(0.35)}$	19.30$_{(1.15)}$	160
$L_{(BCE+LRACWE)}$ [31]	77.65$_{(0.26)}$	19,87$_{(1.50)}$	150
$L_{(BCE+SBACWE)}$ [32]	77.47$_{(0.30)}$	19,73$_{(0.86)}$	118
$L_{(BCE+CV)}$	**78.46**$_{(0.28)}$	**19.20**$_{(1.13)}$	206
L_D	78.65$_{(0.37)}$	20.40$_{(0.66)}$	285
L_D + ACWE	78.66$_{(0.36)}$	20.32$_{(0.64)}$	285
L_D + LRACWE [29]	78.65$_{(0.37)}$	20.26$_{(0.54)}$	285
L_D + SBACWE	78.37$_{(0.33)}$	20.97$_{(0.55)}$	285
$L_{(D+ACWE)}$ [30]	78.61$_{(0.26)}$	19.73$_{(0.58)}$	425
$L_{(D+LRACWE)}$ [31]	78.72$_{(0.41)}$	19.92$_{(0.80)}$	283
$L_{(D+SBACWE)}$ [32]	78.52$_{(0.36)}$	19.87$_{(0.77)}$	224
$L_{(D+CV)}$	**78.76**$_{(0.23)}$	**19.17**$_{(0.76)}$	260

V. DISCUSSION

Given that 50% of our dataset is composed of relabeled images from [22], we use the fact that a simple U-Net architecture achieves the best results in its experiments as a reference point. The best results were obtained by testing an improved U-Net with a pre-trained ResNest296e encoder

path and including scSE blocks on each decoder layer. Our results are consistent with [44], indicating that ResNest296e outperforms EfficientNet-B7 as the backbone of U-Net-scSE, but with a longer time of inference. As a result, we believe that the attention mechanisms described in [38] and [44] can be a critical component of neural networks performing semantic segmentation tasks.

We successfully evaluated the proposed loss function using underwater image data sets containing fish of various sizes and shapes in various habitats. The proposed combined loss function takes advantage of the cross-entropy loss's discriminatory capacity, the Dice loss's ability to handle class imbalance, and the potential to refine the spatial details of the learned active contours based on the Chan-Vese functional. The results of the models evaluated show that our combined loss function outperforms traditional loss functions (L_{CE} and L_D). These results also demonstrate the good performance of the lost function based on active contours and level sets (L_{CV}). In this regard, the results of the ablation study confirm the significant contribution of this component to the proposed combined loss function. Furthermore, we compare the approach of including active contours in the loss function (L_{CV}) with other SOTA approaches such as post-processing and extreme to extreme, obtaining comparable results with the advantage of being easily extendable to multi-class segmentation problems [33], [52], without requiring a post-processing stage and not being sensitive to an ACM method's selection parameters.

VI. CONCLUSION

This work addresses the problem of semantic segmentation of underwater environments for autonomous navigation tasks and fish monitoring using convolutional neural networks. To perform the experiments, we built a dataset that contains 1824 images with pixel annotations for three classes: background water, seafloor/obstacles, and fish.

A combined loss function is used to train the models by incorporating active contour theory and level set approaches into modern CNNs to simultaneously learn local appearances and spatial information of ground truth. Experiment results with SOTA networks evaluated in the dataset confirm the performance of this loss function. We also observed that a U-Net architecture that incorporates attention mechanisms in the decoder layers using scSE blocks outperforms the complex DeepLabV3+ architecture. Combining U-Net-scSE with a pre-trained EfficientNet-B0 lightweight architecture gives a value of 86% mIoU and 26.61 mm of mHD95. By combining U-Net-scSE with heavy architectures such as EfficientNet-B7 and ResNeSt-296e, the advantages of this updated version of ResNet were verified, obtaining a value of 87.45% for mIoU and 23.40 mm for mHD95.

Future work will focus on the inclusion of loss functions based on active contour theory in semi-supervised and unsupervised learning approaches to reduce the laborious task of labeling.

REFERENCES

[1] United Nations. *Oceans*. Accessed: Nov. 30, 2021. [Online]. Available: https://www.un.org/sustainabledevelopment/es/oceans/

[2] R. Danovaro, L. Carugati, M. Berzano, A. E. Cahill, S. Carvalho, A. Chenuil, and C. Corinaldesi, "Implementing and innovating marine monitoring approaches for assessing marine environmental status," *Frontiers Mar. Sci.*, vol. 3, p. 213, Nov. 2016, doi: 10.3389/fmars.2016.00213.

[3] F. Liu and M. Fang, "Semantic segmentation of underwater images based on improved deeplab," *J. Mar. Sci. Eng.*, vol. 8, no. 3, p. 188, Mar. 2020, doi: 10.3390/jmse8030188.

[4] J. Niemeijer, P. Pekezou Fouopi, S. Knake-Langhorst, and E. Barth, "A review of neural network based semantic segmentation for scene understanding in context of the self driving car," in *Proc. Student Conf. Med. Eng. Sci.*, 2017, pp. 1–4.

[5] F. Khan, M. Khan, N. Iqbal, S. Khan, D. M. Khan, A. Khan, and D.-Q. Wei, "Prediction of recombination spots using novel hybrid feature extraction method via deep learning approach," *Frontiers Genet.*, vol. 11, Sep. 2020, Art. no. 539227, doi: .10.3389/fgene.2020.539227.

[6] S. Khan, M. Khan, N. Iqbal, M. Li, and D. M. Khan, "Spark-based parallel deep neural network model for classification of large scale RNAs into piRNAs and non-piRNAs," *IEEE Access*, vol. 8, pp. 136978–136991, 2020, doi: 10.1109/ACCESS.2020.3011508.

[7] N. Inayat, M. Khan, N. Iqbal, S. Khan, M. Raza, D. M. Khan, A. Khan, and D. Q. Wei, "IEnhancer-DHF: Identification of enhancers and their strengths using optimize deep neural network with multiple features extraction methods," *IEEE Access*, vol. 9, pp. 40783–40796, 2021, doi: 10.1109/ACCESS.2021.3062291.

[8] I. Ali, A. U. Rehman, D. M. Khan, Z. Khan, M. Shafiq, and J.-G. Choi, "Model selection using K-means clustering algorithm for the symmetrical segmentation of remote sensing datasets," *Symmetry*, vol. 14, no. 6, p. 1149, Jun. 2022, doi: 10.3390/sym14061149.

[9] R. Prados, R. Garcia, N. Gracias, L. Neumann, and H. Vagstol, "Real-time fish detection in trawl nets," in *Proc. OCEANS-Aberdeen*, Jun. 2017, pp. 1–5, doi: 10.1109/OCEANSE.2017.8084760.

[10] L. Fang, X. Wang, and L. Wang, "Multi-modal medical image segmentation based on vector-valued active contour models," *Inf. Sci.*, vol. 513, pp. 504–518, Mar. 2020, doi: 10.1016/j.ins.2019.10.051.

[11] Z. Zhang, C. Duan, T. Lin, S. Zhou, Y. Wang, and X. Gao, "GVFOM: A novel external force for active contour based image segmentation," *Inf. Sci.*, vol. 506, pp. 1–18, Jan. 2020, doi: 10.1016/j.ins.2019.08.003.

[12] W. Wang, Y. Wang, Y. Wu, T. Lin, S. Li, and B. Chen, "Quantification of full left ventricular metrics via deep regression learning with contour-guidance," *IEEE Access*, vol. 7, pp. 47918–47928, 2019, doi: 10.1109/ACCESS.2019.2907564.

[13] W. Shen, W. Xu, H. Zhang, Z. Sun, J. Ma, X. Ma, S. Zhou, S. Guo, and Y. Wang, "Automatic segmentation of the femur and tibia bones from X-ray images based on pure dilated residual U-Net," *Inverse Problems Imag.*, vol. 15, no. 6, p. 1333, 2021, doi: 10.3934/ipi.2020057.

[14] H. Zhang, W. Zhang, W. Shen, N. Li, Y. Chen, S. Li, B. Chen, S. Guo, and Y. Wang, "Automatic segmentation of the cardiac MR images based on nested fully convolutional dense network with dilated convolution," *Biomed. Signal Process. Control*, vol. 68, Jul. 2021, Art. no. 102684, doi: 10.1016/j.bspc.2021.102684.

[15] P. W. Patil, O. Thawakar, A. Dudhane, and S. Murala, "Motion saliency based generative adversarial network for underwater moving object segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1565–1569, doi: 10.1109/ICIP.2019.8803091.

[16] C. Premachandra, S. Ueda, and Y. Suzuki, "Detection and tracking of moving objects at road intersections using a 360-degree camera for driver assistance and automated driving," *IEEE Access*, vol. 8, pp. 135652–135660, 2020, doi: 10.1109/ACCESS.2020.3011430.

[17] R. Garcia, R. Prados, J. Quintana, A. Tempelaar, N. Gracias, and S. Rosen, "Automatic segmentation of fish using deep learning with application to fish size measurement," *ICES J. Mar. Sci.*, vol. 77, no. 4, 2020, Art. no. 13541366, doi: 10.1093/icesjms/fsz186.

[18] A. King, S. M. Bhandarkar, and B. M. Hopkinson, "A comparison of deep learning methods for semantic segmentation of coral reef survey images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1475–14758, doi: 10.1109/CVPRW.2018.00188.

[19] A. King, S. Bhandarkar, and B. Hopkinson, "Deep learning for semantic segmentation of coral reef images using multi-view information," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, Jun. 2019, pp. 1–10.

[20] I. Alonso, A. Cambra, A. Munoz, T. Treibitz, and A. C. Murillo, "Coral-segmentation: Training dense labeling models with sparse ground truth," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2874–2882, doi: 10.1109/ICCVW.2017.339.

[21] I. Alonso, M. Yuval, G. Eyal, T. Treibitz, and A. C. Murillo, "CoralSeg: Learning coral segmentation from sparse annotations," *J. Field Robot.*, vol. 36, no. 8, Art. no. 14561477, 2019, doi: 10.1002/rob.21915.

[22] M. J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S. S. Enan, and J. Sattar, "Semantic segmentation of underwater imagery: Dataset and benchmark," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 1769–1776, doi: 10.1109/IROS45743.2020.9340821.

[23] P. Drews-Jr, I. D. Souza, I. P. Maurell, E. V. Protas, and S. S. C. Botelho, "Underwater image segmentation in the wild using deep learning," *J. Brazilian Comput. Soc.*, vol. 27, no. 1, p. 12, Dec. 2021, doi: 10.1186/s13173-021-00117-7.

[24] A. Saleh, I. H. Laradji, D. A. Konovalov, M. Bradley, D. Vazquez, and M. Sheaves, "A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis," *Sci. Rep.*, vol. 10, no. 1, p. 14671, Sep. 2020, doi: 10.1038/s41598-020-71639-x.

[25] B. Arain, C. McCool, P. Rigby, D. Cagara, and M. Dunbabin, "Improving underwater obstacle detection using semantic image segmentation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9271–9277, doi: 10.1109/ICRA.2019.8793588.

[26] M. J. Islam, S. Sakib Enan, P. Luo, and J. Sattar, "Underwater image super-resolution using deep residual multipliers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 900–906, doi: 10.1109/ICRA40945.2020.9197213.

[27] M. J. Islam, Y. Xia, and J. Sattar, "Fast underwater image enhancement for improved visual perception," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3227–3234, Apr. 2020, doi: 10.1109/LRA.2020.2974710.

[28] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[29] A. Hatamizadeh, A. Hoogi, D. Sengupta, W. Lu, B. Wilcox, D. Rubin, and D. Terzopoulos, "Deep active lesion segmentation," in *Proc. Mach. Learn. Med. Imag., 10th Int. Workshop Mach. Learn. Med. Imag. (MLMI)*, 2019, pp. 98–105, doi: 10.1007/978-3-030-32692-0_12.

[30] A. Hatamizadeh, D. Sengupta, and D. Terzopoulos, "End-to-end deep convolutional active contours for image segmentation," 2019, *arXiv:1909.13359*.

[31] A. Hatamizadeh, D. Sengupta, and D. Terzopoulos, "End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery," in *Computer Vision–(ECCV)*. Cham, Switzerland: Springer, 2020, pp. 730–746, doi: 10.1007/978-3-030-58610-2_43.

[32] M. Zhang, B. Dong, and Q. Li, "Deep active contour network for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–(MICCAI)*. Cham, Switzerland: Springer, 2020, pp. 321–331, 2020, doi: 10.1007/978-3-030-59719-1_32.

[33] Y. Kim, S. Kim, T. Kim, and C. Kim, "CNN-based semantic segmentation using level set loss," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1752–1760, doi: 10.1109/WACV.2019.00191.

[34] B. Kim and J. C. Ye, "Mumford–Shah loss functional for image segmentation with deep learning," in *IEEE Trans. Image Process.*, vol. 29, pp. 1856–1866, 2020, doi: 10.1109/TIP.2019.2941265.

[35] Y. Lan, Y. Xiang, and L. Zhang, "An elastic interaction-based loss function for medical image segmentation," 2020, *arXiv:2007.02663*.

[36] X. Chen, X. Luo, G. Wangy, and Y. Zhengy, "Deep elastica for image segmentation," in *Proc. IEEE 18th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2021, pp. 706–710, doi: 10.1109/ISBI48211.2021.9433886.

[37] J. Ma, J. He, and X. Yang, "Learning geodesic active contours for embedding object global information in segmentation CNNs," *IEEE Trans. Med. Imag.*, vol. 40, no. 1, pp. 93–104, Jan. 2021, doi: 10.1109/TMI.2020.3022693.

[38] T. H. N. Le, K. G. Quach, K. Luu, C. N. Duong, and M. Savvides, "Reformulating level sets as deep recurrent neural network approach to semantic segmentation," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2393–2407, May 2018, doi: 10.1109/TIP.2018.2794205.

[39] G. Cutter, K. Stierhoff, and J. Zeng, "Automated detection of rockfish in unconstrained underwater videos using Haar cascades and a new image dataset: Labeled fishes in the wild," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops*, Jan. 2015, pp. 57–62, doi: 10.1109/WACVW.2015.11.

[40] N. A. Nezla, T. P. Mithun Haridas, and M. H. Supriya, "Semantic segmentation of underwater images using UNet architecture based deep convolutional encoder decoder model," in *Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2021, pp. 28–33, doi: 10.1109/ICACCS51430.2021.9441804.

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[42] A. G. Roy, N. Navab, and C. Wachinger, "Recalibrating fully convolutional networks with spatial and channel 'squeeze and excitation' blocks," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 540–549, Feb. 2019, doi: 10.1109/TMI.2018.2867261.

[43] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[44] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[45] *ImageNet*. Accessed: Dec. 4, 2021. [Online]. Available: http://www.image-net.org/

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," 2014, *arXiv:1409.0575*.

[47] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[48] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.

[49] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-attention networks," 2020, *arXiv:2004.08955*.

[50] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, doi: 10.1109/3DV.2016.79.

[51] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001, doi: 10.1109/83.902291.

[52] X. Chen, B. M. Williams, S. R. Vallabhaneni, G. Czanner, R. Williams, and Y. Zheng, "Learning active contour models for medical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11624–11632, doi: 10.1109/CVPR.2019.01190.

[53] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," in *Proc. 2nd Int. Conf. Med. Imag. Deep Learn.*, 2019, pp. 285–296, doi: 10.1016/j.media.2020.101851.

[54] M. Harouni and H. Yazdani Baghmaleki, "Color image segmentation metrics," 2020, *arXiv:2010.09907*.

[55] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?" in *Proc. Brit. Mach. Vis. Conf.*, 2013, p. 5244.

[56] J. Ribera, D. Güera, Y. Chen, and E. J. Delp, "Locating objects without bounding boxes," 2018, *arXiv:1806.07564*.

[57] I. Rizwan I Haque and J. Neubert, "Deep learning approaches to biomedical image segmentation," *Informat. Med. Unlocked*, vol. 18, 2020, Art. no. 100297, doi: 10.1016/j.imu.2020.100297.

[58] P. Yakubovskiy. (2020). *Segmentation Models Pytorch*. [Online]. Available: https://github.com/qubvel/segmentation_models.pytorch

[59] S. Malacrino. (2020). *DenseUNet in PyTorch*. [Online]. Available: https://github.com/stefano-malacrino/DenseUNet-pytorch

MIGUEL CHICCHON received the B.S. degree in electronic engineering from Universidad Nacional de Ingeniería (UNI), Lima, Peru, in 2009, and the M.S. degree in informatics with a major in computer science from Pontificia Universidad Católica del Perú (PUCP), Lima, in 2019, where he is currently pursuing the Ph.D. degree in engineering.

From 2009 to 2010, he conducted research in satellite systems at the Information Technology and Communications Center (CTIC) and the National Institute for Telecommunications Research and Training (INICTEL). Since 2019, he has been a Consultant for applied research projects with the Institute of Scientific Research of Universidad de Lima (IDIC-ULIMA). His research interests include machine learning, deep learning, reinforcement learning, computer vision, image processing, the Internet of Things, embedded systems, and control systems.

HECTOR BEDON received the B.Sc. degree in physics and the M.Sc. degree in telematic engineering from the National University of Engineering, Peru, in 2003, and the Ph.D. degree in telematic systems engineering from Universidad Politecnica de Madrid (UPM), Spain, in 2016.

He has taught and conducted research with Universidad Nacional de Ingenieria (UNI), Universidad de Lima, and UPM. Since 2022, he is a Professor at the School of Technical Sciences and Engineering of Madrid Open University (UDIMA). His research interests include the development of new internet services and communication protocols for cubesats networks, Internet of Things and artificial intelligence applied to smart cities, smart fishing, and precision agriculture using small satellites, aerial drones, amphibians, and airships.

CARLOS R. DEL-BLANCO received the Telecommunication Engineering and Ph.D. degrees in telecommunication from Universidad Politécnica de Madrid (UPM), in 2005 and 2011, respectively.

Since 2005 he has been a member of the Image Processing Group, UPM. In addition, since 2011, he has been a member of the Faculty of E.T.S. Ingenieros de Telecomunicación, and since 2021, he has been a Professor of signal theory and communications with the Department of Signals, Systems, and Communications. He has been actively involved in European projects and national projects in Spain. His professional interests include signal and image processing, computer vision, pattern recognition, machine learning, and stochastic dynamic models.

IVAN SIPIRAN received the Ph.D. degree in computer science from the University of Chile.

He was a Postdoctoral Researcher with the University of Konstanz, in 2014 and 2015, and a Professor with the Department of Engineering, Pontifical Catholic University, Peru, from 2015 to 2020. Since 2020, he has been an Assistant Professor with the Department of Computer Science, University of Chile. His current research interests include geometry processing, 3-D computer vision, and applications of computer vision in cultural heritage.