CoralSeg: Learning coral segmentation from sparse annotations

Iñigo Alonso, Matan Yuval, Gal Eyal, Tali Treibitz & Ana C. Murillo

Abstract

Robotic advances and developments in sensors and acquisition systems facilitate the collection of survey data in remote and challenging scenarios. Semantic segmentation, which attempts to provide per-pixel semantic labels, is an essential task when processing such data. Recent advances in deep learning approaches have boosted this task's performance. Unfortunately, these methods need large amounts of labeled data, which is usually a challenge in many domains. In many environmental monitoring instances, such as the coral reef example studied here, data labeling demands expert knowledge and is costly. Therefore, many data sets often present scarce and sparse image annotations or remain untouched in image libraries. This study proposes and validates an effective approach for learning semantic segmentation models from sparsely labeled data. Based on augmenting sparse annotations with the proposed adaptive superpixel segmentation propagation, we obtain similar results as if training with dense annotations, significantly reducing the labeling effort. We perform an in-depth analysis of our labeling augmentation method as well as of different neural network architectures and loss functions for semantic segmentation. We demonstrate the effectiveness of our approach on publicly available data sets of different real domains, with the emphasis on underwater scenarios—specifically, coral reef semantic segmentation. We release new labeled data as well as an encoder trained on half a million coral reef images, which is shown to facilitate the generalization to new coral scenarios.

1 INTRODUCTION

Advances in robotics have facilitated the acquisition of data in challenging environments, such as underwater (Bryant et al., **2017**; González-Rivero et al., **2014**) and aerial (Koh & Wich, **2012**) surveys. In particular, visual sensors are a widely used tool that requires little expertise to produce massive data sets. Effectively, researchers are able to rapidly document large areas with high-resolution images, shifting the bottleneck in wide-scale ecological research and monitoring toward image analysis over image acquisition. When done manually, the extraction of useful data from these collections is an onerous task, which urgently demands new solutions and automation.

Semantic image segmentation is the task of automatically providing a complete understanding of scenes captured in images. The impressive development of deep neural networks, especially convolutional neural networks (CNNs; Garcia-Garcia, Orts-Escolano, Oprea, Villena-Martinez, & Garcia-Rodriguez, **2017**), has led to a significant improvement in semantic segmentation approaches in recent years. Many robotic applications benefited from these improvements, for example, autonomous driving (Luc, Neverova, Couprie, Verbeek, & LeCun, **2017**) and object detection and manipulation (Wong et al., **2017**). For training, however, these methods require extensive amounts of pixel-level labeled data. Dense pixel-level annotation is time-consuming and often requires specific expertise, making the labeling process highly expensive and limited in domains that could benefit from it significantly, such as survey tasks (Beijbom et al., **2016**; Venkitasubramanian, Tuytelaars, & Moens, **2016**). For instance, there is abundant underwater monitoring data in the CoralNet project (Beijbom, Edmunds, Kline, Mitchell, & Kriegman, **2012**), from many different locations, labeled by marine biology experts. Yet, each image is only sparsely labeled, having on average 50–200 labeled pixels. Here we suggest a novel approach of *learning dense labeling from sparse labels* (Figure **1**). It enables the application of recent developments in deep learning for semantic segmentation in a wider range of domains including coral segmentation demonstrated here. Many other monitoring applications, such as traffic or agricultural monitoring (Milioto, Lottes, & Stachniss, **2018**) will also be able to reap the benefits of this study.

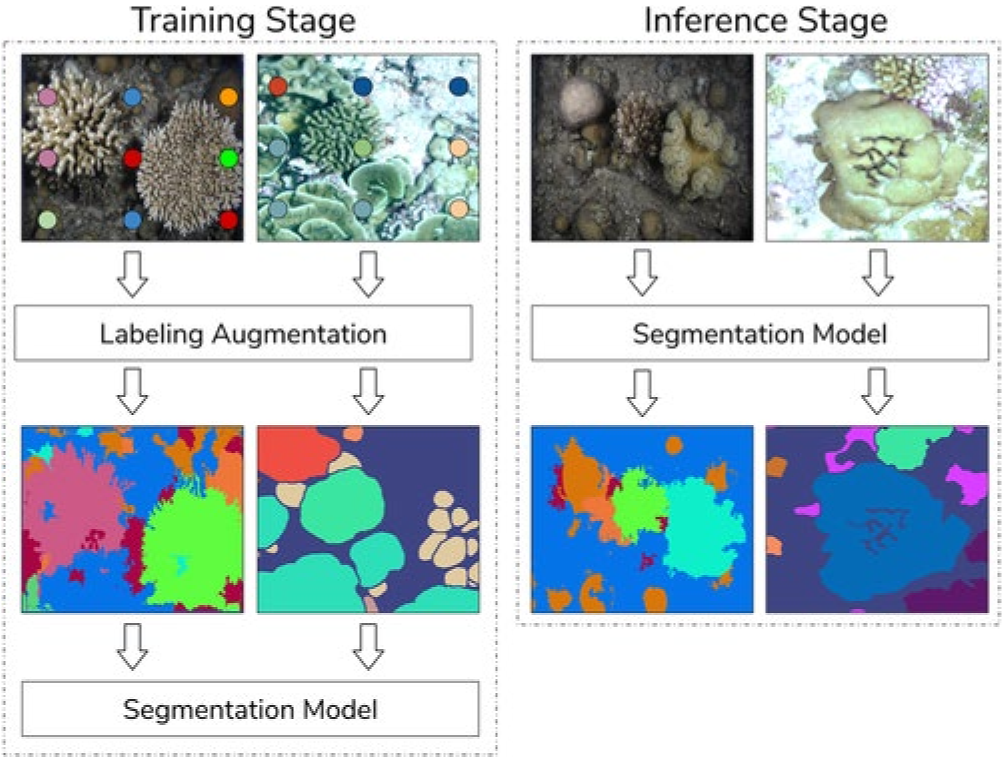


Figure 1

Training semantic segmentation models from sparse labels. In the training stage, we demonstrate how to augment the sparse labels into fully labeled (dense) images, which are used to train the semantic segmentation model. This model is used in later inference stages to obtain dense segmentation of new input images without any supervision. Our pipeline, requiring a much lower labeling effort than prior work, enables effective training of semantic segmentation models

The oceanic underwater environment has remained severely overlooked despite the fact that the ocean covers 71% of the worlds' surface (Visbeck, **2018**). Coral reefs are among the most important marine habitats, occupying an important portion of the ocean, and hosting a substantial amount of all known marine species (Reaka-Kudla, **1997**). Most reef-building corals are colonial organisms of the phylum Cnidaria. Their growth creates epic structures that can be seen from space. These structures not only harbor some of the world's most diverse ecosystems but also provide valuable services and goods such as shoreline protection, habitat maintenance, seafood products, recreation, and tourism. Furthermore, due to their immobility, corals have developed an arsenal of chemical substances that hold great medicinal potential (Cesar, **2000**; Hoegh-Guldberg et al., **2007**; Moberg & Folke, **1999**; Reaka-Kudla, **1997**).

Today, coral reefs face severe threats as a result of climate change and anthropogenic-related stress (Hughes et al., **2017**). Ocean acidification, rising sea surface temperature, overfishing, eutrophication, sedimentation (Fabricius, **2005**), and pollution (Anthony, **2016**; P.-Y. Chen, Chen, Chu, & McCarl, **2015**; Hoegh-Guldberg et al., **2007**; Roberts et al., **2002**) are only a few examples of these menaces. Coral reef ecosystems have suffered massive declines over the past decades, resulting in a marine environmental crisis (Hughes et al., **2003, 2018**).

In coral reefs, dynamics occur over many spatial scales that range from millimeters to kilometers, and the zonation and growth of dominant species form salient patterns (Goreau, **1959**; Huston, **1985**). Studying the complex biological systems together with the structures modified by coral growth and decay remains a challenge in coral reef studies (Stoddart, **1969**). In fact, this hurdle stresses the need for a cross-scale, highly automated approach.

The specific challenges in coral recognition from benthic images are linked directly to the difficulties in underwater imaging and the adaptable nature of corals expressed in their exceptional phenotypic plasticity. Underwater images suffer from color distortion and low contrast. The color of an object imaged underwater varies with distance and the water's optical properties, depending on depth and water type. These dependencies are wavelength-specific, making the color in underwater images an unstable source of information (Akkaynak & Treibitz, **2018**; D. Berman, Treibitz, & Avidan, **2017**), unless corrected (Akkaynak & Treibitz, **2019**). Scleractinian corals are known to display morphological plasticity, that is intraspecific variations in the shape and form of colonial units (Todd, **2008**). These variations represent the feedback between the organism's developmental plan and the surrounding ecological context and settings (Schlichting & Pigliucci, **1998**). They are governed by biotic and abiotic factors such as interspecific interactions, and light regimes along a depth gradient (Eyal et al., **2015**)—all of which make automated image labeling difficult. Moreover, the overall community structure of coral reef assemblages varies greatly, spatially, and temporally (Connell et al., **2004**; Hennige et al., **2010**). Depth-related zonation, a predominant characteristic of coral reefs (Huston, **1985**), also adds to the challenge. Such dissimilarities must be taken into consideration in benthic image analysis, and highlight the need for an adaptive identification tool that is robust to different underwater scenes and can be utilized across an assortment of data sets.

To address this shortcoming, several tools were developed for the annotation of marine images and videos (Gomes-Pereira et al., **2016**). Although some of these offer point predictions (Beijbom et al., **2012**) and area measurements (Kohler & Gill, **2006**), to the best of our knowledge, none possess the novel capabilities of our suggested framework: to learn semantic segmentation from sparse annotations through adaptive labeling augmentation.

To conclude, semantic segmentation represents a leap-forward in benthic image analysis as it not only provides partial presence/absent data but also allows measurement of morphological attributes such as size-frequency distribution of key groups across an image set and observation of wide-scale patterns with minimal labeling effort. As underwater images present one of the hardest use cases for image analysis, our methodology can be adapted easily to a terrestrial setting such as drone-image analysis.

Our experimental results demonstrate that the proposed augmentation of sparse labeling, despite being less accurate than manual annotation, provides valuable and effective information to train a state-of-the-art segmentation model. The results are comparable to approaches trained on densely labeled images, while having the advantage of less intensive annotation requirements. The results also show how different losses for semantic segmentation and architectures affect the different important metrics for semantic segmentation. The presented encoder trained on CoralNet data (half a million images) enhances the semantic segmentation models when fine-tuning the training. This is a similar concept to that of ImageNet (Deng et al., **2009**) but specific to coral reef images. We show how this encoder helps semantic segmentation models learn more general and better features for coral images. Finally, the experimental results show that our method can be applied to and provide the same benefits for other domains, increasing the number of robotic applications that can benefit from it.

The specific contributions of this study are

- A *novel sparse label augmentation method* that enables training dense semantic segmentation models with sparse input labels, providing similar results to those obtained when training with dense labels. This is particularly significant for many ecological applications since most expert labeling efforts consist of sparse labels, and manual dense labeling of many images is essentially infeasible. To demonstrate the general applicability of the proposed strategy, we include experiments on different data sets from other domains captured by different robotic platforms (aerial and urban scenarios).
- We train and release a *generic encoder* for *coral imagery*, trained on over half a million coral reef images. Our experiments demonstrate that this model has learned generic representations for coral imagery that help to learn segmentation models for new specific scenarios with few labeled samples available.
- In addition to the generic model, we make available the new data and developed tools.
- A comparison of different well-known deep learning architectures for semantic segmentation applied to underwater coral reef imagery. We cover not only architectures but also common loss functions and propose a new, more suitable variation of the cross-entropy loss for this problem.

2 RELATED WORK

This section discusses work from areas most relevant to ours: methods for and state-of-the-art of semantic segmentation with special attention on underwater imagery segmentation and strategies to deal with a lack of the required training data, that is, sparse or weak labels.

2.1 Semantic image segmentation

Semantic segmentation is a visual recognition problem consisting of assigning a semantic label to each pixel in the image. The state-of-the-art in this task is currently achieved by solutions based on deep learning, most of them proposing different variations of fully convolutional networks (FCNs; Chen, Papandreou, Schroff, & Adam, **2017**; Chen, Zhu, Papandreou, Schroff, & Adam, **2018**; Jégou, Drozdzal, Vazquez, Romero, & Bengio, **2017**; Long, Shelhamer, & Darrell, **2015**). Some existing solutions for semantic segmentation target instance-level semantic segmentation, for example, Mask-RCNN (He, Gkioxari, Dollár, & Girshick, **2017**), which includes three main steps: region proposal, binary segmentation, and classification. Other solutions, such as DeepLabv3+ (L.-C. Chen et al., **2018**), target class-level semantic segmentation. DeepLabv3+ is a top-performing CNN for semantic segmentation and the base architecture of our work.

Before the surge of deep learning approaches, several algorithms based on superpixel segmentation techniques (Stutz, Hermans, & Leibe, **2018**) were used for this task. These approaches cluster image pixels into several groups of similar and connected pixels (i.e., superpixels). Such approaches classify the superpixels or a superpixel-based labeling propagation (Mičušík & Košecká, **2010**; Tighe & Lazebnik, **2010**). The survey by Zhu, Meng, Cai, and Lu (**2016**) of image segmentation provides a detailed compilation of more conventional solutions for semantic segmentation. A later survey (Garcia-Garcia et al., **2017**) presents a discussion of more recent deep learning-based approaches for semantic segmentation, ranging from new architectures to common data sets. Our work exploits both types of approaches. As we discuss later, while the CNN-based models are the core of our segmentation process, we show that the superpixels are very effective in augmenting sparse labels.

2.1.1 Coral reef community structure analysis

Community ecology is the field that studies the interactions of species that co-occur in space and time (Morin, **2009**). Diversity, a broad term that describes the numerical composition of species, is a feature of ecological communities (Sanders, **1968**). Here, we focus on coral reef communities; the Macro-benthos, and more specifically, Scleractinian corals.

Traditionally, classification, mapping, and depiction of coral reef community structure have been performed in situ by scuba divers trained in marine ecology. Common methods for systematic depiction in quantitative studies of the reef substrate use quadrats and line transects as references to estimate attributes such as life cover, species richness, biodiversity, and population density (Laxton & Stablum, **1974**; Loya, **1972**; Stoddart, **1969**; Weinberg, **1981**). These methods are borrowed from terrestrial ecology, where they are simple to conduct. When studying the reef and its inhabitants in situ, however, divers face limitations such as depth and time. In addition, community structure classification is prone to human bias. Technological developments and engineering have helped to surmount these challenges using an array of sensors—mainly visual and acoustic. Image collections of the substrate present a repeatable, minimal impact tool for observation-based studies. Scalable approaches such as photo-mosaics now allow scientists to capture and systematically describe large-scale ecological phenomena with genus-specific resolution (Finney & Stephen, **2005**; González-Rivero et al., **2014**; Ludvigsen, Sortland, Johnsen, & Singh, **2007**; Singh, Howland, & Pizarro, **2004**). Previous work (Beijbom et al., **2012**) investigated automated approaches for determining the spatial distribution of the various organisms in a coral reef ecosystem using survey images. In particular, this study cropped image patches around the sparse labels and then performed image classification using support vector machine methods. Other works performed coral reef analysis using machine learning methods such as *k*-nearest neighbors (Manderson, Li, Dudek, Meger, & Dudek, **2017**; Mary & Dharma, **2017**; Shihavuddin, Gracias, Garcia, Gleason, & Gintert, **2013**). Nevertheless, as previously mentioned, deep learning approaches are achieving state-of-the-art performance in classification, detection, and segmentation tasks (Garcia-Garcia et al., **2017**), including coral reefs analysis (Moniruzzaman, Islam, Bennamoun, & Lavery, **2017**). Deep learning approaches have also been shown to perform better when learning from multimodal data. For example, Beijbom et al. (**2016**) and Zweifler, Akkaynak, Mass, and Treibitz (**2017**) have presented a wide field-of-view fluorescence imaging system called FluorIS, which classifies coral species better than when only using RGB images.

More recent approaches are shifting to semantic segmentation, which is able to give more detailed information (pixel-level) than the only classification. The first approaches performed image patch classification to thereafter reconstruct the segmentation of the entire image (Manderson et al., **2017**; Shihavuddin et al., **2013**). These kinds of patch-based approaches, however, typically have low accuracy near the edges of the segmented regions. To get the fully segmented image, moreover, they also need to be executed the same amount of times as the number of patches cropped from the image.

In contrast, our work presents an approach to directly learn semantic segmentation models from sparse ground truth labels, as demonstrated later, achieving better performance than earlier works based on patches. This approach is based on our earlier works (Alonso, Cambra, Munoz, Treibitz, & Murillo, **2017**; Alonso & Murillo, **2018**), which exploit superpixel segmentation to propagate the training labels, as we detail in Section **2.2**. Another recent work, demonstrating the benefits of incorporating the use of superpixels for semantic segmentation tasks using CNNs (King, Bhandarkar, & Hopkinson, **2018**), used superpixel segmentation to build a tool to facilitate the labeling process.

2.2 Lack of training data

As previously mentioned, many different projects ranging from autonomous surveys of coral reef ecosystems (Beijbom et al., **2012**; Manderson et al., **2017**) to wildlife monitoring from aerial systems (Hodgson, Baylis, Mott, Herrod, & Clarke, **2016**) focus on monitoring tasks and subsequent data analysis. To enable automatic processing of the data, semantic segmentation models for the different target domains are needed, but their use is often blocked or hampered due to the lack of dense labeling to train semantic segmentation models, especially in domains where an expert is needed to label the images. This common situation motivates the solution presented here: our method to surmount the lack of labeled training data. Before presenting our proposed methodology, we review several methods for overcoming this problem found in prior work.

2.2.1 Models for weakly labeled data

A common strategy for dealing with the lack of annotation is to build approaches that are able to learn from sparse or weakly labeled data. The survey by Hu, Dollár, He, Darrell, and Girshick (**2018**) compares different methods to train semantic segmentation from noisy and weak labels. The work discusses these problems in detail and presents some solutions.

Several recent approaches show how to make use of *per image labels* to obtain per pixel image segmentation models. This study (Kolesnikov & Lampert, **2016**) proposes a new composite loss function to train FCNs directly from image-level labels. Another study (Durand, Mordan, Thome, & Cord, **2017**) proposes a two-step approach: first, teach a CNN classification model trained on image-level labels to learn good representations and, then, use the learned feature maps to get the segmentation result.

Notwithstanding, several recent works have studied learning from *sparse labels* from different perspectives. A recent work (Uhrig et al., **2017**) proposes a new CNN architecture, sparsity invariant CNN, focused on reconstructing a dense depth map from sparse LiDAR information. This approach outputs continuous values in contrast to the classification labels. The authors work with sparse convolutions to learn directly from sparse labeling and show successful results with levels of sparsity between 5% and 70%. Label propagation was also used in Vernaza and Chandraker (**2017**), who show how to simultaneously learn a label-propagator and the image segmentation model, both with deep learning architectures. This approach propagates the ground truth labels from a few traces to estimate the main object boundaries in the image and provides a label for each pixel. In contrast, we use superpixel-based method for the propagation, resulting in better results.

2.2.2 Generating new data

Another strategy for dealing with the lack of training data is to generate additional or new data similar to the real data. Generating data by modifying its original form is a fairly common solution. Many works have used variations of this method, including the well-known Alexnet model (Krizhevsky, Sutskever, & Hinton, **2012**), which was trained using image augmentation by applying image flips and translations and altering RGB values. A more recent data augmentation solution is to generate synthetic data (Gupta, Vedaldi, & Zisserman, **2016**; Ros, Sellart, Materzynska, Vazquez, & Lopez, **2016**). This strategy provides perfect ground truth labels through image rendering. These types of methods do not always transfer generated data to real data properly, in part because, in many situations, it is hard to simulate the right amount of variability needed for the training data. Another recent work (B. Sun & Saenko, **2016**) describes how to adapt an existing model when there is no training data available for the new domain.

Contrary to the above-mentioned approaches, we study an alternative but complementary path that combines the idea of data generation (augmenting the sparse labels) and CNN models for segmentation. We demonstrate how to augment the sparse labeling using superpixel segmentation algorithms and study the effects.

This study is not the first one that uses superpixel segmentation to enhance annotation pipelines. Preliminary results of training dense segmentation models with augmented sparse labels were shown in our earlier work (Alonso & Murillo, **2018**; Alonso et al., **2017**). Other works have also built annotation tools using this approach. For example, Wigness (**2018**) proposes a superpixel labeling interface for semantic image annotation. Very similar to Wigness (**2018**), Labelbox, an online platform for semantic image annotation, commercialized this idea. In contrast to these annotation tools, the present work introduces an iterative (multilevel) and automatic method for augmenting sparse labels. Thanks to this iterative approach, the annotator does not need to change parameters such as the superpixels sizes or the number of generated superpixels. Instead, we iteratively build several levels of superpixels that perform this task automatically.

The single-level strategy that uses a fixed number of superpixels (Alonso & Murillo, **2018**) leads to a strong trade-off between accuracy and the number of unlabeled regions. The higher the amount of superpixels, the better the performance but the greater the number of superpixels that end up unlabeled. The multilevel strategy we propose here solves these problems and improves our earlier results. Our improved approach is more robust, regardless of the modality of the input images and the sparsity of the labeling than our previous results. We present significantly better performance and a more exhaustive validation, including baselines with more superpixel segmentation algorithms, results with new data sets having dense labels as well as an ablation study of several of the method's parameters.

3 TRAINING DENSE SEMANTIC SEGMENTATION WITH SPARSE PIXEL LABELS

This section describes our approach for learning a semantic segmentation model when the available training data only has sparsely labeled pixels. Figure **1** shows a summary of the main stages of our approach.

3.1 Problem formulation

Performing semantic segmentation when only sparse annotations are available is a very challenging task. In this section, we formulate the problem using two different approaches. In the first approach, we crop the image into small patches, perform patch classification and then, stitch these patches back together. In the second method, we perform per pixel classification to directly obtain the image semantic segmentation. We will compare both approaches, focusing more on the second method.

3.1.1 Per patch classification

Semantic segmentation can be formulated as a patch classification problem. When a few annotated pixels are provided, a CNN can be trained on patches cropped around those labeled pixels to get a final image segmentation joining the classification result for each patch. This strategy, which has been successfully applied in existing approaches (Beijbom et al., **2016**; Manderson et al., **2017**), is trained on nn-labeled patches, one per annotation. The training pairs used are of the form $(\mathbf{X}d(i,j),y(i,j))$(Xd(i,j),y(i,j)) where $\mathbf{X}d(i,j)$Xd(i,j) is a patch of dimensions $d\times d$d×d centered around each labeled pixel with coordinates (i,j)(i,j), and $y(i,j)$y(i,j) is a scalar representing the label of this pixel.

3.1.2 Per pixel classification

More frequently, semantic segmentation is formulated as a pixel classification problem where the input and output constitute the entire image. In this case, an end-to-end CNN architecture is trained with dense labels, that is, fully labeled images, to obtain the per-pixel classification directly, that is, the semantic segmentation. In our case where only some sparse labels are available, there are two existing approaches for addressing the sparsity: either propagate the sparse labels into dense labels or, train only on the sparse labels and ignore the nonlabeled pixels. We previously showed (Alonso et al., **2017**) that the first approach provides better results as it provides more data for training.

We consider the most common fully convolutional architectures for this problem: the FCN architecture (Long et al., **2015**), the FCN symmetric architecture (Badrinarayanan, Kendall, & Cipolla, **2017**) and the current state-of-the art, which has a light and small decoder (L.-C. Chen et al., **2018**). In all these architectures, the networks are trained with pairs of images: $(\mathbf{X},\mathbf{Y}')$,(X,Y′), where $\mathbf{X}$X is the original input image, an (m×n×cm×n×c) array (for an RGB image c= 3), and $\mathbf{Y}'$ Y′ is an (m×nm×n) array with a label for each pixel.

3.1.3 Formulation

Both the per patch strategy and per pixel approach are classification problems, whose models are obtained by minimizing the error min(|yˆ−y|)min(|yˆ−y|) between the predicted yˆyˆ and expected values yy. Both strategies are commonly optimized using the cross-entropy loss function

$\mathcal{L}$=−1N∑j=1N∑c=1Myc,jln(yˆc,j),L=−1N∑j=1N∑c=1Myc,jln(yˆc,j),

(1)

where NN is the number of labeled samples (in semantic segmentation, *N* is the number of labeled pixels) and *M* is the number of classes. Yc,jYc,j is a binary indicator (0 or 1) of pixel jj belonging to a certain class *c* and yˆc,jyˆc,j is the CNN predicted probability of pixel jj belonging to a certain class *c*. This probability is calculated by applying the soft-max function to the networks' output. In the per pixel approach, each jj represents a pixel, while in the per patch approach, each jj represents a patch, so N=1N=1 since we only have one label per patch.

3.2 Our approach: Label augmentation with multilevel superpixels

In this section, we detail our proposed strategy for sparse labeling augmentation. The goal is not only the propagation itself but also augmenting our available sparse training data to boost the training and performance of CNN-based methods for semantic segmentation. Our approach for label augmentation is based on existing superpixel segmentation techniques.

3.2.1 Superpixel (single-level)-based labeling propagation

Initially, we consider a simple but intuitive approach: single-level superpixel-based augmentation. This strategy, detailed in our preliminary work (Alonso et al., **2017**), takes an input image with sparse labels and augments them in two steps. First, the image is segmented into a preset number of superpixels, as shown in the examples in Figure **2**. Second, the sparsely labeled pixel values are propagated following the superpixel segmentation, that is, all pixels in each superpixel get the label value that appears the most within that superpixel. Figure **3** shows some binary examples using several superpixel segmentation algorithms we evaluate in this study: contour relaxed superpixels (CRS; Conrad, Mertz, & Mester, **2013**), Pseudo-Boolean (PB; Zhang, Hartley, Mashford, & Burn, **2011**), entropy rate superpixel (ERS; Liu, Tuzel, Ramalingam, & Chellappa, **2011**), simple linear iterative clustering (SLIC; Achanta et al., **2012**), and superpixels extracted via energy-driven sampling (SEEDS; VandenBergh, Boix, Roig, deCapitani, &

VanGool, **2012**). Section **5.1** compares the performance of these methods in the proposed label augmentation strategy.

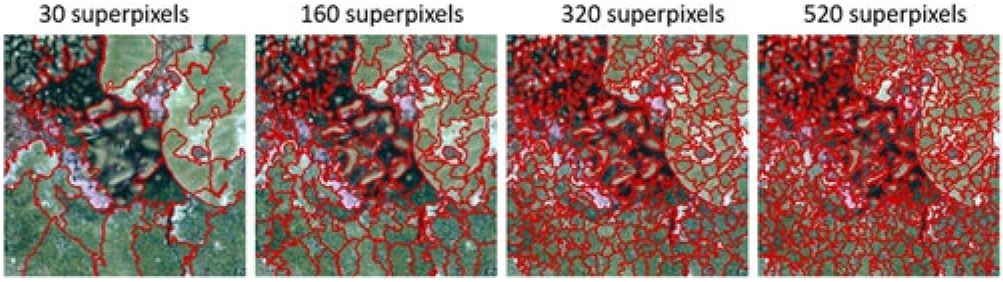


Figure 2

Superpixel segmentation obtained when varying the target number of superpixels (clusters). These images have been segmented using the SEEDs algorithm

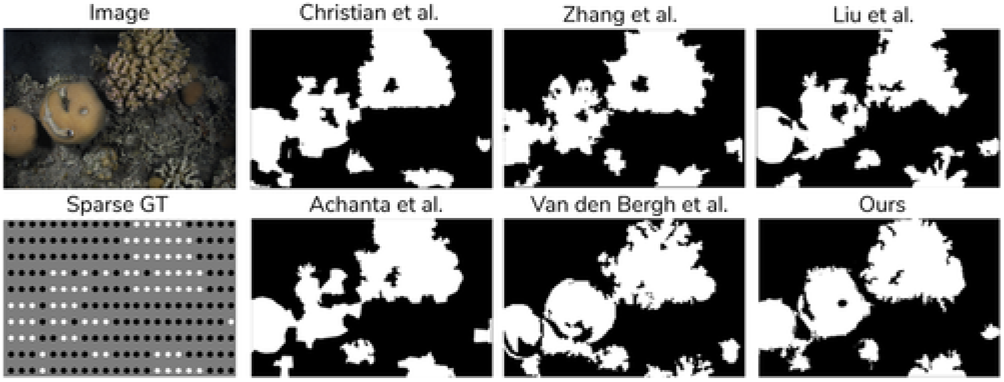
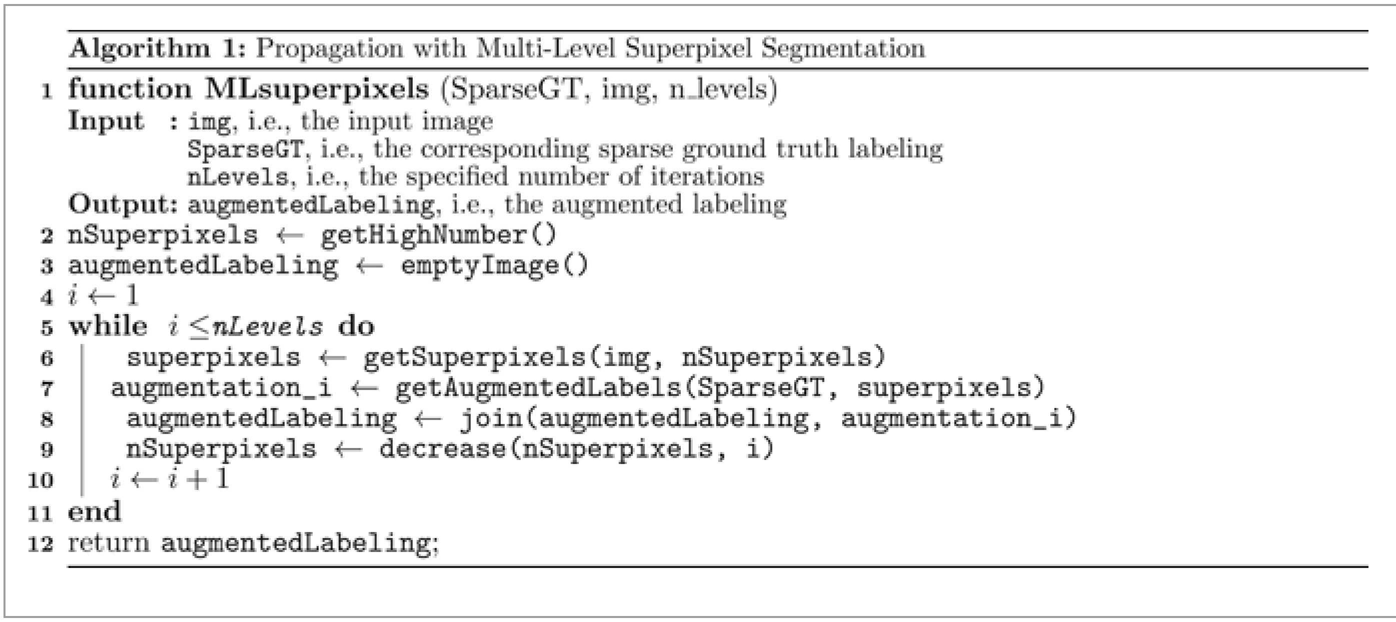


Figure 3

Sparse ground truth label augmentation obtained with different superpixel segmentation techniques (black and white dots represent one single labeled pixel). The top-left view is the original image and the bottom left view is the sparsely available ground truth. The rest of the images are binary (coral/no-coral) labeling augmentations

This single-level superpixel strategy has been used in prior works with promising results (Alonso et al., **2017**; King et al., **2018**) but has some drawbacks because the number of superpixels has to be specified a priori. Consequently, two issues can potentially arise. Either some superpixels may not contain any labeled pixels (and, therefore, generate unlabeled regions) or the superpixels may be too large to fit complex or very small image shapes accurately. This leads to a strong trade-off between proper contour fit and the number of unlabeled regions: a higher number of superpixels fits the actual shapes better, but it increases the number of superpixels that are left without any label. Our proposed multilevel strategy extension solves these problems.



Algorithm 1: Propagation with Multi-Level Superpixel Segmentation

```
function MLsuperpixels (SparseGT, img, n_levels)
Input  : img, i.e., the input image
         SparseGT, i.e., the corresponding sparse ground truth labeling
         nLevels, i.e., the specified number of iterations
Output: augmentedLabeling, i.e., the augmented labeling
nSuperpixels ← getHighNumber()
augmentedLabeling ← emptyImage()
$i \leftarrow 1$
while $i \leq nLevels$ do
    superpixels ← getSuperpixels(img, nSuperpixels)
    augmentation_i ← getAugmentedLabels(SparseGT, superpixels)
    augmentedLabeling ← join(augmentedLabeling, augmentation_i)
    nSuperpixels ← decrease(nSuperpixels, i)
    $i \leftarrow i + 1$
end
return augmentedLabeling;
```

3.2.2 Multilevel superpixel segmentation

The proposed multilevel superpixel segmentation (see Algorithm 1) consists of applying the superpixel image segmentation iteratively, progressively decreasing the number of superpixels generated in each iteration. The input of Algorithm 1 is an image, the sparse ground truth, which is an image with some labeled pixels (nonlabeled pixels will have a special value) and the number of levels, which is a positive integer number and defines the number of iterations to be performed.

In the first iteration, the number of superpixels is very high, leading to very small-sized superpixels for capturing small details of the images (the propagation is performed exactly as the single-level approach). The number of superpixels vis-a-vis the number of labeled pixels is automatically computed. This value can also be given as an extra parameter. In Section **5.1** we evaluate how this parameter affects the quality of the augmentation.

In the first iteration, as the superpixels are small, the label augmentation results in many unlabeled regions. The following iterations decrease the number of superpixels, leading to larger superpixels covering unlabeled pixels (see Figure **4**). Successive iterations do not overwrite information; they only add new labeling information until all pixels are covered. Parameter values for Algorithm 1 are specified in Section **5.1**. Our code is available online.

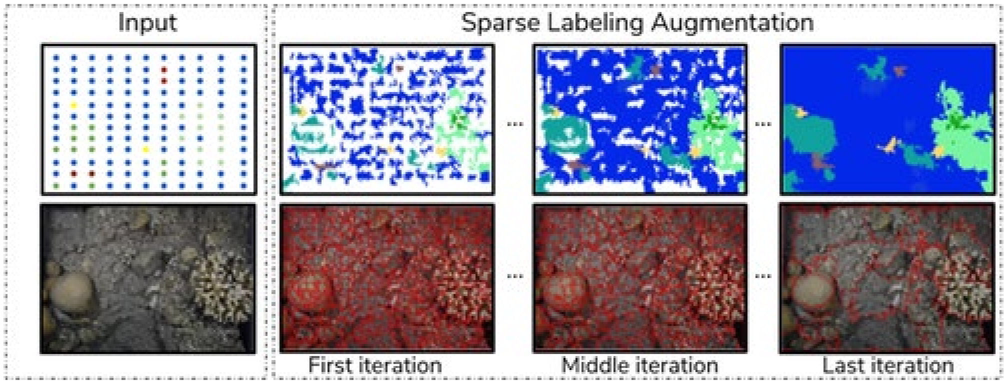


Figure 4

Multilevel superpixel label augmentation algorithm. [Left] The input of the algorithm (available sparse labels and corresponding image). [Right] The augmentation process: augmented labels (top row) after the first, middle and last iteration, and the superpixel segmentation obtained at that level (bottom row). The output of the method is the augmented labeling from the last iteration (right column)

3.3 Semantic segmentation architectures and optimization

3.3.1 Architectures considered

Deep learning architectures for semantic segmentation have advanced since (Long et al., **2015**) blazed a path to build different types of decoders to upsample the learned features of the encoder. Their work uses bilinear interpolation for upsampling the last encoder layer into the output resolution. The second type of FCNs reverses the encoder architecture by constructing a symmetric architecture where the decoder has the same or similar computation as the encoder. This kind of architecture usually performs better but at a higher computational and temporal cost. SegNet (Badrinarayanan et al., **2017**) and FC-Densenet (Jégou et al., **2017**) are two examples of well-known architectures using this type of decoder.

The current state-of-the-art of semantic segmentation, Deeplabv3+ (L.-C. Chen et al., **2018**), follows a third and different strategy. It is based on focusing the computation on the encoder and having a light decoder that learns to decode the learned representation and requires little computation. The main features of Deeplabv3+ are the use of depth-wise separable convolutions (Kaiser, Gomez, & Chollet, **2017**), which allow convolutions to be performed with less computation and perform better when channels are decorrelated; spatial pyramid pooling (He, Zhang, Ren, & Sun, **2014**), which allows joining of information from different resolutions in one stage; and use of dilated convolutions (Yu & Koltun, **2015**), which allows learning of complex relations between spatially separate information without the need to reduce the resolution. For our main study case, coral imagery semantic segmentation, previous work (King et al., **2018**) has also shown that the Deeplab architectures perform better than other architectures.

In our experiments, we compare the Deeplabv3 encoder architecture with the three different types of decoders described above, to see how they affect the architecture. Thus, we compare Deeplabv3, Deeplabv3+, and Deeplabv3-symmetric. We use the official implementation for the first two architectures. **5** For the last architecture, we modify Deeplabv3+, turning it into a symmetric FCN architecture. Section **5.2.2** discusses the results obtained with our trained models using these three alternative architectures, both single-level and multilevel trained from scratch, as well as explore some fine-tuning options.

3.3.2 Loss function comparison

Apart from selecting a suitable neural network architecture, another crucial decision is selecting the loss function, as it directs the learning of the neural network. Deep learning architectures for semantic segmentation are commonly optimized using the cross-entropy loss function. Nevertheless, there are other variations, which we describe below. In this study, we propose a modification of the cross-entropy loss function that takes into account the neighboring pixels without adding much computation.

Cross-entropy loss function

The cross-entropy loss the common loss function for classification and semantic segmentation (see Equation **1** in Section **3.1**). It optimizes the accuracy per pixel. For classification, this fits perfectly, but for semantic segmentation, it is applied to every pixel independently and does not include information about neighboring pixels (DeBoer, Kroese, Mannor, & Rubinstein, **2005**).

Lovasz loss function

Recently, a novel approach for optimizing neural networks for semantic segmentation was developed (M. Berman, Triki, & Blaschko, **2018**). Instead of optimizing the accuracy of every pixel individually, this study tries to optimize the mean intersection over union (MIoU; Garcia-Garcia et al., **2017**), the standard metric for semantic segmentation. One main drawback of this approach is the computation time. Computation of this loss takes around five times more than calculating the cross-entropy loss function.

Cross-entropy loss function with median frequency balancing

This is a modification of the cross-entropy loss function. It consists of adding weights to every semantic class to optimize the mean accuracy per class, reducing the effect of the class imbalance. Every class *c* is weighted according to the following formula: $w(c)=mf/f(c)$w(c)=mf⁄f(c), where *w* is the weight of class *c, m* is the median frequency, and *f* is the frequency of a class *c* (Badrinarayanan et al., **2017**).

Our loss function

We developed a modification of the cross-entropy loss function to take into account the prediction of neighboring pixels without adding much computation. In most semantic segmentation use cases, if one pixel belongs to a certain class, its neighbors (at different distances) are likely to belong to the same class. Thus, following this intuition, we give more importance (higher loss) to pixels whose neighboring pixel predictions are not the same (we consider the pixel connectivity as 4-neighbor, i.e., 4-connectivity). By applying this idea, we achieve two main benefits:

- The loss will prevent the algorithm from predicting isolated pixels, that is, pixels of the same type are usually together. This will help the overall accuracy and MIoU performance.
- The classes with less data will have fewer neighbors of their type; therefore, these classes will have a higher impact on the loss, correcting the class imbalance.

Following the idea of the median frequency balancing, we add some weights to every pixel *p* as follows:

$w(y\hat{})=\text{norm}[1+\sum n=0N\text{gauss}(\sigma,n)(4-\delta(y\hat{}i,j,y\hat{}i,j+2n)+\delta(y\hat{}i,jy\hat{}i+2n,j)+\delta(y\hat{}i,j,y\hat{}i-2n,j)+\delta(y\hat{}i,j,y\hat{}i,j-2n))]$,w(yˆ)=norm[1+∑n=0Ngauss(σ,n)(4−δ(yˆi,j,yˆi,j+2n)+δ(yˆi,jyˆi+2n,j)+δ(yˆi,j,yˆi−2n,j)+δ(yˆi,j,yˆi,j−2n))],

(2)

where δδ is the Kronecker delta (the function is 1 if the variables are equal, and 0 otherwise), NN is the number of neighboring levels to evaluate (a neighboring level nn represent neighboring pixels at distance $2n$2n in pixels) and it is always set as the maximum possible with $y\hat{}$yˆ as the predicted class. We introduce the Gaussian function to force the neighbors closest to the pixel to have more impact on the weight. The σσ value affects the importance that neighboring pixels are given. In two cases, all neighbors have the same weight: when $\sigma=0$σ=0, the multiplicative factor of all the neighbors is zero; and when $\sigma=\text{inf}$σ=inf, the multiplicative factor of all the neighbors is the unity. The weight normalization (norm) consists of getting weights with mean equal to one with respect to all the predicted pixels. In Section 5.2.1 we evaluate the effect of the parameter σσ.

4 DATA SETS AND LABELS

This section details the main data sets and evaluation metrics used in the experiments.

4.1 Data sets

For the coral segmentation experiments, we use four different data sets, summarized in Table **1**.

Table 1. Details of the coral data sets used in this study

Data sets	Train images	Test images	Semantic classes	Label type	Total labeled pixels
CoralNet	416,512	14,556	191	Classification	431,068
Eilat	142	70	10	Sparse	42,400
EilatMixx	23	8	10	Sparse	5,109
Mosaics UCSD	4,193	729	35	Dense	1,290 M

- *CoralNet*. We processed all the CoralNet public data to get a useful and robust data set, containing image crops around the sparse pixel labels having different sizes: 32×32,64×6432×32,64×64 and 128×128128×128. We only kept the semantic classes that had at least two thousand samples. The resulting data set consists of 431,068 images. These images are from over 40 different geographical sources from around the world. Each image has at least one semantic label out of the 191 different coral species this data set considers. We randomly selected 95% of the data for training the encoder and only 5% for testing it. The main use we make of this data set is to train a *generic encoder for coral images* to learn better representations for this type of images. The source data is available at the CoralNet project website.

- *Mosaics UCSD* (Edwards et al., **2017**). The original data set consists of 16 mosaics with resolution of over. 10K × 10K. The data set used in this study is the result of cropping these mosaics into 512×512512×512 images, resulting in 4,193 training images (85% randomly selected) and 729 test images (15% randomly selected). The data set contains 34 different semantic classes plus the background class we ignore and provides dense labels (all pixels in each image are labeled). This data set is used for many of our experiments due to the quantity of labeled images it has and because its labels are dense, allowing more accurate/reliable metrics.

- Eilat (Beijbom et al., **2016**). This is a publicly available coral data set consisting of 142 training images and 70 validation images. The resolution of the original images was 3K × 5K but, for our experiments, we downsized them ×4 due to memory issues when feeding the CNNs. Although the labeling of this data set is sparse and it only has few labeled pixels per image, it also has binary (coral vs. noncoral) dense labels for a subset of its images. Apart from the RGB image channels, it has two additional channels with fluorescence information.

- *EilatMixx*. This data set consists of 31 images from the same geographical area as the Eilat data set but acquired at a significantly different time (3 years later: the Eilat data set is from 2015 and the EilatMixx is from 2018). It contains images of the same coral species at the same resolution and with the same image processing (color correction) as the Eilat data set. This data set and the Eilat data set show how challenging and heterogeneous images acquired at the same areas but at different times are. They are used in our experiments to prove that we can learn and adapt coral semantic segmentation to a new situation when having only a few sparsely labeled pixels. Both Eilat data sets contain coral images from the Red Sea (Israel). In contrast to the Eilat data set and the CoralNet data set, this data set has been annotated such that specific points of interest within the image were chosen rather than having a random or uniform point grid in which not every significant object gets labeled. This data set has fewer images than the other data sets, which is useful in our experiments as it helps to prove how the generic encoder supports learning a model for a new scenario when few training images are available.

Figure **5** shows some examples from all these data sets. The EilatMixx data set is released to the community, including the new images, the original labels and our automatically augmented labels for the Eilat, EilatMixx, and Mosaics UCSD data sets.

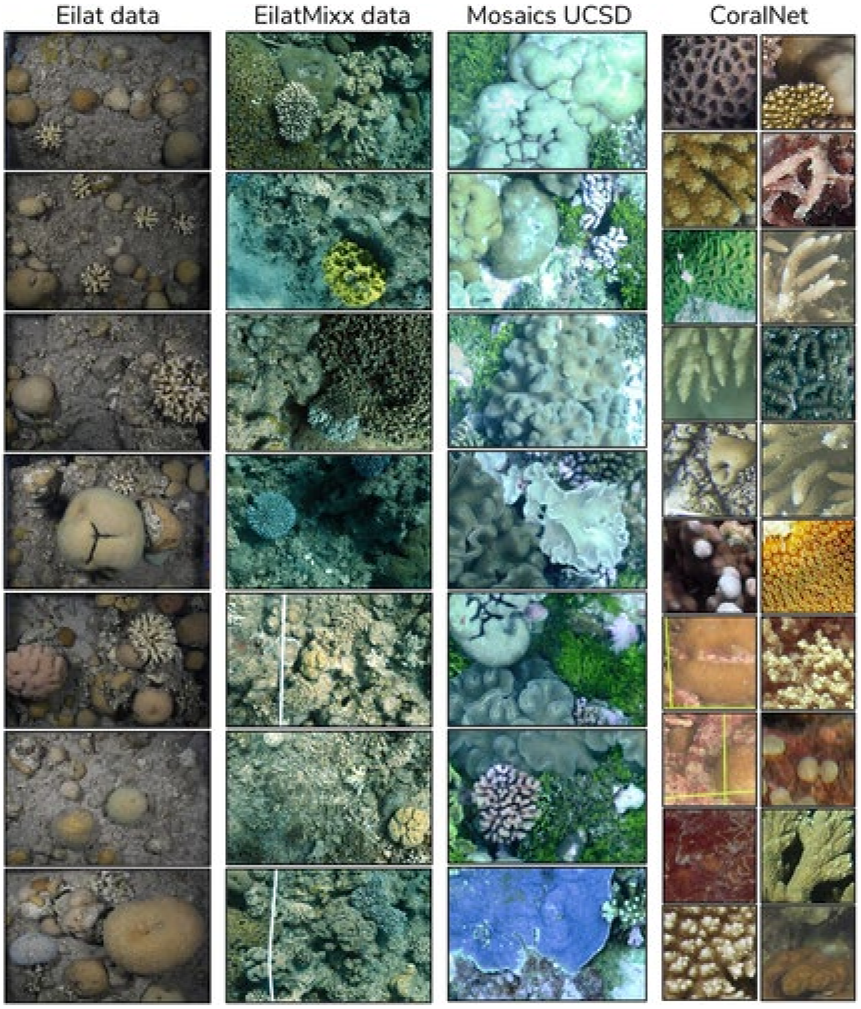


Figure 5

Several images of the four different data sets used in this study. From left to right: Eilat (Beijbom et al., **2016**), EilatMixx (ours), Mosaics UCSD (Edwards et al., **2017**), and CoralNet

4.2 Reference labels

As the data sets have either sparse or dense labels, we use different labels to evaluate the results of the segmentation models obtained, depending on the available labels.

The Eilat and EilatMixx data sets, which only provide sparse annotations, are evaluated with metrics computed using three different reference labels

- *Original-GT*: The original sparse labels available with the data set. This is the least representative and reliable of the three ground truth options since it has very few annotations per image, but it is necessary to perform direct comparisons with previous results that used it.

- *Augmented-GT*: The augmented ground truth obtained by our approach. This is an approximated labeling because it contains some noise. It does, however, provide a very representative reference labeling (Alonso & Murillo, **2018**; Alonso et al., **2017**).
- *Dense-GT*: We use this only for the Eilat data set. It contains a few dense labeled images for binary (coral vs. noncoral) segmentation obtained by an expert coral biologist. It is only available for some images but is the most reliable and representative to use when comparing results of the semantic segmentation task.

The Mosaics UCSD data set is the only one with dense labels. The results using this data set are evaluated using these dense ground truth labels. As this is the most reliable evaluation, the majority of the experiments will be performed with this data set.

4.2.1 Metrics for evaluation

The metrics we use for our evaluation are the standard metrics for semantic segmentation. We just consider different types of ground truth (explained above) to compute it: pixel accuracy (PA); mean pixel accuracy (MPA; per class) and the MIoU.

5 EVALUATION OF OUR PROPOSED APPROACH

5.1 Labeling augmentation quality using multilevel superpixels

This section evaluates our labeling augmentation method detailed in Section **3.2**.

5.1.1 Experiment setup

For all the following experiments, the multilevel superpixel-based augmentation starts with an initial number of superpixels (initnsinitns) set to 10 times the number of labeled pixels for each image. We set the final number of superpixels (finalnsfinalns) to the 10th of labeled pixels per image. Then, given a number of levels (NL) to complete, the number of superpixels (*N*supNsup) to generate at each level (level*i*leveli) would be

*N*sup(level*i*)=initns□□□□(finalnsinitns)level*i*NL□□□□Nsup(leveli)=initns[(finalnsinitns)leveliNL].

(3)

Tables **2** and **3** show the comparison between the single-level augmentation used in recent previous work (Alonso et al., **2017**) and other work that followed aimed at building an annotation tool (King et al., **2018**), and the proposed multilevel augmentation (ours) using different superpixel segmentation techniques.

Table 2. Labeling augmentation quality when using the single-level and multilevel (15 levels) approaches

	Metrics		
Augmentation approach	**PA**	**MPA**	**MIoU**
SEEDS single-level (Alonso et al., 2017)	82.60	81.75	62.05
SEEDS multilevel (*ours*)	88.66	86.28	75.74
SLIC single-level (Alonso et al., 2017)	86.93	85.72	73.20
SLIC multilevel (*ours*)	**88.94**	**87.00**	**76.96**
CRS single-level (Alonso et al., 2017)	80.02	78.82	58.77
CRS multilevel (*ours*)	87.03	84.91	72.88
ERS single-level (Alonso et al., 2017)	79.52	80.09	59.42
ERS multilevel (*ours*)	86.65	84.56	73.13
PB single-level (Alonso et al., 2017)	78.66	81.02	57.41
PB multilevel (*ours*)	**85.74**	**83.01**	**70.70**

- *Note*: Data set: Mosaics UCSD. Evaluation on the dense labels.
- Abbreviations: CRS, contour relaxed superpixel; ERS, entropy rate superpixel; MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean; SEEDS, superpixels extracted via energy-driven sampling; SLIC, simple linear iterative clustering.
- Bold numbers highlight the best performing method on each metric.

Table 3. Labeling augmentation quality when using the single-level and multilevel (15 levels) approaches on different input modalities (RGB and fluorescence images)

	Metrics		
Augmentation approach	**PA**	**MPA**	**MIoU**
Evaluation based on Dense-GT			
Using RGB			
SEEDS single-level (Alonso et al., 2017)	92.21	80.20	72.90
SEEDS multilevel (*ours*)	93.23	84.91	75.37
SLIC single-level (Alonso et al., 2017)	92.03	81.93	73.87
SLIC multilevel (*ours*)	92.76	83.60	75.37
Evaluation based on Dense-GT			
Using fluorescence			
SEEDS single-level (Alonso et al., 2017)	93.38	86.86	77.86
SEEDS multilevel (*ours*)	**94.20**	**87.50**	**79.88**
SLIC single-level (Alonso et al., 2017)	93.22	84.96	77.44
SLIC multilevel (*ours*)	93.86	85.37	78.37

- *Note*: Data set: Eilat.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean; SEEDS, superpixels extracted via energy-driven sampling; SLIC, simple linear iterative clustering.
- Bold numbers highlight the best performing method on each metric.

As Table **2** shows, we perform a more exhaustive comparison with the Mosaic UCSD data set because it has dense labeling and more semantic classes. We compare the two approaches using five different superpixel segmentation algorithms (SEEDS, Van den Bergh et al., **2012**; CRS, Conrad, Mertz, & Mester, **2013**; ERS, Liu et al., **2011**; SLIC, Achanta et al., **2012**; and PB, Zhang et al., **2011**). Our multilevel approach outperforms the single-level method by 3.76% MIoU using SLIC and by 14.11% using CRS. This is a significant improvement because the augmented labeling has to be the most accurate as possible if we want to learn a semantic segmentation model from it. Figure **6** shows some visual examples, comparing the single-level and multilevel augmentations. Clearly, the multilevel algorithm outperforms the single-level method and fits the coral reef shapes better.

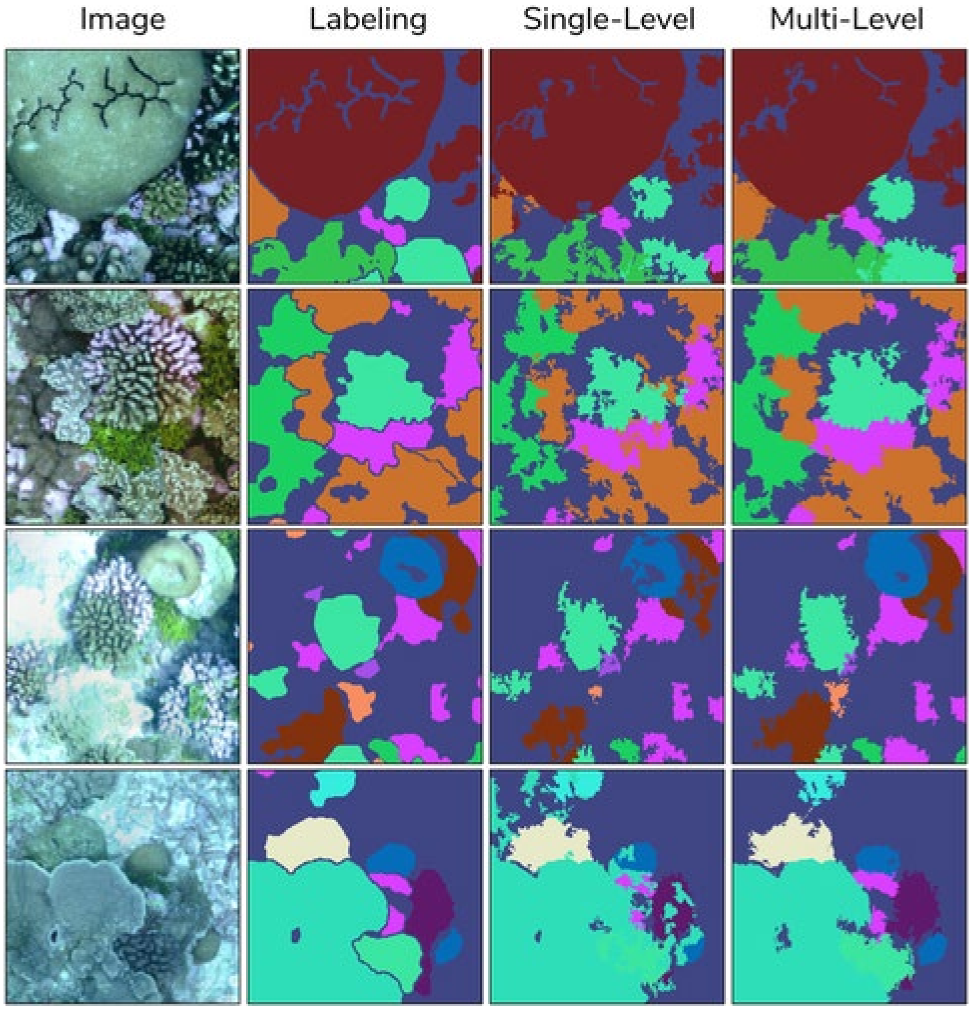


Figure 6

Comparison between the single-level and the multilevel approaches. Both are augmented from 300 labeled pixels and use the superpixels extracted via an energy-driven sampling algorithm

Regarding the Eilat data set, the SLIC and SEEDS superpixel algorithms also outperform the ERS, CRS, and PB methods. What is especially interesting about this data set is the multimodal information (fluorescence) it provides. Fluorescence is a very relevant and informative source of information regarding coral reefs (Beijbom et al., **2016**; Zweifler et al., **2017**). In Table **3** we show how this fluorescence information can enhance the labeling augmentation process.

We perform two small experiments to show the temporal cost of our proposed method and how the performance of our multilevel approach changes when varying the number of levels and the image resolution. Table **4** shows how the resolution (*r*) and the number of levels (*n*) of our multilevel algorithm affect the quality of the augmented labeling and the execution time. This experiment uses SLIC superpixels because they perform better on this data set (see Table **2**). Although the resolution barely affects the performance, as might be expected, it does affect the number of superpixels.

Table 4. Performance (MIoU/time in seconds using an Intel Core i7-6700) when varying the number of levels in the labeling augmentation and the image resolutions

	***N*-Levels**			
Resolution	**1**	**5**	**15**	**30**
256 × 256	73.12/0.3	74.80/1.12	76.78/3.23	77.11/6.12
512 × 512	73.20/1.34	74.85/5.87	76.96/5.72	77.21/30.03
1024 × 1024	73.31/8.4	74.91/39.76	77.10/113.56	77.25/219.56

- *Note*: Experiment performed on Mosaics UCSD data set. Evaluation on the dense labels. Sparsity used as input: 0.1% of the labeled pixels (300 pixels).

The number of superpixels considered in this evaluation increases from 1 (single-level) to 5, 15, and 30. As a result of this evaluation, we can see that as the number of superpixel increases, the accuracy of the method improves. The upper limit of the number of superpixels, at which point the accuracy starts to converge, is around 15–30 superpixels. This is why in the majority of our experiments, we use 15 superpixels as the default number for the multilevel approach. Note that our algorithm is linear in the number of levels $O(n)$O(n) and quadratic in the resolution $O(r2)$O(r2), that is, linear in the number of pixels.

One important improvement in our approach in this study, compared to our previous work, is the speed-up. Whereas in our earlier version (Alonso & Murillo, **2018**), processing 1024×10241024×1024 pixel image required 113 s, in this improved version, it takes 40 s.

Our proposed algorithm also requires other parameters to be set, such as the initial number of superpixels. This number has to be set empirically. A high number (i.e., in the order of 103103) is sufficient for the proper functioning of the system (see Figure **2** for a visual representation of the effects of this number). We analyze the influence of varying this parameter with a small experiment. Table **5** shows that an initial value in the order of 102102 works worse than one in the order of 103103, which is very similar to the order of 104104 (our algorithm's resolution, linear in the number of pixels; see above). Therefore, some thousands of superpixels are enough to capture the small details of the images. In contrast, the final or the last number of superpixels has to be set as a low number to be able to fill out and label all the pixels of the image, for example, five superpixels.

Table 5. Performance (MIoU) when varying the number of superpixels in the first level of our algorithm

***N* superpixels**	**MIoU**
×100 the number of labeled pixels (30,000)	77.07
×10 the number of labeled pixels (3,000)	76.96
×1 the number of labeled pixels (300)	74.87

- *Note*: Experiment performed on Mosaics UCSD data set. Evaluation on the dense labels. Sparsity used as input: 0.1% of the labeled pixels (300 pixels). We use 15 levels for this experiment.
- Abbreviation: MIoU, mean intersection over union.

5.2 Analysis of semantic segmentation methods

This section discusses all the semantic segmentation experiments described in Section **3.3**.

5.2.1 Efficient semantic segmentation

This experiment compares the performance of different common losses for semantic segmentation including our proposed modification of the cross-entropy detailed in Section **3.3.2**.

Experiment setup

To perform a fair comparison, for all the executions we use the same semantic segmentation model: Deeplabv3+ (L.-C. Chen et al., **2018**). We train it for 600 epochs with an initial learning rate of 10−310−3 with a polynomial learning rate decay schedule. During the training, we perform data augmentation: vertical and horizontal flips, contrast normalization, and random image shifts and rotations. For this experiment, we use the Mosaics UCSD data set because it has dense annotations that facilitate a fair evaluation.

Loss function comparison

Table **6** shows a comparison between the most common losses used in semantic segmentation using deep learning and our proposed modification of the cross-entropy loss. The level of performance of the functions is close; however, our modification performs slightly better than the cross-entropy loss for the most important metrics for semantic segmentation. In contrast, the median frequency balancing performs better for mean accuracy, as might be expected, having a negative effect on the accuracy per pixel and on the MIoU. Our proposed modification has no negative effect on any of the metrics. Analyzing its properties in more detail, we see that increasing the number of neighboring pixels to take into account ($\sigma>0\sigma>0$) increases the performance. We also see that giving less weight to far neighboring pixels ($\sigma<\sigma<$ inf) also has a positive effect on the performance. In this experiment, we set $\sigma=3\sigma=3$, as an example of $0<\sigma<0<\sigma<$ inf. We empirically found that values $2<\sigma<52<\sigma<5$ work very similarly. Regarding the time for performance, using the Mosaics UCSD data set, one epoch takes almost 8 min, but the Lovasz loss takes 37 min per epoch, which is almost five times more than the other losses.

Table 6. Semantic segmentation performance using different loss functions for training

	Metrics		
Loss configuration	**PA**	**MPA**	**MIoU**
Cross-entropy (De Boer et al., 2005)	85.31	55.78	45.60
Median freq. balancing (Badrinarayanan et al., 2017)	82.11	**61.96**	43.02
Lovasz (M. Berman et al., 2018)	85.15	59.91	47.28
Ours ($\sigma=0\sigma=0$)	85.54	58.17	47.59
Ours ($\sigma=3\sigma=3$)	**86.11**	59.90	**49.16**
Ours ($\sigma=\sigma=$ inf)	85.97	59.72	48.76

- *Note*: Experiment performed on Mosaics UCSD data set. Evaluation on the dense labels.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.
- Bold numbers highlight the best performing method on each metric.

5.2.2 Semantic segmentation architectures

This experiment compares the performance of the different common architectures for semantic segmentation detailed in Section **3.3.1**.

Experimental setup

To perform a fair model comparison, we use the same configuration for all models. The training configuration is the same as in the previous experiment with the exception that we use the same loss: our modification of the cross-entropy. We use the Mosaics UCSD data set for this experiment because it has dense annotations that facilitate a fair evaluation. The batch size is set to 8, except for the Deeplabv3-symmetric (batch size of 6) due to memory issues.

Architecture comparison

Table **7** shows the performance comparison of different Deeplabv3-based architectures, that is, the same state-of-the-art encoder with different decoder options to achieve the segmentation (more details are given in Section **3.3.1**). The performance gap between the Deeplabv3 and Deeplabv3+ models is small in our case, compared with the larger increases observed in prior work using other data sets (L.-C. Chen et al., **2018**). The results using our modification of Deeplabv3-symmetric show that the symmetric architecture performs better, but demands a noteworthy increase in the computation and inference times. The symmetric architecture has a larger decoder that is able to learn how to decode the features better. One possible problem of such a deep architecture is the vanishing gradient problem, but the skip connections between the early layers of the encoder and the later layers of the decoder solve this problem. As the symmetric architecture has more convolutional layers and, therefore, more parameters to learn, this architecture performs slightly better than the other architectures. Nevertheless, some applications may not be able to afford the additional computation and time costs.

Table 7. Semantic segmentation performance of different architectures

	Metrics					
Architecture	**PA**	**MPA**	**MIoU**	**GPU time**	**GFlops**	**Params**
Deeplabv3 (L.-C. Chen et al., 2017)	85.72	58.73	48.41	**22 ms**	**48.80**	**40.89 M**
Deeplabv3+ (L.-C. Chen et al., 2018)	86.11	59.90	49.16	26 ms	51.44	41.05 M
Deeplabv3-symmetric	**87.16**	**61.12**	**51.57**	41 ms	65.63	43.33 M

- *Note*: GPU time is the inference time on a Titan XP GPU. Experiment performed on Mosaics UCSD data set. Evaluation on the dense labels.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.
- Bold numbers highlight the best performing method on each metric.

5.3 Training with augmented labels

This experiment aims to answer one of the main research questions of this study: *Can we get a semantic segmentation model trained from sparse labels that is similar to one trained using dense labels?*

5.3.1 Experimental setup

To answer this question, we compare the semantic segmentation results of a model trained on dense labels and models trained on our augmented labels from sparse labels. We trained the Deeplabv3-symmetric architecture (the one that performed best in Section **5.2**) with the dense labeling and two different augmented labeling setups: augmented labeling from 300 labeled pixels (0.1% of the dense labels) and with only 30 labeled pixels (0.01% of the dense labels). Regarding the augmentation process, we set the number of levels to 15. These three models are evaluated with dense labels. To perform a fair model comparison, we use the same configuration for all models. The training configuration is the same as the previous experiment with the exception that here we use the same loss, our modification of the cross-entropy that gives the best results.

5.3.2 Results on the mosaics UCSD data set

The results shown in Table **8** suggest, as expected, that having more labeled pixels, the results improve. Nevertheless, training with only some labels and augmenting them with our approach leads to similar performance while significantly reducing the labeling annotation cost. The main reasons for the great performance of our method are that neural networks can learn and generalize representations even with some noise in the labels (C. Sun, Shrivastava, Singh, & Gupta, **2017**) and that our augmented labeling as shown in Table **2** and Figure **6** is fairly similar to the dense labels (superpixel techniques adjust quite well to object edges).

Table 8. Semantic segmentation performance of different training approaches: Training with dense labels, augmented labels (from 300 labeled pixels) and augmented labels (from 30 labeled pixels)

	Metrics		
Trained on	**PA**	**MIoU**	**MPA**
Dense labels	87.16	61.12	51.57
Augmented labels (300 labeled pixels)	86.30	60.00	49.93
Augmented labels (30 labeled pixels)	84.10	59.19	48.73

- *Note*: The experiment used the Mosaics UCSD data set. Evaluation on the dense labels
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.

We show that with our modified architecture (Deeplabv3-symmetric) and training with our augmented labeling from 30 pixels (Table **8**), we get the same results as training with Deeplabv3 with the dense labels (Table **7**). We also get even better performance when training with our Deeplabv3-symmetric architecture and the augmented labeling from 300 pixels than when training with Deeplabv3+ and the dense labels. One thing to take into account in our labeling augmentation approach is that its performance depends on how detailed the data set is. This means that the more objects in the images, and the smaller they are, the more difficult to augment the labeling. In other words, our method needs to have at least one labeled pixel per object/instance in the image to be able to properly augment the labeling.

For the multilevel augmentation, we evaluated other potential improvements, which did not improve the augmented labeling results. The most interesting modification studied is weighting the loss corresponding to different augmentation levels differently. The intuition is that the augmented labels near the seeds (the sparse labels from which we augment) should have more impact on the loss because they should be more reliable and have a higher probability of being correctly labeled. The experiment results, however, did not show significant improvements.

5.3.3 Results on the Eilat data set

Regarding the Eilat data set, we compare our approach with prior work published by the authors of the data set for multiclass semantic segmentation. The authors (Beijbom et al., **2016**) perform a patch-based classification approach (explained in Section **3.1**, the same approach that other works have followed; Manderson et al., **2017**). We also compare our approach to our baseline and previous work (Alonso et al., **2017**).

Table **9** summarizes these results. We compare results from Beijbom et al. (**2016**; Patch-based v1) with our implementation of it using a newer CNN model (Patch-based v2). Note that (v2) performs the same as or better than the original (v1) and that (v1) is shown only where the original publication included results. Results also include our previous work (Baseline) with the single-level label augmentation (Alonso et al., **2017**), and our work presented here (Ours). We show the original-GT scores because some related work has published results using this. Note, however, how the proposed method significantly outperforms previous work on the more significant dense scores.

Table 9. Semantic segmentation performance when training from sparse labels

	Metrics		
Method	**PA**	**MPA**	**MIoU**
Evaluation on dense scores: Augmented-GT			
Patch-based v1 (Beijbom et al., 2016)	–	–	–
Patch-based v2 (Beijbom et al., 2016)	73.61	25.32	17.89
Baseline (Alonso et al., 2017)	85.88	42.25	31.12
Ours	**90.02**	**47.61**	**40.65**
Evaluation on sparse scores: Original-GT			
Patch-based v1 (Beijbom et al., 2016)	87.80	48.50	–

	Metrics		
Method	PA	MPA	MIoU
Patch-based v2 (Beijbom et al., 2016)	**90.20**	53.10	43.66
Baseline (Alonso et al., 2017)	81.23	41.97	28.14
Ours	84.80	**54.65**	**44.01**

- *Note*: The experiment used the Eilat data set.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.
- Bold numbers highlight the best performing method on each metric.

6 A GENERIC PRETRAINED CORAL ENCODER

6.1 Pretraining and fine-tuning

In this section, we study how to train models for coral segmentation that can generalize to other regions or across time.

Pretraining deep learning models on large general data sets and then fine-tuning for more specific tasks is a widespread practice that improves deep learning performance (LeCun, Bengio, & Hinton, **2015**), especially when large amounts of labeled data are not available. It consists of training the model on a large database of a similar domain and using that trained model as initialization for training with the specific task data. This pretraining generalizes the final model and prevents overfitting when the specific training data is not large enough or heterogeneous. The fine-tuning of a pretrained model can be carried out in different ways, including adjusting the number of layers vis-a-vis the original model. This process depends mostly on how different the pretrained domain is from the target domain (the more different, the more layers we need to adjust) and how much labeled data from the target domain are available (the fewer data we have, the fewer number of layers we would typically fine-tune).

We built a generic model using a large set of coral reef data from many different locations. Our pretrained encoder is the equivalent of what is commonly done with general-purpose detection and classification, through pretrained encoders on Imagenet (Deng et al., **2009**), but ours is specifically for corals. One of the largest existing sources of coral data, CoralNet (Beijbom et al., **2012**), is a resource for benthic images analysis and also serves as a repository and collaboration platform. In cooperation with the CoralNet team, we extracted and cleaned their public data to get a useful and robust data set. This data set consists of 431,068 images of 191 different coral species (see Table **1**). Its training set has between one and 2,500 images per coral reef class, and the test set has up to 100 images per coral reef class.

We trained the encoder used on the three Deeplabv3 architectures using this CoralNet data set (see Figure **7**).

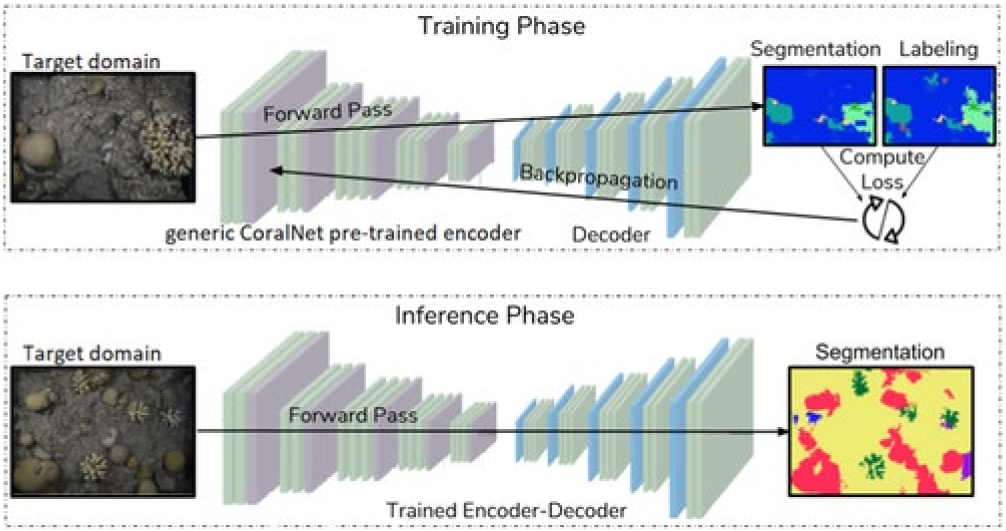


Figure 7

Training and inference phases. In the training phase, the encoder used is the generic encoder pretrained on CoralNet. The training is performed with the target domain data using augmented labeling. The inference phase provides the semantic segmentation result from the target domain with data for a new scenario using a generic pretrained encoder

Note that training a general semantic segmentation was not feasible due to lack of data and difficulty in generalizing coral appearances. Since having a general segmentation model that contains all possible classes of interest on all the coral reef scenarios is the objective, our goal is to provide a generic encoder that has learned good features representing this kind of underwater imagery. Coral segmentation models for specific new scenarios can benefit from this pretrained encoder. In the following experiments, we demonstrate two main benefits of using this pretrained generic encoder we have now made available: better performance and faster convergence.

Our semantic segmentation approach learns mostly the different colors and textures between the different coral species, since the model used captures chiefly this kind of visual local features rather than shape (Geirhos et al., **2019**). Nevertheless, as we trained our encoder on 200 different coral species, where same species with different morphology are actually annotated as different semantic classes, the resulting segmentation model is also learning implicitly some of the morphological differences.

6.2 Experiments

The aim of this experiment is to learn a good feature encoder that is able to generalize on the basis of several coral reefs species to be used as a pretrained model for training on other coral data sets.

6.2.1 Set up

We trained the Deeplabv3 encoder from scratch on the CoralNet data set for 70 epochs. We set an initial learning rate of 10−310−3 with a polynomial learning rate decay schedule. Our data augmentation included: vertical and horizontal flips, contrast normalization, and random shifts and rotations. For the semantic segmentation experiments, we used the better setup so far, with our proposed modified loss and the Deeplabv3-symmetric architecture.

6.2.2 Trained encoder

The resulting trained encoder learned a balanced feature encoding of the coral domain. The mean accuracy per patch (over the 431,068 patches) of the model is 53.64, the mean accuracy per class (over the 191 different semantic classes) is 50.87 and the mean precision per class is 52.53. This result shows that the encoder has learned useful representations for the coral reef images with no class imbalance.

6.2.3 Benefits of the pretrained CoralNet encoder

Table **10** shows the effect of the pretrained encoder on the Mosaics UCSD data set, which is a medium-sized data set of four thousand training images and on the Eilat data set, which is a small data set of 100 training images. The pretraining shows two main benefits. The earlier convergence on both data sets and the improved performance of both data sets. This experiment shows the power of pretraining on deep learning. The CoralNet pretrained encoder we release will be useful for all the coral reef semantic segmentation models. Moreover, as all the deep learning classification architectures are encoders, this pretrained encoder would also benefit coral reef classification tasks. This experiment shows the results without freezing any layer and training all the network. Other experiments performed showed that freezing layers did not help the performance.

Table 10. Semantic segmentation performance of models trained from scratch and pretrained on the CoralNet data set

	Metrics			
Loss configuration	**PA**	**MIoU**	**MPA**	**Epochs to converge**
Mosaics from scratch	87.16	51.57	61.12	500
Mosaics pretrained on CoralNet	**87.82**	**53.63**	**63.74**	300
Eilat from scratch	90.02	40.65	47.61	600
Eilat pretrained on CoralNet	**90.17**	**42.45**	**50.65**	300

- *Note*: The experiment was performed on the Mosaics UCSD data set (evaluated with dense labels) and Eilat data set (evaluated with Augmented-GT).

- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.
- Bold numbers highlight the best performing method on each metric.

6.2.4 From Eilat to EilatMixx: Generalizing to the same coral domain

Having demonstrated that through pretraining and fine-tuning we can learn more general and better models, a question may arise: Can a learned model be used for the same domain but in different images or data sets without the need for retraining? This question is very interesting because it opens up the possibility for learning general models capable of being trained only once and then used for different applications for the same domain.

A quick experiment is enough to show that the answer is that it is very difficult because of coral reef variability over time and over different geographical areas (shape, sizes, color, and appearance; Zhou et al., **2018**) and that model fine-tuning is essential for achieving good segmentation results.

As detailed in Section **4**, the EilatMixx data set contains the same types of corals as the Eilat data set and both data sets are from the same geographical area. In this short experiment, we compare how a model trained on the Eilat data set performs on the EilatMixx data set without any training. We compare this method with different training approaches on the EilatMixx data set: from scratch, pretraining on the Eilat data set and using the pretrained CoralNet encoder.

Table **11** shows that the worst segmentation results are obtained with no training on the new data of EilatMixx. Although the model trained on Eilat does not reach satisfactory segmentation results on EilatMixx data, it is better than just a random solution (obtained by the mean of 10 executions with random initialization of the CNN)—which means that the Eilat data has helped to learn useful features for the EilatMixx data.

Table 11. Semantic segmentation performance of different training approaches, including no training on the target EilatMixx data

	Metrics		
Loss configuration	PA	MIoU	MPA
Evaluation on dense scores: Augmented-GT			
Random initialization (no training)	8.32	2.11	9.56
Eilat trained model (no training)	23.54	5.10	11.46
From scratch	**46.73**	10.13	16.55

	Metrics		
Loss configuration	PA	MIoU	MPA
Pretrained on Eilat	44.36	10.39	16.67
Pretrained on CoralNet	44.07	**12.45**	**21.27**
Evaluation on sparse scores: Original-GT			
Random initialization (no training)	8.39	5.78	11.13
Eilat trained model (no training)	29.02	8.21	15.24
From scratch	46.45	10.62	17.52
Pretrained on Eilat	48.19	12.68	19.74
Pretrained on CoralNet	**49.71**	**14.61**	**25.86**

- *Note*: Experiment performed on EilatMix data set.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.
- Bold numbers highlight the best performing method on each metric.

Better results are obtained after training a model on the target data set, the EilatMixx. Moreover, the models pretrained on other coral reef data sets achieve better results. This is the same conclusion as that obtained with the previous experiment on the Mosaics UCSD data set and the Eilat data set (see Table **10**). One interesting point to consider regarding pretraining is the following: the amount of pretraining data is more relevant than having data from a very close domain for pretraining, that is, pretraining on CoralNet, very large but not that similar to EilatMixx as Eilat, is the best performing option.

Figure **8** shows some visual results for the three coral reef data sets we use to get the semantic segmentation. We can see that the augmented labeling fits the coral reef images reasonably well and that the semantic segmentation obtained is good even though it has been learned from sparse labels and from a very low number of images.

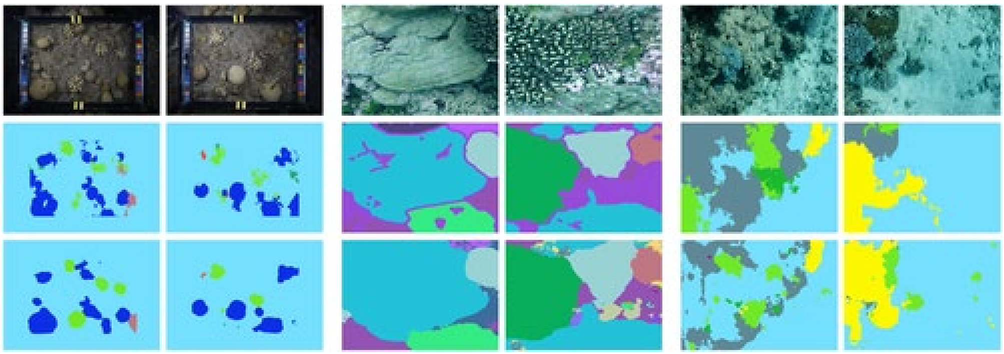


Figure 8

Visual samples of the Eilat data set (left), Mosaics UCSD data set (center), and the EilatMixx data set (right). The first row (top) corresponds to the RGB image, the center row corresponds to the augmented labeling obtained with our approach and the last row (bottom) corresponds to the semantic segmentation obtained with a model trained on the augmented labeling

7 APPLICABILITY TO NONCORAL DOMAINS

We demonstrated in previous sections that our proposed method for sparse labeling augmentation allows the training of coral reef semantic segmentation models as if training with dense labels. Other domains also suffering from lack of dense labels for semantic segmentation may benefit from our method or can take advantage of the reduction in annotation cost offered by it. This section demonstrates that our proposed method can be applied to other domains.

7.1 Data and evaluation

7.1.1 Data sets

For evaluating our labeling augmentation method, we use three data sets from different domains and with assorted objectives: the Camvid data set (urban scenarios), RIT data set (drone views), and VOC 2012 data set (general-purpose images).

- Camvid (Brostow, Fauqueur, & Cipolla, **2009**) is an autonomous driving data set with 11 different classes, frequently used to train existing state-of-the-art approaches for urban area image segmentation models.
- RIT (Kemker, Salvaggio, & Kanan, **2018**) is an aerial imagery data set with multispectral data from 18 classes. RIT does not provide test image labeling, so we evaluate its results by separating part of the evaluation set it provides.
- Pascal VOC 2012 (Everingham, VanGool, Williams, Winn, & Zisserman, **2010**) is a well-known general-purpose data set for semantic segmentation with 20 different classes.

7.1.2 Evaluation

All these data sets have dense labels and, therefore, the evaluation metrics are computed with respect to these dense labels. The sparse labels of these data sets are obtained automatically by sampling the dense labeling following a grid. The default of this simulated sparse labeling is 0.1%0.1% of the dense labels (e.g., from a 500 × 500 image, the simulated sparse ground truth contains 250 labeled pixels). We use the same metrics as in the previous evaluations (PA, MPA, and MIoU).

7.2 Approach performance on additional domains

7.2.1 Labeling augmentation quality

In this experiment, we compare the dense labels available on each data set and the results from applying our approach to augment the *simulated* sparse labeling.

Table **12** summarizes the quantitative comparison of the augmented labeling with the original dense labeling (augmentation from the 0.1% of the dense labels), showing very good results in the three different domains. As noted in Section **3.2**, our proposed augmentation method propagates existing sparse labels; therefore, it needs to have at least one labeled pixel per object or instance. The sparse labeling simulation (sampling) can miss samples from very small instances. Consequently, the PASCAL VOC 2012, the data set with bigger and fewer objects (see Figure **9**), gets the highest scores. We show that the augmented labeling obtained with our approach is very close to the original dense labels. Figure **9** shows the qualitative results of these experiments. We can see that although our approach is not perfect and introduces some noise on the labels, it gets satisfactorily similar dense labels.

Table 12. Labeling augmentation quality of our proposed method

	Metrics		
Data sets	**PA**	**MPA**	**MIoU**
Camvid	91.95	76.91	65.05
RIT	97.44	72.31	59.18
VOC 2012	96.87	95.77	93.31

- *Note*: Evaluation on the original dense labels.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.

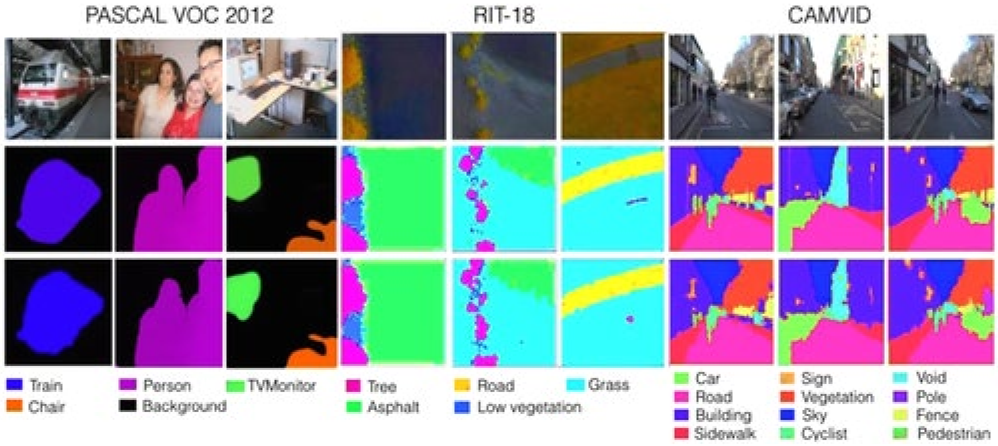


Figure 9

Examples of labeling augmentation evaluation with different data sets. Input images (top), original dense labeling (middle), and augmented labeling recovered from just 0.1% of the originally labeled pixels (bottom)

Table **13** compares our approach using different sparsity levels (different numbers of labeled pixels for the augmentation), with other recent label augmentation or propagation methods using the PASCAL VOC 2012 data set. One of these works (Vernaza & Chandraker, **2017**) uses traces as the input of the augmentation process (Traces) as well as the learned boundaries (learned by a neural network) using the RAWKS algorithm (v1) to augment the trace sparse labeling. V2 indicates the evaluation is done on 94% of the pixels, where the model is confident enough. Our baseline and previous work (Alonso et al., **2017**) use the single-level version of our approach and the same grid structure of sparse pixels as our multilevel superpixel augmentation. We show that our approach gets the highest scores when the input labeled pixels are more than 0.1% of the dense labels.

Table 13. Labeling augmentation quality of different approaches

	MIoU
Augmentation from traces	
Traces (SPCON) (Vernaza & Chandraker, 2017)	76.50
Traces (RAWKS v1) (Vernaza & Chandraker, 2017)	75.80
Traces (RAWKS v2) (Vernaza & Chandraker, 2017)	81.20

	MIoU
Augmentation from sparse pixel labels	
Baseline from 0.1% of pixels (300 pixels) (Alonso et al., 2017)	86.36
Ours from 0.01% of pixels (30 pixels)	74.40
Ours from 0.1% of pixels (300 pixels)	93.31
Ours from 1% of pixels (3,000 pixels)	97.25

- *Note*: Experiment performed on the PASCAL VOC 2012 data set. Evaluation on the original dense labels.

7.2.2 Training with augmented labels

In this experiment, we compare the quality of the segmentation obtained from a model trained on the original dense labeling and from a model trained on the augmented labeling using our augmentation method.

Table **14** shows a summary of the results using the Camvid and RIT data sets, which are the two data sets that obtained the lower augmentation scores in Table **12**. The results obtained after training with our augmented labels are comparable to training with the original dense labels. This could be expected since we already validated that the augmented labeling is very close to the original labeling. Figure **10** shows the visual comparison between the semantic segmentation results obtained with the model trained with dense labels and the model trained with augmented labels.

Table 14. Semantic segmentation performance when training on the original dense labels (dense) and our augmented labeling (augmented)

	Metrics		
Data sets	PA	MPA	MIoU
Camvid (dense)	**88.68**	**48.81**	**44.36**

	Metrics		
Data sets	**PA**	**MPA**	**MIoU**
Camvid (augmented)	87.70	46.97	42.95
RIT (dense)	**94.23**	**20.36**	**19.16**
RIT (augmented)	89.30	19.65	17.85

- *Note*: Experiment performed on the Camvid and RIT data sets. Evaluation on the original dense labels.
- Abbreviations: MIoU, mean intersection over union; MPA, mean pixel accuracy; PB, Pseudo-Boolean.
- Bold numbers highlight the best performing method on each metric.

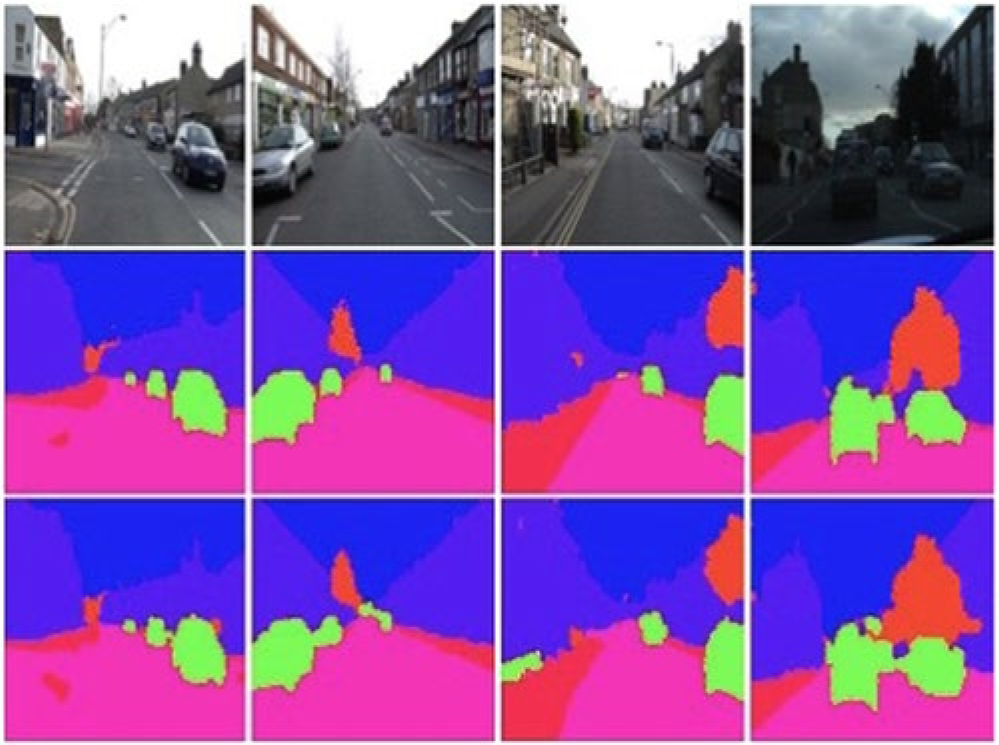


Figure 10

Semantic segmentation on Camvid. Original images (top), results using a model trained on original dense labeling (middle), and results using a model trained with our proposed augmented labeling (bottom)

The conclusions are the same as the ones obtained with the coral reef data sets. They prove that our method is both applicable to and valuable for other domains and applications.

8 CONCLUSION

Existing acquisition systems, such as autonomous robots or remote-controlled platforms, have made it possible to acquire large amounts of environmental monitoring data, but methods to automatically process these huge amounts of data remain an open challenge in many domains. The contributions presented in this study help tackle this challenge, especially when there are not enough resources to label large amounts of detailed training data. The new tools provided by our work enable further work on scene understanding for numerous robotics applications such as remote monitoring from UAVs or underwater devices.

Our main contribution is an approach to enable effective training of semantic segmentation models from sparsely labeled data. Our approach propagates the sparse labels (sparse pixel annotations) by an iterative method based on superpixel segmentation techniques. Our multilevel approach outperforms by an average of 11% of the MIoU compared with previous single-level approaches, including our earlier version of this study. The exhaustive experimentation presented here shows the effect of the various method parameters.

The limitations of our approach come from the trade-off between number of levels in the segmentation versus computational cost. A higher number of levels yields better performance but considerably increases execution time. Another limitation to consider is that our approach relies on superpixel techniques; therefore, the image has to have clear gradients for good performance. The results in this study demonstrate that our propagated labels are highly reliable for training, as the semantic segmentation models trained with them result in performance equivalent to training with ground truth dense labels (fully annotated images).

Our core experimentation was run on a realistic and challenging scenario—underwater coral reef monitoring data. Besides the well-known environmental value of these underwater regions and consequent interest in their monitoring, they present a challenging and real-world use case where most of the available labeling efforts, made by marine biology experts, consist of sparse labels. Although the experimentation in this study is focused on underwater imagery, we also demonstrated the applicability of our approach to different applications with data from different robotic acquisition platforms (aerial surveillance and urban driving scenarios).

Further, this study contributes to the field of automatic underwater image processing as follows. We present a comparison of the main semantic segmentation architectures run in an underwater domain, in particular, coral reef image segmentation. We not only present a detailed comparison of common architectures and loss functions for the coral reef segmentation use case but also propose a more suitable variation of the cross-entropy loss for this task. We observed that our modified version of the cross-entropy loss enhances the results by 2% of the MIoU. Our experiments demonstrate that this encoder helps segmentation models for new coral reef scenarios, having little training labeled data available, learn better. Specifically, we are able to train models in half the time (early convergence) and enhance results by 4% MIoU. We also show that when using our pretrained encoder, we get better results than pretraining the encoder with data from the same geographical localization. This study releases several useful tools for the research community, namely, the obtained generic encoder pretrained on over half a million images of corals, all the data including new labeled data for coral segmentation, and the tools developed (to facilitate replication and training on new coral data).

We aim to expand our study and proposed pipeline to 3D input data, since many of the monitoring acquisition systems now provide 3D scans of the environment, rather than regular images. We also plan on disseminating the presented tools to researchers in coral reef analysis.

ACKNOWLEDGMENTS

The authors would like to thank NVIDIA Corporation for the donation of the Titan Xp GPUs used in this study. We thank Aviad Avni for fieldwork assistance and the Interuniversity Institute for Marine Sciences in Eilat for making their facilities available to us. This project was partially funded by the Spanish Government project PGC2018-098817-A-I00, Aragón Regional Government (DGA T45_17R/FSE), and the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skolodowska-Curie grant agreement no. 796025 to G. E; T. T. was supported by the Israel Ministry of National Infrastructures, Energy, and Water Resources (Grant 218-17-008) and Israel Science Foundation (Grant 680/18); M. Y. was supported by the PADI Foundation application no. 32618 and the Murray Foundation for student research.

[1] https://coralnet.ucsd.edu/

[2] Tools, model, and data publicly available on https://sites.google.com/a/unizar.es/semanticseg/home.

[3] https://labelbox.com/

[4] https://github.com/Shathe/ML-Superpixels

[5] https://github.com/tensorflow/models/tree/master/research/deeplab

[6] https://datadryad.org/resource/doi:10.5061/dryad.t4362

[7] Data sets available on https://sites.google.com/a/unizar.es/semanticseg/home.

REFERENCES

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(11), 2274– 2282.
- Akkaynak, D., & Treibitz, T. (2018). A revised underwater image formation model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6723– 6732.
- Akkaynak, D., & Treibitz, T. (2019). Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1682– 1691.

- Alonso, I., Cambra, A., Munoz, A., Treibitz, T., & Murillo, A. C. (2017). Coral-segmentation: Training dense labeling models with sparse ground truth. In *IEEE International Conference on Computer Vision Workshops*, pp. 2874–2882.
- Alonso, I., & Murillo, A. C. (2018). Semantic Segmentation from Sparse Labeling Using Multi-Level Superpixels. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5785-5792).
- Anthony, K. R. (2016). Coral reefs under climate change and ocean acidification: Challenges and opportunities for management and policy. *Annual Review of Environment and Resources*, **41**(1), 59– 81.
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, **39**(12), 2481– 2495.
- Beijbom, O., Edmunds, P. J., Kline, D. I., Mitchell, B. G., & Kriegman, D. (2012). Automated annotation of coral reef survey images. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1170– 1177).
- Beijbom, O., Treibitz, T., Kline, D. I., Eyal, G., Khen, A., Neal, B., Loya, Y., Mitchell, B. G., & Kriegman, D. (2016). Improving automated annotation of benthic survey images using wide-band fluorescence. *Scientific Reports*, **6**, 1– 11.
- Berman, D., Treibitz, T., & Avidan, S. (2017). Diving into haze-lines: Color restoration of underwater images. In Proc. British Machine Vision Conference (BMVC) (Vol. 1, No. 2).
- Berman, M., Triki, A. R., & Blaschko, M. B. (2018). The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4413– 4421).
- Brostow, G. J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, **30**(2), 88– 97.
- Bryant, D., Rodriguez-Ramirez, A., Phinn, S., González-Rivero, M., Brown, K., Neal, B., Hoegh-Guldberg, O., & Dove, S. (2017). Comparison of two photographic methodologies for collecting and analyzing the condition of coral reef ecosystems. *Ecosphere*, **8**(10), e01971.
- Cesar, H. S. (2000). Coral reefs: Their functions, threats and economic value. Collected Essays on the Economics of Coral Reefs, pp. 14– 39.
- Chen, L.-C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV) (pp. 801– 818).

- Chen, P.-Y., Chen, C.-C., Chu, L., & McCarl, B. (2015). Evaluating the economic damage of climate change on global coral reefs. *Global Environmental Change*, **30**, 12– 20.
- Conrad, C., Mertz, M., & Mester, R. (2013). Contour-relaxed superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 280– 293.
- Connell, J. H., Hughes, T. P., Wallace, C. C., Tanner, J. E., Harms, K. E., & Kerr, A. M. (2004). A long-term study of competition and diversity of corals. *Ecological Monographs*, **74**(2), 179– 210.
- DeBoer, P.-T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research*, **134**(1), 19– 67.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248– 255). IEEE.
- Durand, T., Mordan, T., Thome, N., & Cord, M. (2017). Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 642– 651).
- Edwards, C. B., Eynaud, Y., Williams, G. J., Pedersen, N. E., Zgliczynski, B. J., Gleason, A. C., Smith, J. E., & Sandin, S. A. (2017). Large-area imaging reveals biologically driven non-random spatial patterns of corals at a remote reef. *Coral Reefs*, **36**(4), 1291– 1305.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, **88**(2), 303– 338.
- Eyal, G., Wiedenmann, J., Grinblat, M., D'Angelo, C., Kramarsky-Winter, E., Treibitz, T., … Loya, Y. (2015). Spectral diversity and regulation of coral fluorescence in a mesophotic reef habitat in the Red Sea. *PLOS One*, **10**(6), e0128697.
- Fabricius, K. E. (2005). Effects of terrestrial runoff on the ecology of corals and coral reefs: Review and synthesis. *Marine Pollution Bulletin*, **50**(2), 125– 146.
- Finney, S., & Stephen, J. (2005). Photo mosaics in shallow water environments: Challenges and results. *WIT Transactions on the Built Environment*, **79**, 1– 9.
- Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*.
- Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*.

- Gomes-Pereira, J. N., Auger, V., Beisiegel, K., Benjamin, R., Bergmann, M., Bowden, D., … Santos, R. S. (2016). Current and future trends in marine image annotation software. *Progress in Oceanography*, **149**, 106– 120.

- González-Rivero, M., Bongaerts, P., Beijbom, O., Pizarro, O., Friedman, A., Rodriguez-Ramirez, A., … Hoegh-Guldberg, O. (2014). The Catlin Seaview survey-kilometre-scale seascape assessment, and monitoring of coral reef ecosystems. *Aquatic Conservation: Marine and Freshwater Ecosystems*, **24**(S2), 184– 198.

- Goreau, T. F. (1959). The ecology of Jamaican coral reefs i. species composition and zonation. *Ecology*, **40**(1), 67– 90.

- Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic data for text localisation in natural images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2315– 2324).

- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *International Conference on Computer Vision*, pp. 2980– 2988.

- He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, Springer, pp. 346– 361.

- Hennige, S. J., Smith, D. J., Walsh, S.-J., McGinley, M. P., Warner, M. E., & Suggett, D. J. (2010). Acclimation and adaptation of scleractinian coral communities along environmental gradients within an indonesian reef system. *Journal of Experimental Marine Biology and Ecology*, **391**(1–2), 143– 152.

- Hodgson, J. C., Baylis, S. M., Mott, R., Herrod, A., & Clarke, R. H. (2016). Precision wildlife monitoring using unmanned aerial vehicles. *Scientific Reports*, **6**, 22574.

- Hoegh-Guldberg, O., Mumby, P. J., Hooten, A. J., Steneck, R. S., Greenfield, P., Gomez, E., … Hatziolos, M. E. (2007). Coral reefs under rapid climate change and ocean acidification. *Science*, **318**(5857), 1737– 1742.

- Hu, R., Dollár, P., He, K., Darrell, T., & Girshick, R. (2018). Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4233-4241.

- Hughes, T. P., Baird, A. H., Bellwood, D. R., Card, M., Connolly, S. R., Folke, C., … Roughgarden, J. (2003). Climate Change, human impacts, and the resilience of coral reefs. *Science*, **301**(5635), 929– 933.

- Hughes, T. P., Barnes, M. L., Bellwood, D. R., Cinner, J. E., Cumming, G. S., Jackson, J. B., … Scheffer, M. (2017). Coral reefs in the anthropocene. *Nature*, **546**(7656), 82.

- Hughes, T. P., Kerry, J. T., Baird, A. H., Connolly, S. R., Dietzel, A., Eakin, C. M., … Torda, G. (2018). Global warming transforms coral reef assemblages. *Nature*, **556**(7702), 492.

- Huston, M. (1985). Patterns of species diversity on coral reefs. *Annual Review of Ecology and Systematics*, **16**(1), 149– 177.
- Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1175– 1183). IEEE.
- Kaiser, L., Gomez, A. N., & Chollet, F. (2017). Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*.
- Kemker, R., Salvaggio, C., & Kanan, C. (2018). Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, **145**, 60– 77.
- King, A., Bhandarkar, S. M., & Hopkinson, B. M. (2018). A comparison of deep learning methods for semantic segmentation of coral reef survey images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1394– 1402.
- Koh, L. P., & Wich, S. A. (2012). Dawn of drone ecology: Low-cost autonomous aerial vehicles for conservation. *Tropical Conservation Science*, **5**(2), 121– 132.
- Kohler, K. E., & Gill, S. M. (2006). Coral point count with excel extensions: A visual basic program for the determination of coral and substrate coverage using random point count methodology. *Computers & Geosciences*, **32**(9), 1259– 1269.
- Kolesnikov, A., & Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In European Conference on Computer Vision (pp. 695– 711).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, **25**, 1097– 1105.
- Laxton, J., & Stablum, W. (1974). Sample design for quantitative estimation of sedentary organisms of coral reefs. *Biological Journal of the Linnean Society*, **6**(1), 1– 18.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, **521**(7553), 436.
- Liu, M.-Y., Tuzel, O., Ramalingam, S., & Chellappa, R. (2011). Entropy rate superpixel segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431– 3440.
- Loya, Y. (1972). Community structure and species diversity of hermatypic corals at Eilat, Red Sea. *Marine Biology*, **13**(2), 100– 123.

- Luc, P., Neverova, N., Couprie, C., Verbeek, J., & LeCun, Y. (2017). Predicting deeper into the future of semantic segmentation. In *International Conference on Computer Vision*, p. 10.

- Ludvigsen, M., Sortland, B., Johnsen, G., & Singh, H. (2007). Applications of geo-referenced underwater photo mosaics in marine biology and archaeology. *Oceanography*, **20**(4), 140– 149.

- Manderson, T., Li, J., Dudek, N., Meger, D., & Dudek, G. (2017). Robotic coral reef health assessment using automated image analysis. *Journal of Field Robotics*, **34**(1), 170– 187.

- Mary, N. A. B., & Dharma, D. (2017). Coral reef image classification employing improved LDP for feature extraction. *Journal of Visual Communication and Image Representation*, **49**, 225– 242.

- Mičušík, B., & Košecká, J. (2010). Multi-view superpixel stereo in urban environments. *International Journal of Computer Vision*, 106– 119.

- Milioto, A., Lottes, P., & Stachniss, C. (2018). Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. In *2018 IEEE International Conference on Robotics and Automation* (pp. 2229– 2235). IEEE.

- Moberg, F., & Folke, C. (1999). Ecological goods and services of coral reef ecosystems. *Ecological Economics*, **29**(2), 215– 233.

- Moniruzzaman, M., Islam, S. M. S., Bennamoun, M., & Lavery, P. (2017). Deep learning on underwater marine object detection: A survey. In *International Conference on Advanced Concepts for Intelligent Vision Systems* (pp. 150– 160). Springer.

- Morin, P. J. (2009). *Community ecology* (2, pp. 1– 424). United States: Wiley-Blackwell.

- Reaka-Kudla, M. L. (1997). The global biodiversity of coral reefs: A comparison with rain forests. *Biodiversity II: Understanding and Protecting our Biological Resources*, **2**, 551.

- Roberts, C. M., McClean, C. J., Veron, J. E., Hawkins, J. P., Allen, G. R., McAllister, D. E., … Werner, T. B. (2002). Marine biodiversity hotspots and conservation priorities for tropical reefs. *Science*, **295**(5558), 1280– 1284.

- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

- Sanders, H. L. (1968). Marine benthic diversity: A comparative study. *The American Naturalist*, **102**(925), 243– 282.

- Schlichting, C. D., & Pigliucci, M. (1998). Phenotypic evolution: A reaction norm perspective. United States: Sinauer Associates Incorporated.

- Shihavuddin, A., Gracias, N., Garcia, R., Gleason, A. C., & Gintert, B. (2013). Image-based coral reef classification and thematic mapping. *Remote Sensing*, **5**(4), 1809– 1841.
- Singh, H., Howland, J., & Pizarro, O. (2004). Advances in large-area photo mosaicking underwater. *IEEE Journal of Oceanic Engineering*, **29**(3), 872– 886.
- Stoddart, D. R. (1969). Ecology and morphology of recent coral reefs. *Biological Reviews*, **44**(4), 433– 498.
- Stutz, D., Hermans, A., & Leibe, B. (2018). Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, **166**, 1– 27.
- Sun, B., & Saenko, K. (2016). Deep CORAL: correlation alignment for deep domain adaptation. *CoRR*, abs/1607.01719.
- Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE international conference on computer vision (pp. 843– 852).
- Tighe, J., & Lazebnik, S. (2010). Superparsing: scalable nonparametric image parsing with superpixels. In *European Conference on Computer Vision*.
- Todd, P. A. (2008). Morphological plasticity in scleractinian corals. *Biological Reviews*, **83**(3), 315– 337.
- Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., & Geiger, A. (2017). Sparsity invariant CNNs. In *IEEE International Conference on 3D Vision*, pp. 11– 20.
- Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., & Van Gool, L. (2012). SEEDS: Superpixels extracted via energy-driven sampling. In *European Conference on Computer Vision*, pp. 13– 26.
- Venkitasubramanian, A. N., Tuytelaars, T., & Moens, M.-F. (2016). Wildlife recognition in nature documentaries with weak supervision from subtitles and external data. *Pattern Recognition Letters*, **81**(C), 63– 70.
- Vernaza, P., & Chandraker, M. (2017). Learning random-walk label propagation for weakly-supervised semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7158– 7166).
- Visbeck, M. (2018). Ocean science research is key for a sustainable future. *Nature Communications*, **9**(1), 690.
- Weinberg, S. (1981). A comparison of coral reef survey methods. *Bijdragen tot de Dierkunde*, **51**(2), 199– 218.
- Wigness, M. (2018). Superlabel: A superpixel labeling interface for semantic image annotation. Technical report. Adelphi, MD: Army Research Lab.

- Wong, J. M., Kee, V., Le, T., Wagner, S., Mariottini, G.-L., Schneider, A., … Torralba, A. (2017). Segicp: Integrated deep semantic segmentation and pose estimation. In *International Conference on Intelligent Robots and Systems*.
- Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.
- Zhang, Y., Hartley, R., Mashford, J., & Burn, S. (2011). Superpixels via pseudo-boolean optimization. In *IEEE International Conference on Computer Vision*, pp. 1387– 1394.
- Zhou, Z., Ma, L., Fu, T., Zhang, G., Yao, M., & Li, M. (2018). Change detection in coral reef environment using high-resolution images: Comparison of object-based and pixel-based paradigms. *ISPRS International Journal of Geo-Information*, **7**(11), 441.
- Zhu, H., Meng, F., Cai, J., & Lu, S. (2016). Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, **34**, 12– 27.
- Zweifler, A., Akkaynak, D., Mass, T., & Treibitz, T. (2017). In situ analysis of coral recruits using fluorescence imaging. *Frontiers in Marine Science*, **4**, 273.