International Journal of Computer Vision (2023) 131:1367–1388
https://doi.org/10.1007/s11263-023-01755-4

Context-Driven Detection of Invertebrate Species in Deep-Sea Video

R. Austin McEver[1] · Bowen Zhang[1] · Connor Levenson[1] · A S M Iftekhar[1] · B. S. Manjunath[1]

Received: 28 April 2022 / Accepted: 10 January 2023 / Published online: 22 February 2023

Abstract
Each year, underwater remotely operated vehicles (ROVs) collect thousands of hours of video of unexplored ocean habitats revealing a plethora of information regarding biodiversity on Earth. However, fully utilizing this information remains a challenge as proper annotations and analysis require trained scientists' time, which is both limited and costly. To this end, we present a Dataset for Underwater Substrate and Invertebrate Analysis (DUSIA), a benchmark suite and growing large-scale dataset to train, validate, and test methods for temporally localizing four underwater substrates as well as temporally and spatially localizing 59 underwater invertebrate species. DUSIA currently includes over ten hours of footage across 25 videos captured in 1080p at 30 fps by an ROV following pre-planned transects across the ocean floor near the Channel Islands of California. Each video includes annotations indicating the start and end times of substrates across the video in addition to counts of species of interest. Some frames are annotated with precise bounding box locations for invertebrate species of interest, as seen in Fig. 1. To our knowledge, DUSIA is the first dataset of its kind for deep sea exploration, with video from a moving camera, that includes substrate annotations and invertebrate species that are present at significant depths where sunlight does not penetrate. Additionally, we present the novel context-driven object detector (CDD) where we use explicit substrate classification to influence an object detection network to simultaneously predict a substrate and species class influenced by that substrate. We also present a method for improving training on partially annotated bounding box frames. Finally, we offer a baseline method for automating the counting of invertebrate species of interest.

Keywords Context driven · Substrate classification · Deep sea · Invertebrate classification · Underwater · Video dataset

1 Introduction

Marine scientists spend enormous amounts of resources on understanding and studying life in our oceans. These studies hold numerous benefits for environmental protection and scientific advancement, including the ability to identify areas of the ocean where certain habitats and substrates exist and where certain species gather. As scientists better understand biodiversity in the oceans and where in the ocean life flourishes, they can begin working toward more focused conservation efforts with those areas in mind. Further, scientists can revisit those same areas and perform surveys in the future to monitor how life is changing in the ocean as a result of conservation efforts (Fig. 1).

A common method for studying underwater habitats consists of planning underwater routes, called transects, then following those paths and recording the environment either by a diver with a camera or using an underwater ROV (Shester et al., 2017; Drap et al., 2015). Once the transects have been recorded and videos matched with their GPS locations, common annotation methods require researchers to review each video several times, annotating the substrates that the camera passes over in the first few annotation passes, then counting invertebrates in another pass, and then counting fish species in a final pass to give a better idea of where in the ocean which substrates exist and where different species live. This information is vital to determining species hotspots and finding

Communicated by SILVIA ZUFFI.

✉ R. Austin McEver
mcever@ucsb.edu

Bowen Zhang
bowen68@ucsb.edu

Connor Levenson
clevenson@ucsb.edu

A S M Iftekhar
iftekhar@ucsb.edu

B. S. Manjunath
manj@ucsb.edu

[1] University of California, Santa Barbara, CA 93106, USA

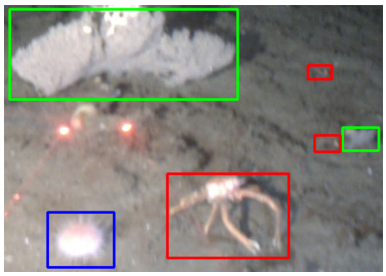

Fig. 1 Cropped frame from DUSIA with examples of three classes of interest: fragile pink urchin (blue), gray gorgonian (green), and squat lobster (red). Variations in perspective, occlusion, and size can create large differences across appearances of species individuals and make some individuals, like small squat lobsters, almost invisible, especially in a single frame. Crop shown is 690x487 pixels from a 1920x1080 frame (Color figure online)

ways to protect the environment while also meeting human needs for usage of our oceans. These studies ultimately lead to new discoveries as they facilitate exploration of unknown oceanic regions. Currently, however, the sheer amount of data researchers collect can be overwhelmingly expensive and difficult to annotate and utilize as their annotation methods' multiple passes can push annotations times to many times the duration of the video. Additionally, researchers spend a lot of time sifting through videos of just bare substrate (like rocks or mud) with no visible life, and methods that can help tell where there is no life may aid researchers in more quickly filtering those sections of video out of invertebrate counting.

Computer vision and machine learning models can significantly aid in managing, utilizing, analyzing, and understanding these videos, ultimately reducing the overall costs of these studies and freeing researchers from tedious annotation tasks. However, developing and training these models require annotated data. Further, the types of annotations generated and used by domain scientists do not directly correspond with the typical types of annotations generated and used by computer vision researchers, requiring new approaches to learning from video data and their annotations.

As a step toward advancement in efficiently computationally analyzing videos from a marine science setting, we introduce DUSIA, a real world scientific dataset including videos collected and annotated by marine scientists who directly use a superset of these videos to advance their own research and exploration. To our knowledge, DUSIA is the first public dataset to contain videos recorded in this challenging moving-camera setting where an underwater ROV drives and records over the ocean floor. This dataset allows us to create solutions to a host of difficult computer vision problems that have not yet been explored such as classifying and temporally localizing underwater habitats and substrates, counting and tracking invertebrate species as they appear in ROV video, and using these explicit substrate and habitat classifications to help detect and classify invertebrate species. Further, the types of annotations provided in DUSIA differ from those of typical computer vision datasets, requiring new approaches to learning.

Our contributions can be summarized as follows:

- DUSIA provides the first publicly available dataset of annotated, full-length videos captured via an underwater ROV. DUSIA's videos are annotated by expert marine scientists with temporal labels indicating substrates, count labels for 59 invertebrate species, partial bounding box labels for ten invertebrate species of interest in the training set, and full bounding box labels for those species of interest in the validation and testing sets.
- We introduce the novel Context-Driven Detector (CDD), which uses implicit context representations and explicit context labels to improve bounding box detections. In our case, context refers to explicit class labels of the background. Specifically, our context labels describe the substrate present on the ocean floor, which determine the environment and habitat in which the organisms live. In natural images, context might refer to indoor vs outdoor images or subcategories within such as school, office, library, or supermarket.
- We propose Negative Region Dropping, an approach for improving performance of an object detector trained on a dataset with partially annotated images.
- Finally, we offer a baseline method for counting invertebrate species individuals in this challenging setting using a detection plus tracking pipeline.

In Sect. 2 we review other datasets and methods with similar data and highlight how DUSIA differs from previous datasets. Next, in Sect. 3 we discuss the contents and collection of DUSIA's data and annotations. Section 4 describes some of the tasks for which DUSIA can be used, and Sect. 5 discusses our approaches to those tasks including the novel CDD, Negative Region Dropping, and baseline tracking method. Section 6 describes our experiments and results, and Sect. 7 discusses our findings.

2 Related Works

Analyzing underwater animals and habitats remains a challenge for computer vision models. Marine scientists collect a wide variety of visual data for an even wider variety of tasks, so when it comes to solving specific tasks, there often exists a scarcity of well-annotated underwater data. Although there are a few efforts from the computer vision community to collect and annotate underwater data (Pedersen et al., 2019; King et al., 2018; Boom et al., 2014; Marini et al., 2018; Joly et al., 2014), it is hardly enough to tackle this daunting prob-

lem, and few of these efforts collect data in the same way or provide annotations for the same goals. In general, collecting underwater image or video data is far more difficult than land data and day to day images of common objects. Collecting underwater data is so difficult, in fact, that Ishiwaka et al. (2021) proposed a method for generating synthetic datasets. DUSIA aims to be a collaborative, comprehensive effort to guide the exploration and automated analysis of underwater ecosystems.

2.1 Underwater Marine Datasets

Many of the existing underwater marine datasets are developed in order to detect and recognize the various behaviors, or simply presence, of fish (Konovalov et al., 2019; Måløy et al., 2019; Boom et al., 2014; Joly et al., 2014; Levy et al., 2018). Numerous current works (Konovalov et al., 2019; Måløy et al., 2019; Levy et al., 2018; Ditria et al., 2020) have validated their fish detection and fish behavior recognition models on these datasets. Interestingly, these methods mainly focus on developing novel data-hungry algorithms, but the data on which the algorithms perform is limited by its static perspective. For example, Måløy et al. (2019) proposed a dual spatial-temporal recurrent network, but the algorithm is trained and tested on a dataset that is constrained by having no camera movement and working in a covered area. Similarly, Konovalov et al. (2019) augments the dataset of underwater fish images that they use with the underwater non-fish images from VOC2012 (Everingham et al., 2015) by restricting their model to generating only binary (fish vs. no fish) predictions. In the same way, (Ditria et al., 2020; Levy et al., 2018) confined their models to do analysis only on one single fish. Similarly, Marini et al. (2018) works on automating the counting of fish without distinguishing among different species. In contrast, DUSIA provides dynamic, high definition ROV video showcasing a rich and varied environment with many species occurring in intermingling groups.

Additionally, unlike existing datasets, a novel feature of DUSIA is the utilization of explicit, human-annotated, contextual information such as substrates or habitat in the analysis workflow. Such contextual information can play a vital role in making accurate predictions, especially in the case of identifying fish or other marine animals. Recently, Rashid and Chennu (2020) has developed a large scale dataset for habitat mapping using both RGB images and hyperspectral images. This dataset contains a large number of annotated images for classifying different coral reef habitats, but marine animal information is not included in this dataset. DUSIA, in contrast, is unique in this aspect, as it has both explicit substrate and invertebrate annotations. Tables 1 and 2 highlight the differences in many underwater image and video datasets.

2.2 Methodologies

Beery et al. (2020) propose Context R-CNN to utilize long-term and short-term temporal context to improve recognition in passive monitoring deployments, though they lack explicit labels for the background context of their data. Because their data is collected via static cameras, the background context is unchanging, does not have explicit labels, and may not contribute much to their detection.

As mentioned in the previous section, recently, different works have developed deep learning-based algorithms to detect marine species (mostly fishes). Li et al. (2015) uses a Fast-RCNN (Girshick, 2015) based network to classify twelve different species of fish. Salman et al. (2016) present a deep network to detect fish in 32×32 size video frames. Siddiqui et al. (2018) use a pre-trained object detection CNN network as a generalized feature extractor. The extracted features are then fed to an SVM (support vector machine) for classification of fish.

Our baseline method aims to alleviate some of these methods' shortcomings by using explicit substrate predictions to enhance species detections.

3 Dataset

DUSIA consists of over 10 h of footage captured from preplanned transects along the ocean floor near the Channel Islands of California. This includes 25 HD videos recorded using RGB video cameras attached to an observation class ROV equipped with multiple lighting fixtures recording at depths between 100 and 400 ms. Three of the 25 videos do not contain species of interest, so they are excluded from experiments presented in this paper. DUSIA's videos are part of a large collection, and we plan to release more similar videos from different excursions in the future.

DUSIA's videos can assist in studies of the 57 annotated invertebrate species because many of those species are widely distributed along the west coast of North America and beyond. For example, the fragile pink urchin, Strongylocentrotus droebachiensis, inhabits the upper continental slope along the entire eastern North Pacific from Alaska to Baja California, ranging in depth from 200–1200 m off central California (Taylor et al., 2014). Several species of squat lobster (Munidopsis spp.) are also common across the Eastern Pacific, and are similar looking such that collections would be needed to identify them to species (Wicksten, 1989). The yellow gorgonian, Acanthogorgia gracillima, is also found across the North Pacific from at least Japan to California (Horvath, 2019).

Recently, the diversity and abundance of megafaunal taxa, such as those labelled in DUSIA, have been identified as a

Table 1 Underwater datasets with labelled images

Image datasets	Environment	Recording type	# Images	Annotation type	Class description
Barrett et al.[1]	Reef	AUV	1258	Point	> 10 species and abiotic elements
Beijbom et al.[2]	Reef	Photoquadrat	2055	Point	∼ 30 coral
Anantharajah et al.[3]	–	Various	3960	Bbox	468 fish species
BENTHOZ-2015[4]	Deep sea	AUV	9874	Point	> 70 invertebrate and substrate
Bett and Ruhl[5]	Deep sea	Static	1047	Camera location	No species labels
Jager et al.[6]	–	Various	794	Bbox	12 fish species
Beijbom et al.[7]	Reef	Photoquadrat	212	Points	10 coral and substrate
J-EDI[8]	Deep sea	ROV	1,500,000	Image level	> 20 fish and invertebrate
King et al.[9]	Reef	Static	413	Segmentation	10 invertebrate and substrate
Marini et al.[10]	Ocean (20 m deep)	Static	20,000	Bbox, contour	Fish/not fish
Levy et al.[11]	Sea	Aerial	272	Bbox	Shark, ray, diver
Pedersen et al.[12]	Brackish strait	Static	14,518	Bbox	Big/small fish, 4 invertebrate
MOUSS[13]	Ocean floor	Static	159	Bbox	∼ 10 fish and scallop
AFSC[13]	Ocean	ROV	571	Points	∼10 fish and scallop
MBARI[13]	Ocean floor	–	666	Bbox	∼10 fish and scallop
NWFSC[13]	Ocean floor	ROV	123	Points	∼10 fish and scallop
Langenkämper et al.[14]	Deep sea	Various	20,000	Image level	23 megafauna morphotypes
Ditria et al.[15]	Seagrass meadows	–	6080	Segmentation	Fish
Rashid et al.[16]	Reef	HyperDiver	147	Contour	47 sessile biota and substrate
fathomNet[17]	Deep sea	Various	85,000	Bbox	> 2,000 concepts

Numbers of images are approximate.
Abbreviations: Bbox, bounding box; AUV, autonomous underwater vehicle. Citations are 1: (Barrett et al., 2011), 2: (Beijbom et al., 2012), 3: (Anantharajah et al., 2014), 4: (Bewley et al., 2015), 5: (Bett & Ruhl, 2015), 6: (Jäger et al., 2015), 7: (Beijbom et al., 2016), 8: (Jamstec e-library of deep-sea images, 2016), 9: (King et al., 2018), 10: (Marini et al., 2018), 11: (Levy et al., 2018), 12: (Pedersen et al., 2019), 13: (Richards et al., 2019), 14: (Langenkämper et al., 2020), 15: (Ditria et al., 2020), 16: (Rashid & Chennu, 2020), 17: (Katija et al., 2022)

Table 2 Underwater datasets with labelled videos

Video Datasets	Environment	Recording type	Total footage	Res.	Frame rate	Annotation description	Class description
Boom et al.[1]	Reef	Static	88 kh	240p, 480p	5 fps	Contour, fish track	15 fish species
Joly et al.[2]	Reef	Static	117 kh	240p	8 fps	Bbox	Fish
Maloy et al.[3]	Farming site	Static	6 h	244 × 244	24 fps	Video class	Feeding/not feeding
Šiaulys et al.[4]	Ocean floor (3–65 m deep)	AUV	23 min	1080p	5 fps	Segmentation	12 taxons
DUSIA (ours)	Ocean floor (100–400 m)	ROV	10 h	1080p	30 fps	Bbox, substrate classes, and CABOF	57 invertebrate, 4 substrate

Note that static camera datasets include footage from cameras that are fixed underwater and run continuously for days on end.
Abbreviations: Res., resolution; bbox, bounding box; AUV, autonomous underwater vehicle. Citations are 1: (Boom et al., 2014), 2: (Joly et al., 2014), 3: (Måløy et al., 2019), 4: (Šiaulys et al., 2021)

high priority essential variable for understanding changes in marine ecosystems (Danovaro et al., 2020).

3.1 Data Collection

Surveys of wildlife on the ocean floor generally start with planning a group of paths, called transects, across some region in order to efficiently cover and survey one section of the ocean (Shester et al., 2017); however, to protect these fragile ecosystems, DUSIA does not make specific GPS coordinates publicly available.

Some surveys use scuba divers to collect video along transects, but DUSIA covers larger, deeper areas using an ROV attached to a 77-foot catamaran. During the collection pro-

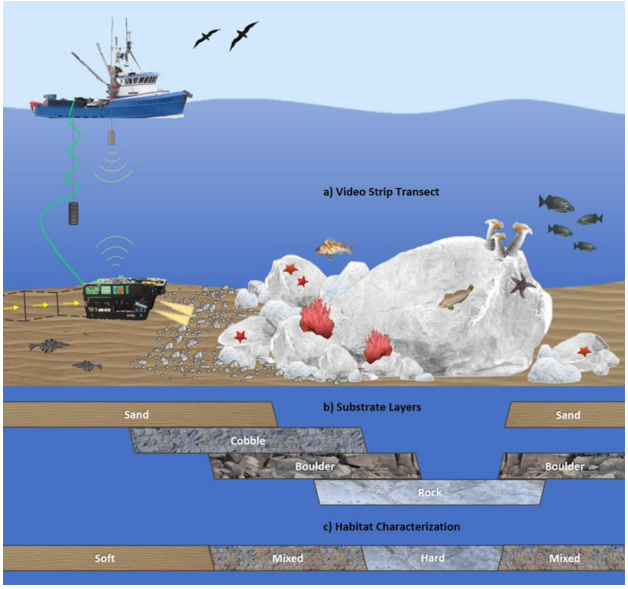

Fig. 2 Illustration of the ROV attached to the catamaran, substrate layers, and habitat characterization. Substrates are divided into soft (mud, cobble and sand), hard (rock and boulder), or mixed (a combination of any soft and hard substrates). Illustration courtesy of Marine Applied Research and Exploration (MARE) Group

cess, the ROV is attached via cable to the catamaran. Once the boat arrives near the beginning of the desired transects, the ROV is placed in the water and remains on a long leash attached to the boat such that the catamaran can follow the transects roughly while the ROV follows its path more precisely via inputs from a remote operator on the boat who makes use of the ROV's cameras, lights, GPS, and other instruments that indicate the ROV's location relative to the boat, which allows for computing its GPS location. Figure 2 roughly illustrates the ROV rig used for data collection.

3.2 Substrate Classes and Annotations

After the collection stage, researchers return to a laboratory where they review, analyze, and annotate each video. DUSIA includes four different substrates: boulder, cobble, mud, and rock. An illustration of each one is shown in Fig. 2 and frames from the dataset are shown in Fig. 3. The difference between each depends on the nature of the material makeup of the ocean floor. A description of each substrate can be found in Tables 3, and 4 shows a toy example of the annotation format.

Each of these substrates may overlap such that a given frame can have multiple substrate labels if enough of multiple substrates are visible. The annotation process includes multiple passes, one for each substrate, where the annotators indicate the start and end times of each substrate occurrence. This arduous process can be alleviated by our methods.

3.3 Invertebrate Classes and Annotations

Once the substrate annotations are completed, scientists make yet another pass over each video, this time annotating invertebrate species, often referencing substrate labels as certain species have a tendency to occur in certain substrates. When a group or individual of a species touches the bottom of the video frame, they pause the video, count the species touching the bottom of the frame, and make note of the time stamp at which the count occurred, giving domain researchers insight into where in the video, in the ocean, and in which substrate, each species tends to occur. We refer to these labels as CABOF, Count At the Bottom of the Frame, labels. Figure 4 illustrates the CABOF label collection procedure.

Count labels provide guidance in learning to classify and detect invertebrate species, they ensure that species individuals are not counted multiple times, and a human could use

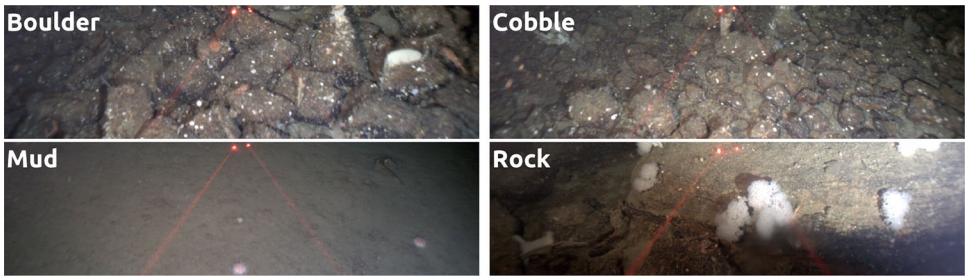

Fig. 3 Example frames each containing just one substrate each, indicated by the in frame text

Table 3 Description of the four substrates present in DUSIA

Substrate	Description
Boulder	Rocky substrate larger than 25 cm in diameter that is detached and clearly movable
Cobble	Rocky substrate that is 6 to 25 cm in diameter
Mud	Very fine sediments that stay suspended in the water when disturbed (loss of visibility)
Rock	Consolidated rocky substrates that appear attached to the bottom and not movable

Table 4 Example of combined substrate and CABOF, Count At the Bottom of the Frame, annotations. Substrates are labeled with beginning and end times, and invertebrate CABOF labels include a single timestamp shown in the Begin column and count

Annotation	Begin	End	Count
Boulder	0:00:20	0:00:25	
FPU	0:00:21		2
Cobble	0:00:23	0:01:30	
Mud	0:00:40	0:01:20	
SL	0:00:49		1
SL	0:00:51		3
Rock	0:01:00	0:03:50	
Mud	0:02:10	0:02:15	

these labels to learn to label further videos. However, current computer vision methods do not perform as well with weak supervision as they do with strong supervision (Bearman et al., 2016; McEver & Manjunath, 2020; Ahn et al., 2019), and count labels of this nature are unusual for current machine learning methods.

3.3.1 Bounding Box Labels

To address this difficulty, we further annotate a subset of the dataset with bounding box tracks to help enable current computer vision methods, which often require bounding boxes for training and testing, and to validate those methods on DUSIA, using the marine scientists' CABOF labels. First, we select a subset of species to annotate with stronger annotations. We choose ten species, each visualized in Fig. 5 because they are some of the most abundant species in the dataset. Appendix A shows the counts of all invertebrate species annotated with count labels across DUSIA.

To generate our training set, we randomly select a subset of frames containing count labels for our species of interest. We seek to those frames and back up in the video until the annotated species individual or group, i.e. our annotation target(s), is either in the top half of the screen or first appearing. In the ROV viewpoint, objects typically appear at the top of the frame as the ROV moves forward. Once we back up sufficiently far, we then draw a bounding box or boxes on the annotated target(s), ignoring other instances of species of interest (thus creating partial annotations) due to annotation budget and visibility constraints.

We then jump 10–30 frames at a time adjusting the box location for the annotation target(s) in each frame we land on, referred to as *keyframes*. This process allows for efficient annotation and allows us to interpolate box locations between keyframes for additional annotation points.

The result of this annotation process is a partially annotated training set for learning to detect and later count species of interest. These annotations are partial because we did not attempt to always label every individual of each species of interest in the training set. Instead, we focused only on the annotation targets. Because some individuals of the ten species of interest may be labelled while other individuals of the ten species may not be, we consider these partial labels.

We chose to partially annotate the our training set so that we could collect boxes tracking each species. In populated areas, there are many species hiding, coming, and going, making collecting full annotations extremely difficult, especially across many frames.

Additionally, we provide some fully annotated frames where we guarantee that all individuals of the ten species of interest in the bottom half of each frame are labelled with a bounding box. We were constrained to the bottom half of the frame due to darkness, murky waters, low visibility, and text embedded in the videos during the collection process. Therefore, we use only the fully annotated bottom half of the validation and testing frames during training, testing, and presenting our detection results. Seeing as the marine scientists count the creatures that touch the bottom of the frame, we expect the bottom half of the frame to provide a good metric for count estimations. These frames are provided for validation and testing.

In order to generate these fully annotated validation and testing frames, we randomly selected a subset of count annotated frames in the validation and test sets. For each of those selected frames, we labelled all instances of species of interest in the bottom half of the frame including but not limited to the original targets. For rare species, we often labelled frames a second or two before and/or after the count annotated frame in order to provide more validation and testing frames. Still, the number of validation and testing frames is limited by the difficulty in collecting these fully annotated frames as well as the scarcity of some species.

These fully annotated frames took on average 146.5 s per frame for trained individuals to annotate. For reference, it took annotators approximately 22.1 s per image to fully annotate with single point annotation and 34.9 s per image with squiggle supervision in the VOC2012 natural image dataset of 20 classes including cats, busses, and similar common object classes (Bearman et al., 2016). Collecting bounding boxes, consisting of two precise points, with half the number of classes should take a similar amount of time, but the difference in time spent per image illustrates the challenge of annotating DUSIA as each annotator struggled to find every object of interest even after being trained to specifically to localize the species of interest. An example of a fully labelled validation frame is shown in Fig. 6.

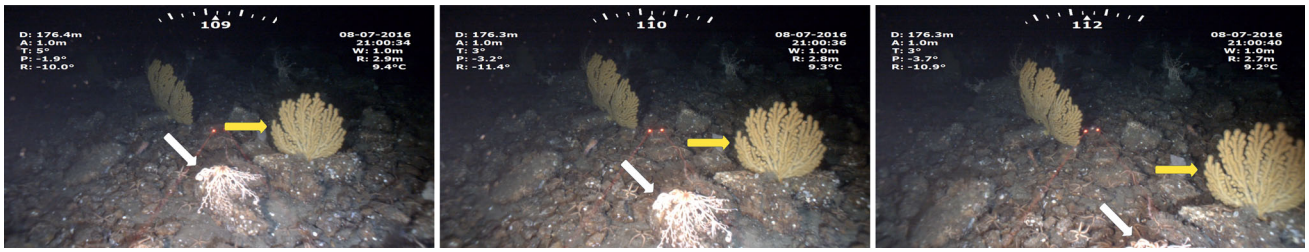

Fig. 4 Sequence of video begins on the left and continues to the right as indicated by the time stamps in the top right of the video. The basket star indicated by the white arrow will be counted when it first touches the bottom of the frame in the middle frame at time 21:00:36. The yellow gorgonian indicated by the yellow arrow will be counted when it touches the bottom of the frame later in the video

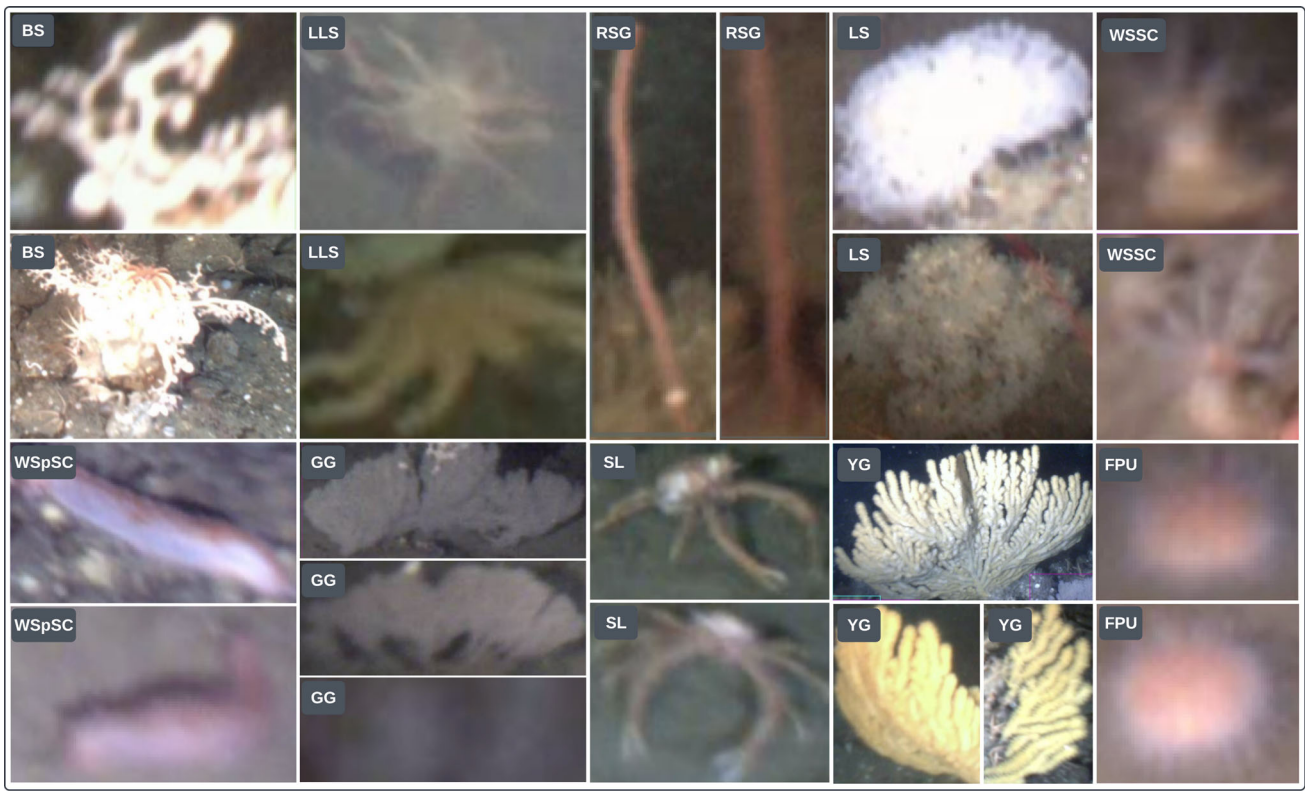

Fig. 5 Cropped screenshots of each of the ten species of interest: basket star (BS), fragile pink urchin (FPU), gray gorgonian (GG), long-legged sunflower star (LLS), red swifita gorgonian (RSG), squat lobster (SL), laced sponge (LS), white slipper sea cucumber (WSSC), white spine sea cucumber (WSpSC), and yellow gorgonian (YG)

3.4 Dataset Splits

We provide a split of the dataset into training, validation, and testing sets with 13, 3, and 6 videos in each split respectively. The training set includes 8682 keyframes used for training the detector (described in detail in Sect. 3.3). The validation and test sets respectively include 514 and 677 frames with fully annotated lower halves. Between each split, we attempted to maintain a relatively even distribution across our species of interest; however, preserving this distribution leads to a slightly uneven distribution of substrate occurrences.

3.5 Statistical Analysis of Data

Table 5 shows the frequency of each of the substrate classes present in our dataset.

Table 6 shows the frequency of bounding box labels for invertebrate species of interest represented in our dataset, and Table 7 illustrates the frequency of CABOF labels for invertebrate species.

Table 8 illustrates the distributions of CABOF labels for each species across the different substrates. While not weighted against the relative presence of each substrate, this

Fig. 6 Fully annotated frame example. Color to species map is as follows: yellow: laced sponge, magenta: white spine sea cucumber, cyan: white slipper sea cucumber, green: squat lobster

Table 5 Distribution of number of frames containing each substrate across DUSIA and its splits. Note that a given frame may have multiple labels

	Boulder	Cobble	Mud	Rock	Total
Train	70,248	247,764	259,535	183,020	760,567
Val	14,899	28,694	23,656	63,322	130,571
Test	30,742	91,695	102,422	87,399	312,258
Total	115,889	368,153	385,613	333,741	1,203,396

table still illustrates that certain species occur much more frequently in certain substrates. For example, fragile pink urchins (FPU) rarely occur in the boulder substrate, and frequently occur in mud while laced sponges (LS) almost always occur in a substrate that includes rock. These correlations suggest that learning to predict substrate may aid in learning the relationship between substrate and species and motivate a context driven approach for species detection and counting.

4 Tasks

While our dataset has a plethora of uses, we present two specific tasks for which our dataset is well suited.

4.1 Substrate Temporal Localization

The first step marine researchers take to analyzing the videos that they collect is to define the temporal spans of each substrate by indicating the start and end times of each substrate as the substrate changes while the ROV drives over the ocean floor. Many substrates may occur simultaneously, which slightly complicates the problem making it a mutli-label classification problem. Our dataset makes it possible to develop and test automated methods for this problem.

Localization Evaluation We evaluate the performance of substrate temporal localization using mean Average Precision (mAP). For each frame, we make a prediction for each substrate class with some confidence value. We use these pre-

Table 6 Distribution of bounding box annotations of each species across splits

	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	Total
Train	1247	3675	3294	735	775	3264	1071	1397	819	1024	17,301
Val	61	394	259	20	85	594	91	439	51	38	2032
Test	124	653	277	61	79	1181	98	506	28	180	3187
Total	1432	4722	3830	816	939	5039	1260	2342	898	1242	22,520

Note that one species individual may be annotated with multiple bounding boxes as it occurs across multiple frames

Table 7 Distribution of CABOF labels across DUSIA and its splits.

	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	Total
Train	292	2828	398	269	190	1649	517	832	279	103	7357
Val	17	154	80	8	19	208	40	164	22	9	721
Test	52	420	78	29	48	742	75	317	17	38	1816
Total	361	3402	556	306	257	2599	632	1313	318	150	9894

As described in Sect. 3.3, each species individual is counted only once when it touches the bottom of the frame

Table 8 Percentage of total species individuals occuring in each substrate according to CABOF labels

	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG
B	0.302	0.059	0.362	0.206	0.198	0.219	0.168	0.224	0.176	0.340
C	0.773	0.370	0.797	0.575	0.712	0.581	0.454	0.754	0.601	0.887
M	0.288	0.813	0.185	0.951	0.471	0.689	0.372	0.467	0.896	0.127
R	0.670	0.424	0.464	0.297	0.716	0.745	0.998	0.585	0.324	0.380

Note that a given frame may have multiple substrate labels, so a given individual may occur in multiple substrates at one time

dictions and ground truth to compute per class AP and take the mean of the per class AP scores to compute mAP.

4.2 Counting Species Individuals

DUSIA also makes it possible to count the number of individuals of species occurring in the videos. Counting can be achieved in three stages: detection, tracking, and then counting. We present a simple baseline method for achieving this. While many computer vision methods for counting may rely on localization information such as bounding boxes, marine researchers are interested in the number of individuals occurring in the video and are less interested in where exactly in the frame an organism occurs. They can use video timestamps of those individuals' occurrence to map those timestamps back to their GPS coordinate time log from the expedition in which the video was captured, generating population density maps for different species.

Additionally, we provide bounding box labels for ten species of interest as described in Sect. 3.3.

Detection Evaluation We use these bounding box labels to evaluate the performance of the object detection stage of counting with mean Average Precision (mAP). For each bounding box prediction, we compute its intersection over union (IOU) with each ground truth box. If a box's IOU is over a threshold, this box is counted as a true positive. Each ground truth box can correspond with only one prediction, and additional predicted boxes with high IOU with that ground truth box are counted as negative. Using this negative, positive system, we can compute the average precision (AP) for each class, and take the mean of per class APs to compute mean Average Precision (mAP). This object detection computation follows standard practice (Lin et al., 2014; Everingham et al., 2015).

Counting Evaluation To evaluate our counting performance, we simply compute relative error (RE)

$$RE = \frac{Predicted - Actual}{Actual} \tag{1}$$

using our predicted counts. Negative RE indicates that species was under-counted, and a positive sign indicates that a species was over-counted. In order to summarize the performance of our counting method, we take the mean of the absolute values of per class REs.

5 Methods

While our dataset can be used to train models to solve a wide variety of problems including substrate classification, species hotspot estimation, species counting, and invertebrate tracking, we present methods for substrate temporal localization and invertebrate species detection using partially supervised frames with our primary focus on invertebrate species detection. We feed our detection results to ByteTrack's tracking algorithm (Zhang et al., 2021) to track invertebrate species and present a simple method for using these tracks to count invertebrate individuals.

5.1 Substrate Classification

For a baseline, we train two basic classifiers for substrate classification. First, we trained an out-of-the-box ResNet-50 based (He et al., 2016) classification CNN, pre-trained on ImageNet (Deng et al., 2009), on frames pulled from training videos to predict four substrates at once. Then, we trained four separate ResNet-50 classifiers, one per substrate, and combined the prediction results from each of the classifiers by simply assigning each of their confidence predictions to each class since substrate classification allows multiple substrates to be present in a single frame.

5.2 Invertebrate Species Detection

We trained an out-of-the-box Faster RCNN model using our partially annotated keyframes (see Sect. 3.3 for partial annotation description). We chose Faster RCNN for its adaptability and ability to classify smaller boxes, with which some object detectors struggle. As shown in Fig. 10, many classes in DUSIA are made up of small boxes.

Figure 7 shows vanilla Faster RCNN in black. An image is fed to a backbone network, and image features are fed to a region proposal network. Then, region of interest pooling selects proposed regions. Finally, fully connected layers classify each region and regress the bounding box coordinates to

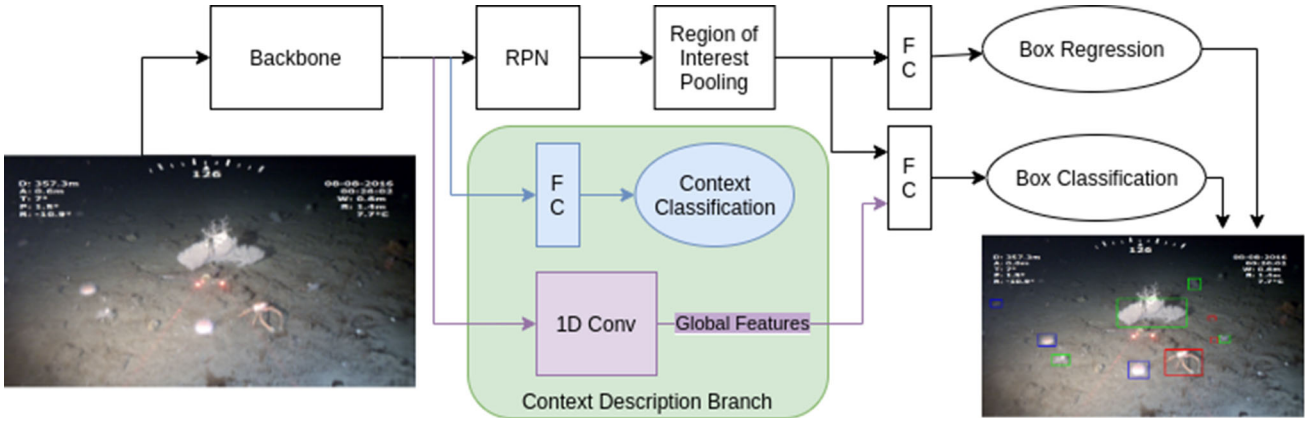

Fig. 7 Context-Driven Detector: the Context Description Branch (green) takes features from the backbone, classifies context explicitly (blue), and feeds a global representation of context (purple) to the box classification layer to enhance detections. We show that using this branch enhances the detections overall indicating that learning from explicit context labels can enhance detections (Color figure online)

refine their localization. We made no modifications to Faster R-CNN for this baseline model and refer to this version as vanilla Faster RCNN with the loss function, L_v, described by Ren et al. (2015):

$$L_v = L_d + L_p \tag{2}$$

where L_d is the loss for the detector and L_p is the loss for the region proposal network. Since we make no modifications to this part of the loss, we leave the details of the original loss description to the source paper.

5.2.1 Negative Region Dropping

Because much of our partially annotated training set contains unlabelled individuals of species of interest, we propose an approach for teaching the detection network to pay more attention to the true positive labels, and to pay less attention to potential false positives during training because a false positive may actually just be an unlabelled positive. There is generally no way of being sure whether an individual of interest is not present given a partially labelled training set, but all of the boxes provided for training are correct, true positive examples. Since humans can make sense of such a scenario, we aim to create a method for a detector to emulate that process.

Faster RCNN's region proposal network (RPN) generates proposals and computes a loss to learn which proposal contains an object of interest or not. Each proposal is assigned a label, positive or negative, based on whether it has sufficient overlap with a ground truth box (positive) or not (negative). Because DUSIA's training set contains unlabelled positives, we propose randomly dropping out a percentage of the negative proposals, thereby giving negative examples a lower weight and positive examples a higher weight. Dropping these negative proposals simply equates to not including them in the RPN's loss, L_p.

We explore different percentages, ρ, to drop in Sect. 6, and show that dropping negative proposals in this way leads to significant improvement in detection performance on DUSIA.

5.2.2 Context Driven Detection

To improve invertebrate detection using context annotations, we introduce the novel Context Description Branch as shown in green in Fig. 7. The first iteration of the context description branch (blue in Fig. 7) flattens the feature map from the backbone network and feeds this flattened vector to a fully connected layer which is trained in tandem with the detection branch to predict the multi-class substrate label. Simply backpropagating a weighted binary cross entropy loss to the backbone network to predict the substrate label increases the model's performance and generalizability (as measured by performance on the test set) by teaching the network about context via explicit context classification. This joint optimization generates cues in the backbone feature map that improve the invertebrate detection. For this iteration of the network, the loss function looks the same as Eq. (2) with the additional loss for explicit context classification.

$$L = L_v + \alpha * L_c \tag{3}$$

where α is a hyperparameter weight and L_c is a binary cross entropy loss for context labels.

By feeding global features alongside local features to the box classification layer, we can also enhance the model's performance; however, for the network to learn from them simultaneously, the global and local features must be on similar orders of magnitude. For vanilla Faster RCNN, the local

box features are vectors of size 1,024. Global features from the ResNet-50 backbone, though, are much larger. To address this size mismatch, we add a 1D convolution layer to the context description branch, which reduces the dimension of the backbone's feature map. This reduced map represents the global context information, which is largely the visible substrate, to a dimensionality on the same order of magnitude as each of the box features that are fed to the box classification head's fully-connected layer. Along those lines, we also scale the global features to match the local box feature vector by simply multiplying the global features element-wise with a scalar hyperparameter, β.

Because Faster RCNN predicts the class of each box based on a set of box features, which is a local representation of the object that is being classified, we enhance these box classifications by concatenating each image's global context information to each of its box features. This concatenation fuses together local and global features and allows the network to draw more immediate conclusions about the global information, object features, and their relationship, which is especially relevant when classifying invertebrate species in this setting. Here, we make no changes to the loss function from Eq. (3), and the 1D convolution kernel is learned.

5.3 Invertebrate Tracking and Counting

To illustrate an example pipeline for invertebrate counting, we use a detection plus tracking approach. First, we train our detector on keyframes from our training set, and then we run inference on the full validation and testing videos at 30 fps saving all detections including their spatial and temporal locations, class labels, and confidence scores.

As an intermediate step, we filter out all low confidence detections under different thresholds so that the tracker does not see low confidence detections.

ByteTrack (Zhang et al., 2021) takes as input the detections (box coordinates and confidence scores) of a single class at a time and metadata from the images (e.g. image size). In short, ByteTrack performs a modified Kalman filter based algorithm to the detections in order to link them in adjacent frames and assign each detection a track ID, or filter it out.

We apply a second filter to the output of ByteTrack such that track IDs that occur in too few frames are filtered out.

Finally, we count species individuals. To emulate the process used by marine scientists, we only count species individuals that touch the bottom of the frame. So, if a tracked species' box touches the bottom of the frame, we mark its track ID as counted and simply increment its class's count. This way, for each video, we can compute a total number of species per video that we can then compute relative error using our predicted counts and the sum of each video's CABOF labels.

6 Experiments

We test a few models and methods for the substrate temporal localization task in an effort to provide a baseline for other works to improve upon.

6.1 Substrate Temporal Localization

6.1.1 Single Classifier

We test a simple ResNet-50 based image classifier trained with a batch size of 32, learning rate of 0.1, and up to 50 epochs, selecting the epoch weights that perform best on the validation set. We also tested learning rates of 0.01 and 0.001 for our classifiers, and these models performed similarly but slightly worse. Table 9 shows the results of these experiments as predictions were made on the fully annotated frames of our validation and testing sets. These two sets are included for comparison with the context classification performance of CDD with explicit context classification, though CDD is optimized to perform detection simply using substrate prediction as a guiding sub-task. For substrate localization, though, we have annotations for almost every frame. So, we also present our classification performance on the test_1fps set, which includes many more frames from the test videos. To generate test_1fps we simply sample the entire test videos uniformly at one frame per second. We then classify each frame, and present the AP scores. test_1fps aims to illustrate the performance of the substrate classifiers across the length of the entire video rather than only on small parts of the video containing bounding box labels for species of interest.

6.1.2 Combination of Binary Classifiers

As mentioned in previous sections, substrate annotations are currently completed by trained marine scientists in multiple passes through each video, one pass per substrate. Inspired by this approach, we use one binary classifier network per substrate class. Each ResNet-50 image classification network is trained independently on the training set; however, each network is trained to simply indicate whether one substrate is present or not. We use each classifier's prediction together to predict the multi-class label and refer to this method as our combined approach. Table 9 shows that this method improves performance over a single multi-classifier for most substrates, indicating that each approach may have different use cases.

All classifiers seem to struggle with correctly identifying the boulder substrate, and, given the nuance in differences between hard substrates, this is not surprising considering the classifiers have little scale information to use to determine and differentiate exact sizes of different pieces of cobble, boulders, or larger rock formations. Additionally, the chang-

Table 9 Substrate classifier performance. Per class APs are shown for the test_1fps set, described in Sect. 6.1.1. CDD shows the classification performance of the CDD with $\alpha = 0.0001$ and $\rho = 0.75$, which was not run on test_1fps because CDD is not a dedicated substrate classifier

	val mAP	test mAP	test_1fps per class APs Boulder	Cobble	Mud	Rock	test_1fps mAP
Binary	**0.588**	**0.646**	**0.274**	**0.802**	0.750	**0.826**	0.663
Single	0.551	0.572	0.259	0.777	**0.951**	0.781	**0.692**
CDD	0.517	0.596	–	–	–	–	–

ing perspective of the ROV makes it difficult to understand scale in the videos. That said, a dedicated boulder detector out-performed the single classifier method overall due to its impressive performance classifying the mud class.

6.2 Invertebrate Species Detection

In order to detect species individuals, we present mean average precision (mAP) results for object detection with an intersection over union (IOU) threshold of 0.5 because our counting task is not particularly sensitive to high overlap. This metric is known as AP_{50} from the popular COCO evaluation metric suite (Zhao et al., 2019). As long as the object is detected reliably, the quality of the localization is not as important as coming up with the correct counts of species individuals. We present the full COCO suite of evaluations for a more in depth analysis of our best CDD model in Table 16.

We offer a comparison of single-stage, transformer-based, and two-stage out-of-the-box detection models on DUSIA in Table 10. YOLOv5l (Jocher et al., 2022) is the large model of the single stage object and performs best of all default YOLOv5 model sizes. The DEtection TRansformer (DETR) is a recent object detection model that uses a transformer-based object detector to make object detections.

For each Faster-RCNN and CDD detection experiment, we initialize our models with weights pretrained on ImageNet and then train the network for up to 15 epochs. All detection networks (including YOLOv5 and DETR) only ever see the bottom half of any given video frame. That is, the top half is cropped out, and the models are trained on the bottom half. Section 3.3.1 describes more on the reasoning for avoiding the top half of DUSIA's video frames for object detection.

We select the model from the epoch with the best performance on the fully annotated frames of the validation set. Then, we run inference on the fully annotated frames of the test set using those selected model weights. We repeat the training and testing procedure four times for each experiment and report the average results over the four runs because PyTorch does not support deterministic training for our model at the time of writing.

We first train vanilla Faster RCNN (Ren et al., 2015) with a batch size of 8 and try several learning rates after initializing with weights pre-trained on COCO (Lin et al., 2014) provided by PyTorch (Paszke et al., 2019). The results are shown in Table 11.

Table 10 Detection models tested, the approximate number of parameters of each model, and their performance on DUSIA. YOLOv5l and DETR were trained and tested with default parameters

Models	# params	val mAP	test mAP
YOLOv5l	46 M	0.485	0.363
DETR	41 M	0.499	0.387
Faster RCNN	42 M	0.490	0.391
CDD	41 M	0.524	0.447

Table 11 Performance of vanilla Faster RCNN with varying learning rates

lr	val mAP	test mAP
0.1	0.454	0.361
0.01	**0.490**	**0.391**
0.001	0.482	0.367

We then perform hyperparameter searches for each of our method contributions described in Sect. 5: α for explicit context learning and backbone refinement, β for global context feature fusion, and ρ for Negative Region Dropping. After testing each hyperparameter independently, we try combinations of each and discuss the results. We prioritize test mAP over val mAP as test mAP is more indicative of the generalizability of our model since the best model weights are selected on best val mAP.

6.2.1 Negative Region Dropping Percent ρ

Table 12 shows that Negative Region Dropping consistently improves the training on DUSIA by teaching the network to focus more on learning from true examples than negative examples. Interestingly, setting ρ to 1.0 detrimentally harms performance indicating that having some negative regions contribute to the region proposal loss is still important.

6.2.2 Global Feature Fusion Scalar β

By creating a global feature representation and feeding it later in the network, the network is better able to classify boxes

Table 12 Performance of Faster-RCNN with varying Negative Region Dropping percentages

lr	ρ	val mAP	test mAP
0.01	0	0.490	0.391
0.01	0.5	0.492	0.413
0.01	0.75	**0.509**	**0.439**
0.01	0.9	0.492	0.403
0.01	1	0.297	0.264
0.001	0.75	0.479	0.380
0.001	0.9	0.481	0.380

Table 13 Performance of the Context Driven Detector given different β scalar values

lr	β	val mAP	test mAP
0.01	0	0.490	0.391
0.01	0.1	0.471	0.371
0.01	0.01	0.491	0.397
0.01	0.001	**0.499**	0.396
0.01	0.0001	0.494	**0.410**
0.01	1.0E-05	0.496	0.406
0.01	1.0E-06	0.482	0.394
0.001	0.01	0.475	0.374
0.001	0.001	0.477	0.371

Table 14 Performance of the Context Driven Detector given different context loss scaling α values

lr	α	val mAP	test mAP
0.01	0	0.490	0.391
0.01	0.1	0.470	0.389
0.01	0.01	0.494	0.419
0.01	0.001	0.487	0.401
0.01	0.0001	0.502	**0.420**
0.01	1.0E-05	**0.507**	0.410
0.01	1.0E-06	0.501	0.408
0.001	0.01	0.456	0.358
0.001	0.001	0.453	0.361

Table 15 Average performance of best models from each hyperparameter combination

α	β	ρ	val mAP	test mAP
0	0	0	0.490	0.391
0	0.0001	0	0.494	0.410
0.01	0.1	0	0.480	0.420
0.0001	0	0	0.502	0.420
1.0E-06	0.01	0.75	0.517	0.430
0	0	0.75	0.509	0.439
0.0001	0	0.75	0.514	0.439
0	0.01	0.75	**0.524**	**0.447**

correctly, but concatenating a global feature representation with the local box features requires that the features come in at similar scales. As described in Sect. 5.2.2, β is used as an element-wise multiplicative scalar to re-scale of the global features. Table 13 shows the effect of different scalar values for this fusion.

6.2.3 Context Loss Weight α

By modifying the detector to simultaneously classify the context of an image in parallel with detection, we demonstrate that simply backpropagating information useful for classifying substrate to the backbone also serves to help improve detection performance. Training a joint task in this way leads to less powerful context classifications than a dedicated context classifier, but it leads to a more powerful object detector. Table 14 shows the effects of α on the detection performance.

6.2.4 Hyperparameter Combinations

We illustrate that each hyperparameter alone can improve the detector performance over the baseline out-of-the-box models. We further illustrate that Negative Region Dropping and context driven detection can work in tandem to further improve performance. We also find that a context driven detector with both implicit attention to context (global feature fusion) and explicit context classification does not necessarily outperform implicit context usage or explicit classification only. Training on both implicit and explicit context simultaneously may interfere with each other. Still, we emphasize that learning from context can significantly improve object detection performance in this setting, and we aim to find even better ways to utilize contextual information to better classify objects in future work.

Table 15 highlights the best hyperparameter settings revealed during our search, and Appendix B goes into more detail on the settings tested for this study. Note that the β column set to zero indicates that global features are not being scaled by 0, rather they are not being concatenated with the local box features at all.

Table 16 shows the performance of the best CDD model over the whole COCO evaluation suite, which is commonly used to evaluate the performance of object detectors (Zhao et al., 2019). This suite shows the performance of object detectors over a range of IOU thresholds and for different box sizes: small, medium, and large. AP_{50} is the metric shown in other object detection performance tables as our tasks are not particularly sensitive to high IOU detections. The metrics show that our detector struggles to find small objects.

Table 16 Full COCO suite showing performance of the best single CDD model with β=0.01 and ρ=0.75 on both the val and test sets

metric	val	test
$AP_{50:95}$	0.226	0.201
AP_{50}	0.517	0.449
AP_{75}	0.150	0.155
AP_S	0.011	0.010
AP_M	0.154	0.144
AP_L	0.310	0.302

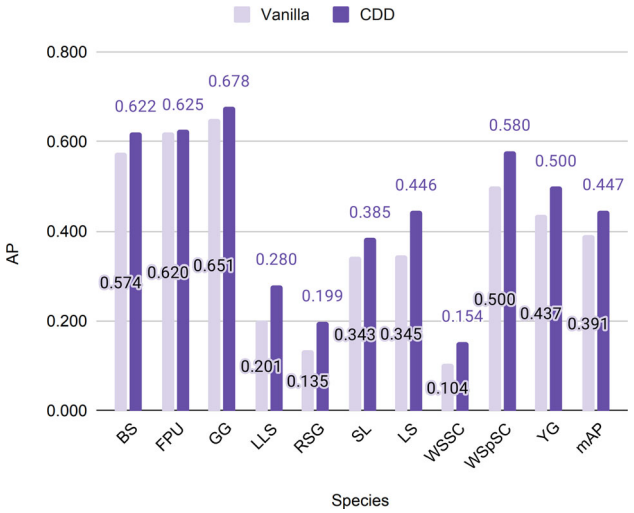

Fig. 8 Per class test AP comparison of vanilla Faster RCNN and the best Context Driven Detector

We find that Negative Region Dropping increases the overall performance of both vanilla Faster RCNN and context driven detectors. While explicit and implicit context usage may conflict with one another in training, independently they can achieve performance increases. The best model overall is achieved with global context feature fusion and Negative Region Dropping, and a model with explicit context classification and Negative Region Dropping follows close behind. We find that using context to influence detections leads to a 7.4% increase, using negative region dropping leads to a 12.3%, and together they can achieve a 14.3% increase in mAP on the fully annotated frames in DUSIA's test set.

Figure 8 illustrates the per class AP detection performance of our best model compared with vanilla Faster RCNN showing that our model significantly increases performance on all classes. Figure 9 shows qualitative examples of success and failure cases of the best version of CDD.

6.3 Invertebrate Species Counting

There are some noteworthy differences between the detection and counting problems. As mentioned in Sect. 3.4, we partition DUSIA's videos into three sets: training, validation, and testing sets. However, the detector sees only a small fraction of each video as only a small subset of each video has bounding box annotations. Further, while we refer to three of our videos as validation videos, our detection models do not train on those videos at all, and only 514 frames from those 124,000 validation video frames are used in the detection validation process to select our best model weights.

In contrast, our counting method runs our detector on the entire lengths of the videos in the validation and testing sets, posing a great challenge to the generalizability and robustness of an object detection model. That is, the sets of frames used for the counting task are much larger than those used for detection. Also, the frames annotated with invertebrate species (i.e. all the frames in the detector's training set) all include instances of those species of interest. In contrast, each video contains long time spans of both densely and sparsely annotated areas including some long regions with no species of interest. As a result, counting species individuals poses a very challenging problem, and much work remains to be done in the power of a detector and its ability to differentiate between background and species of interest in both sparsely and densely populated environments.

Still, we aim to demonstrate the challenge of this problem with a simple baseline method, though much work remains to be done to achieve a result that would be able to replace the annotation abilities of trained marine scientists. We hope that DUSIA can aid in pushing the limits of computer vision models and extend computer vision methods' usefulness into more challenging, scientific data.

In order to count invertebrate individuals, we first run the best performing version of CDD on each of our val and test videos at the full frame rate of 30 fps and save all detections. Then, we filter out all detections with confidence scores under a threshold, τ, before feeding all detections to ByteTracker. We then filter the output of ByteTrack by discarding any track IDs with less than γ detections in the track. That is, if a track ID is assigned to boxes in only a few frames, we discard that track ID. We experimented with ByteTracker's hyperparameters and found that their effect was significantly smaller than the effects of τ and γ, so we opt to use the default hyperparameter settings for ByteTracker. We leave the details of ByteTracker to the original work (Zhang et al., 2021). Finally, for each species, we count the number of that species' track IDs that touch the bottom of any frame.

We applied the two aforementioned filters because, without any filters, our method vastly over counts all species through all videos. Figure 9 shows examples of a few false positive detections, and these types of errors likely contribute heavily to our method's over counting as the detector is run over hours of videos, accumulating false positive results.

To address the over counting issue, we opted to feed the tracker only our most confident detections and to only count tracks that occur across multiple frames. This filtering significantly improved the performance, but the error remains unacceptably high.

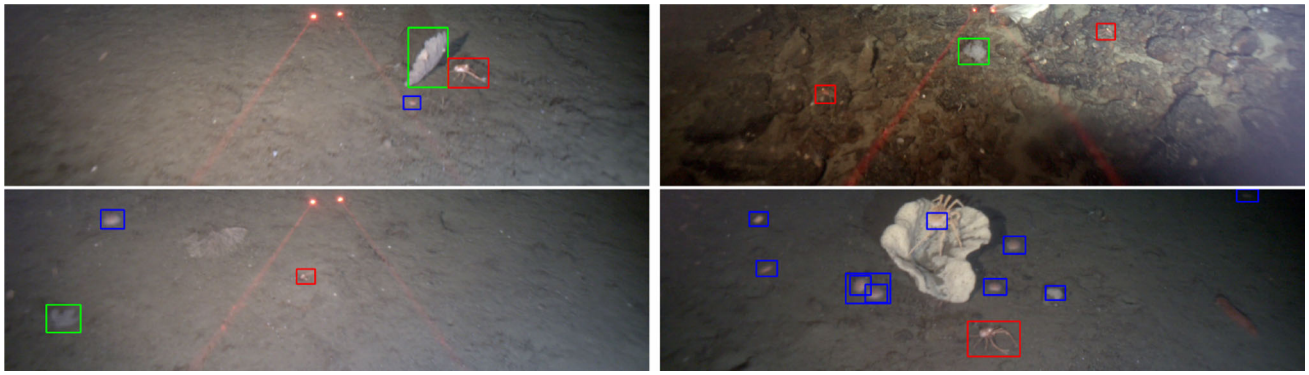

Fig. 9 Detection examples from our dataset. Blue indicates fragile pink urchin; green, gray gorgonian; and red, squat lobster. We show the success of our detector with the exception of the bottom right image. A crab (not a species of interest) is mislabeled as a fragile pink urchin toward the top center of the image. In the left side of the image, two pieces of floating debris are labelled as urchins, and close to the center two urchins are counted thrice. Right of center, a rock is labelled as an urchin. These failure cases demonstrate some of the challenges of DUSIA. In the top right corner of the bottom right image, a very difficult to see pink urchin is correctly detected (Color figure online)

Table 17 Relative errors of our counting method with no thresholding and the best threshold settings. Darker color indicates better performance. See Table 7 for ground truth counts for each species

val set per species relative errors												
γ	τ	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	mean
0	0	11.2	4.04	5.75	25.6	60.9	3.18	0.35	2.98	2.32	18.7	13.5
20	0.5	−0.18	−0.091	−0.34	1.13	−0.11	−0.50	−0.90	−0.88	−0.27	0.00	0.439
test set per species relative errors												
γ	τ	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	mean
0	0	6.00	4.73	15.38	46.66	70.23	3.29	2.57	2.84	3.71	12.21	16.8
20	0.5	−0.56	0.14	−0.03	1.28	−0.25	−0.51	−0.84	−0.91	−0.24	−0.39	0.515

Table 17 shows the relative error for each class on the val and test videos as well as the mean relative error, averaged over all classes, as we vary the τ and γ parameters. We leave the error sign to indicate over (positive error) or under (negative error) counting, but we compute the mean errors using the absolute value of the error values for each class. Clearly, the detector hardly learns some of the rarer classes (e.g. long-legged sunflower star and red swiftia gorgonian) and regularly misclassifies background, which may include species outside of our ten species of interest, as our species of interest. Appendix B contains more experiment error results for varying these filter thresholds.

Ultimately, these baseline results indicate that this simple method is not powerful enough to put into practice given the effectiveness of our current detection model. Much work on methods for this problem is left to be done. We could look deeper into per class thresholds, but we expect improving object detections, false positive filtering, and the tracking algorithm would be more robust. We leave these improvements to future work.

7 Discussion and Future Work

Our baseline methods' detection and counting performance leaves plenty of room for improvement as a counting system 51.5% average error cannot replace human annotators. Our detection methods can improve because they do not enforce any sort of temporal continuity present in the ROV videos, which could likely improve performance, and the methods do not yet take advantage of the abundant, weak CABOF labels during training. Further, Table 16 reveals that our detector struggles to find small objects. This weakness may be an area to improve in future work.

It is interesting to find the difference in performance of the different types of substrate classifiers. Overall, the substrate classification results are good enough for some substrates, and in future work we hope to see results good enough to fully automate this process. Additionally, marine scientists are interested in real time substrate classifiers that can indicate which substrates the ROV is passing in real time. Any indication of species hotspots in real time during expeditions

can improve each excursion's productivity by reducing more manual means of searching for given substrates, habitats, and species hotspots.

The detection results of the Context Driven Detector provide a baseline, but in order to fully translate these detections to tracks with individual re-identification and counting, there is much work to be done. We hope to next take full advantage of the CABOF labels and to use context in more powerful ways to improve detection performance in future work. Further, we plan to enforce temporal continuity to improve our counting predictions. These improvements can lead us to eventually begin automating some of the invertebrate counting that is currently done manually.

By making DUSIA public, we also invite other collaborators to work independently or in cooperation with us to help improve our methods.

Supplementary Information

DUSIA's data, annotations, and baseline methods will be made publicly available at the time of publication.

Acknowledgements This research was supported in part by National Science Foundation (NSF) award: SSI # 1664172. We would like to thank Dirk Rosen and Andy Lauermann from Marine Applied Research & Exploration group for their video collection, guidance, and help through this project. We would also like to thank Dr. Robert Miller, Anmol Kapoor, and Shafin Haque for their contributions to the project.

Funding This work was partially supported by National Science Foundation (NSF) award: SSI # 1664172

Availability of Data and Materials Data and annotations will be made publicly available at the time of publication.

Code Availability Code and implementation of methods will be made publicly available at the time of publication.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval N/A

Consent to Participate N/A

Consent for Publication N/A

Appendix A: Species Statistics

See Table 18 and Fig. 10.

Table 18 All species and their counts in DUSIA. Bold shows the species that also include bounding box annotations. UI stands for unidentified and is used when organism's exact species cannot be determined

Species	Count	Species	Count
Fragile pink urchin	**3402**	Spot prawn	18
Squat lobster	**2593**	UI anemone	17
UI lobed sponge	1753	Thorny sea star	16
White slipper sea cucumber	**1313**	UI anemone 2	14
Laced sponge	**632**	California king crab	13
Gray gorgonian	**556**	UI trumpet sponge	12
UI hairy boot sponge	426	Pom-pom anemone	11
Basket star	**361**	UI prawn	9
White spine sea cucumber	**318**	Crested sea star	8
Long legged sunflower star	**306**	White sea pen	8
Red swiftia gorgonian	**257**	Red sea star	8
UI branched sponge	228	UI sea pen	7
UI vase sponge	210	Solaster sun star complex	6
Yellow gorgonian	**150**	UI octopus	5
UI boot sponge	128	UI nipple sponge	4
Cookie star	90	UI gorgonian	3
UI anemone 4	67	Spiny/thorny star complex	3
UI sea star	54	Gray moon sponge	2
UI tubeworm	50	Brown box crab	2
Henricia complex	47	Decorator crab	2
UI large yellow sponge	44	UI sand dwelling anemone	2
UI thin red star	39	UI nudibranch	1
UI orange gorgonian	38	Orange puffball sponge	1
Mushroom soft coral	36	Red octopus	1
Black coral	34	Red gorgonian	1
Benthic siphonophore	25	Rose star	1
Bubblegum coral	24	Cushion star	1
Deep sea cucumber	20	UI urchin	1
Fish eating star	19	UI anemone 1	1
Spiny red star	18		

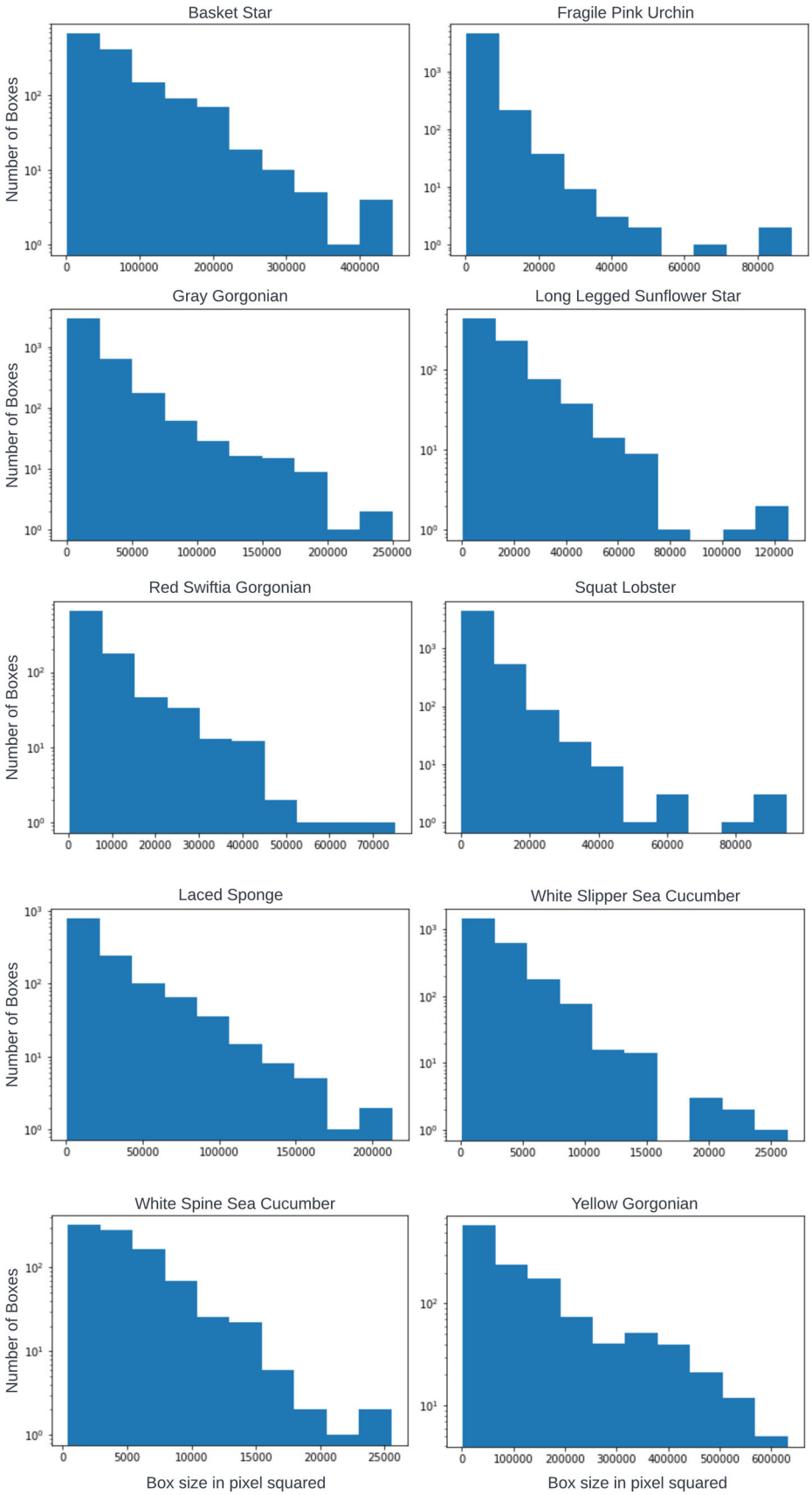

Fig. 10 Histograms illustrating the distributions of box sizes (in pixels squared) for each species of interest

Appendix B: Hyperparameter Search Summary

See Tables 19, 20, 21.

Table 19 Results of hyperparameter search experiments on learning rate, α, β, and ρ

lr	α	β	ρ	val mAP	test mAP	lr	α	β	ρ	val mAP	test mAP
0.1	0	0	0	0.454	0.361	0.01	1.00E-04	0.01	0	0.487	0.405
0.01	0	0	0	0.490	0.391	0.01	1.00E-05	0.01	0	0.489	0.404
0.001	0	0	0	0.482	0.367	0.01	1.00E-06	1.00E-02	0	0.486	0.404
0.01	0.1	0	0	0.470	0.389	0.01	1.00E-04	1.00E-04	0	0.471	0.395
0.01	0.01	0	0	0.494	0.419	0.01	1.00E-05	1.00E-04	0	0.487	0.383
0.01	1.00E-03	0	0	0.487	0.401	0.01	1.00E-04	0.001	0	0.491	0.388
0.01	1.00E-04	0	0	0.502	0.420	0.001	0.01	0.001	0	0.469	0.373
0.01	1.00E-05	0	0	0.507	0.410	0.001	0.001	0.001	0	0.477	0.377
0.01	1.00E-06	0	0	0.501	0.408	0.01	0.01	0	0.75	0.491	0.405
0.001	0.01	0	0	0.456	0.358	0.01	0.001	0	0.75	0.487	0.406
0.001	0.001	0	0	0.453	0.361	0.01	1.00E-04	0	0.75	0.514	0.433
0.01	0	0.1	0	0.471	0.371	0.01	1.00E-05	0	0.75	0.503	0.417
0.01	0	0.01	0	0.491	0.397	0.01	1.00E-06	0	0.75	0.503	0.433
0.01	0	1.00E-03	0	0.499	0.396	0.01	0.1	0	0.9	0.500	0.431
0.01	0	1.00E-04	0	0.494	0.410	0.01	0.01	0	0.9	0.504	0.421
0.01	0	1.00E-05	0	0.482	0.395	0.01	0	0.1	0.75	0.512	0.435
0.01	0	1.00E-06	0	0.482	0.394	0.01	0	0.01	0.75	**0.524**	**0.447**
0.001	0	0.01	0	0.475	0.374	0.01	0	1.00E-03	0.75	0.513	0.426
0.001	0	0.001	0	0.477	0.371	0.01	0	1.00E-04	0.75	0.506	0.420
0.1	0	0	0.75	0.456	0.354	0.01	0	1.00E-05	0.75	0.506	0.436
0.01	0	0	0.25	0.485	0.392	0.01	0	0.01	0.9	0.497	0.402
0.01	0	0	0.5	0.492	0.413	0.01	0	0.001	0.9	0.512	0.430
0.01	0	0	0.75	0.509	0.439	0.01	0.01	1.00E-02	0.75	0.503	0.412
0.01	0	0	1	0.297	0.264	0.01	0.01	1.00E-01	0.75	0.502	0.414
0.01	0	0	0.9	0.492	0.403	0.01	0.1	0.01	0.75	0.515	0.427
0.001	0	0	0.75	0.479	0.380	0.01	0.1	0.1	0.75	0.513	0.437
0.001	0	0	0.9	0.481	0.380	0.01	0.01	0.001	0.75	0.516	0.428
0.1	0.1	0.1	0	0.451	0.372	0.01	0.1	0.001	0.75	0.510	0.419
0.1	0.01	0.1	0	0.462	0.371	0.01	0.01	0.01	0.9	0.508	0.420
0.1	0.01	0.01	0	0.454	0.375	0.01	0.01	0.001	0.9	0.497	0.418
0.01	0.1	0.1	0	0.450	0.370	0.01	0.1	0.001	0.9	0.503	0.417
0.01	0.01	0.1	0	0.480	0.420	0.01	0.001	0.01	0.75	0.509	0.427
0.01	0.1	0.01	0	0.497	0.399	0.01	1.00E-04	0.01	0.75	0.510	0.425
0.01	0.01	0.01	0	0.489	0.403	0.01	1.00E-04	1.00E-04	0.75	0.515	0.433
0.01	0.01	0.001	0	0.488	0.408	0.01	1.00E-05	0.01	0.75	0.509	0.428
0.01	0.001	0.01	0	0.486	0.396	0.01	1.00E-06	0.01	0.75	0.517	0.430
0.01	0.001	0.001	0	0.492	0.396						

Table 20 Relative errors of our counting method with different settings across the validation set's videos. γ represents the threshold for number of frames per track ID to count track. τ represents detection confidence score threshold. Darker color indicates better performance. Note that we include the sign for per species errors to indicate over (postive) or under (negative) counting, but the absolute values of relative error are used in the mean computation

val set per species errors												
γ	τ	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	mean
0	0	11.24	4.04	5.75	25.63	60.89	3.18	0.35	2.98	2.32	18.7	13.5
10	0	1.65	0.05	0.01	2.50	3.00	−0.26	−0.70	−0.74	−0.14	1.89	1.09
15	0	0.76	−0.06	−0.14	1.25	0.68	−0.41	−0.80	−0.80	−0.27	1.22	0.641
18	0	0.53	−0.09	−0.17	1.00	0.37	−0.45	−0.82	−0.84	−0.32	1.00	0.560
20	0	0.53	−0.09	−0.20	1.00	0.11	−0.47	−0.85	−0.86	−0.32	0.89	0.531
22	0	0.41	−0.10	−0.25	1.00	−0.05	−0.51	−0.85	−0.87	−0.36	0.78	0.519
25	0	0.24	−0.12	−0.26	1.00	−0.16	−0.54	−0.85	−0.91	−0.41	0.33	0.482
27	0	0.00	−0.13	−0.29	1.00	−0.26	−0.55	−0.85	−0.92	−0.41	0.00	0.441
30	0	−0.12	−0.14	−0.36	0.88	−0.42	−0.55	−0.85	−0.93	−0.41	−0.22	0.488
0	0.5	7.24	3.95	4.59	25.75	58.42	2.74	−0.35	2.80	2.14	15.2	12.3
10	0.5	0.12	−0.03	−0.24	2.13	0.63	−0.40	−0.85	−0.83	−0.14	0.78	0.613
15	0.5	−0.12	−0.06	−0.30	1.25	0.16	−0.49	−0.87	−0.87	−0.23	0.22	0.457
18	0.5	−0.18	−0.08	−0.32	1.13	−0.05	−0.50	−0.87	−0.88	−0.27	0.11	0.440
20	0.5	−0.18	−0.09	−0.34	1.13	−0.11	−0.50	−0.90	−0.88	−0.27	0.00	0.439
22	0.5	−0.24	−0.10	−0.35	1.13	−0.11	−0.52	−0.90	−0.89	−0.32	−0.11	0.466
25	0.5	−0.29	−0.11	−0.37	1.13	−0.26	−0.54	−0.90	−0.90	−0.36	−0.22	0.510
27	0.5	−0.41	−0.12	−0.37	1.13	−0.32	−0.55	−0.90	−0.91	−0.36	−0.33	0.540
30	0.5	−0.47	−0.13	−0.42	0.88	−0.47	−0.55	−0.90	−0.91	−0.36	−0.33	0.544
0	0.9	0.18	1.23	0.23	8.00	9.26	0.42	−0.90	−0.37	0.45	1.67	2.27
10	0.9	−0.41	−0.05	−0.32	1.63	0.21	−0.49	−0.90	−0.87	−0.32	−0.44	0.564
15	0.9	−0.47	−0.08	−0.35	1.38	−0.11	−0.52	−0.90	−0.91	−0.36	−0.56	0.563
18	0.9	−0.53	−0.10	−0.39	1.25	−0.26	−0.53	−0.90	−0.91	−0.36	−0.56	0.579
20	0.9	−0.59	−0.10	−0.39	1.25	−0.37	−0.54	−0.90	−0.91	−0.36	−0.56	0.597
22	0.9	−0.59	−0.10	−0.42	1.13	−0.42	−0.55	−0.92	−0.91	−0.36	−0.56	0.596
25	0.9	−0.59	−0.11	−0.45	1.13	−0.58	−0.56	−0.92	−0.92	−0.41	−0.56	0.623
27	0.9	−0.59	−0.12	−0.47	1.13	−0.58	−0.56	−0.92	−0.93	−0.41	−0.56	0.627
30	0.9	−0.65	−0.13	−0.49	0.88	−0.68	−0.58	−0.92	−0.93	−0.45	−0.56	0.627

Table 21 Relative errors of our counting method with different settings across the test set's videos

test set per species errors												test	val
γ	τ	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	mean	mean
0	0	6.00	4.73	15.38	46.66	70.23	3.29	2.57	2.84	3.71	12.21	16.8	13.5
10	0	0.19	0.33	1.22	3.62	2.02	−0.32	−0.45	−0.77	0.00	1.08	1.00	1.09
15	0	−0.15	0.20	0.44	2.03	0.17	−0.44	−0.64	−0.85	−0.06	0.37	0.535	0.641
18	0	−0.25	0.17	0.31	1.52	−0.21	−0.48	−0.71	−0.89	−0.18	0.03	0.473	0.560
20	0	−0.35	0.16	0.23	1.34	−0.35	−0.51	−0.77	−0.91	−0.24	−0.13	0.499	0.531
22	0	−0.38	0.14	0.13	1.07	−0.44	−0.53	−0.80	−0.91	−0.24	−0.18	0.482	0.519
25	0	−0.44	0.10	−0.01	0.93	−0.56	−0.54	−0.80	−0.92	−0.24	−0.26	0.481	0.482
27	0	−0.46	0.09	−0.06	0.79	−0.56	−0.55	−0.83	−0.93	−0.24	−0.29	0.480	0.441
30	0	−0.60	0.07	−0.10	0.62	−0.67	−0.57	−0.84	−0.93	−0.35	−0.34	0.509	0.488
0	0.5	4.90	4.24	11.88	44.66	66.31	2.82	1.15	2.59	3.82	9.26	15.2	12.3
10	0.5	−0.44	0.24	0.36	2.79	0.65	−0.42	−0.71	−0.82	−0.18	0.03	0.663	0.613
15	0.5	−0.52	0.17	0.06	1.79	0.00	−0.47	−0.80	−0.85	−0.18	−0.29	0.514	0.457
18	0.5	−0.54	0.15	−0.01	1.34	−0.12	−0.50	−0.84	−0.89	−0.24	−0.37	**0.500**	0.440
20	0.5	−0.56	0.14	−0.03	1.28	−0.25	−0.51	−0.84	−0.91	−0.24	−0.39	0.515	0.439
22	0.5	−0.56	0.13	−0.08	1.03	−0.29	−0.53	−0.84	−0.91	−0.24	−0.39	0.501	0.466
25	0.5	−0.58	0.11	−0.13	0.93	−0.42	−0.54	−0.84	−0.92	−0.24	−0.42	0.512	0.510
27	0.5	−0.60	0.10	−0.15	0.86	−0.46	−0.56	−0.84	−0.93	−0.24	−0.45	0.518	0.540
30	0.5	−0.62	0.08	−0.18	0.66	−0.52	−0.57	−0.85	−0.94	−0.35	−0.45	0.521	0.544
0	0.9	−0.44	1.51	0.79	13.34	11.00	0.48	−0.79	−0.36	0.82	0.71	3.03	2.27
10	0.9	−0.71	0.17	−0.17	1.90	−0.02	−0.49	−0.88	−0.89	−0.41	−0.53	0.616	0.564
15	0.9	−0.73	0.12	−0.24	1.07	−0.42	−0.55	−0.88	−0.91	−0.41	−0.63	0.596	0.563
18	0.9	−0.75	0.10	−0.28	0.76	−0.54	−0.57	−0.88	−0.93	−0.47	−0.71	0.599	0.579
20	0.9	−0.75	0.09	−0.31	0.66	−0.56	−0.58	−0.88	−0.94	−0.47	−0.71	0.595	0.597
22	0.9	−0.75	0.08	−0.31	0.55	−0.58	−0.59	−0.89	−0.95	−0.47	−0.74	0.591	0.596
25	0.9	−0.75	0.06	−0.35	0.41	−0.67	−0.60	−0.92	−0.95	−0.47	−0.76	0.594	0.623
27	0.9	−0.75	0.05	−0.35	0.41	−0.71	−0.60	−0.93	−0.96	−0.47	−0.79	0.602	0.627
30	0.9	−0.75	0.03	−0.36	0.21	−0.75	−0.61	−0.93	−0.96	−0.53	−0.82	0.595	0.627

References

Ahn, J., Cho, S., & Kwak, S. (2019). Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2209–2218).

Anantharajah, K., Ge, Z., McCool, C., Denman, S., Fookes, C., Corke, P., . . . Sridharan, S. (2014). Local inter-session variability modelling for object classification. In *IEEE winter conference on applications of computer vision* (pp. 309–316).

Barrett, N., Meyer, L., Hill, N., & Walsh, P. (2011). Methods for the processing and scoring of AUV digital imagery from South Eastern Tasmania.

Bearman, A., Russakovsky, O., Ferrari, V., & Fei-Fei, L. (2016). What's the point: Semantic segmentation with point supervision. In *European conference on computer vision* (pp. 549–565).

Beery, S., Wu, G., Rathod, V., Votel, R., & Huang, J. (2020). Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13075–13085).

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., & Kriegman, D. (2016). Improving automated annotation of benthic survey images using wide-band fluorescence. *Scientific Reports, 6*(1), 1–11.

Bett, B. J., & Ruhl, H. (2015). Time-lapse images of the porcupine abyssal plain sustained observatory seafloor (4850 m water depth), May 2012 to April 2013. British Oceanographic Data Centre, Natural Environment Research Council. Retrieved from https://www.bodc.ac.uk/data/publisheddatalibrary/catalogue/10.5285/21e9ef8a-7562-4b9e-e053-6c86abc0ccb8/. https://doi.org/10.5285/21E9EF8A-7562-4B9E-E053-6C86ABC0CCB8

Bewley, M., Friedman, A., Ferrari, R., Hill, N., Hovey, R., Barrett, N., et al. (2015). Australian seafloor survey data, with images and expert annotations. *Scientific Data, 2*(1), 1–13.

Boom, B. J., He, J., Palazzo, S., Huang, P. X., Beyan, C., Chou, H.-M., & Fisher, R. B. (2014). A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecological Informatics, 23,* 83–97.

Danovaro, R., Fanelli, E., Aguzzi, J., Billett, D., Carugati, L., Corinaldesi, C., et al. (2020). Ecological variables for developing a global deep-ocean monitoring and conservation strategy. *Nature Ecology & Evolution, 4*(2), 181–192.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).

Ditria, E. M., Lopez-Marcano, S., Sievers, M., Jinks, E. L., Brown, C. J., & Connolly, R. M. (2020). Automating the analysis of fish abundance using object detection: Optimizing animal ecology with deep learning. *Frontiers in Marine Science, 7,* 429.

Drap, P., Seinturier, J., Hijazi, B., Merad, D., Boi, J.-M., Chemisky, B., & Long, L. (2015). The ROV 3D Project: Deep-sea underwater survey using photogrammetry: Applications for underwater archaeology. *Journal on Computing and Cultural Heritage (JOCCH), 8*(4), 1–24.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision, 111*(1), 98–136.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Horvath, E. A. (2019). A review of gorgonian coral species (Cnidaria, Octocorallia, Alcyonacea) held in the Santa Barbara Museum of Natural History research collection: Focus on species from Scleraxonia, Holaxonia, Calcaxonia—Part III: Suborder Holaxonia continued, and suborder Calcaxonia. *ZooKeys, 860,* 183.

Ishiwaka, Y., Zeng, X. S., Eastman, M. L., Kakazu, S., Gross, S., Mizutani, R., & Nakada, M. (2021). Foids: Bio-inspired fish simulation for generating synthetic datasets. *ACM Transactions on Graphics (TOG), 40*(6), 1–15.

Jäger, J., Simon, M., Denzler, J., Wolff, V., Fricke-Neuderth, K., & Kruschel, C. (2015). Croatian fish dataset: Fine-grained classification of fish species in their natural habitat. *Swansea: Bmvc*, *2*.

Jamstec e-library of deep-sea images. (2016). Retrieved from 2022 September, 27 https://www.godac.jamstec.go.jp/jedi/e/

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., . . . xylieong (2022, August). ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations. Zenodo. Retrieved from https://doi.org/10.5281/zenodo.7002879

Joly, A., Goëau, H., Glotin, H., Spampinato, C., Bonnet, P., Vellinga, W.-P., . . . Müller, H. (2014). Lifeclef 2014: Multimedia life species identification challenges. In *International conference of the cross-language evaluation forum for European languages* (pp. 229–249).

Katija, K., Orenstein, E., Schlining, B., Lundsten, L., Barnard, K., Sainz, G., et al. (2022). Fathomnet: A global image database for enabling artificial intelligence in the ocean. *Scientific Reports, 12*(1), 1–14.

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).

Konovalov, D. A., Saleh, A., Bradley, M., Sankupellay, M., Marini, S., & Sheaves, M. (2019). Underwater fish detection with weak multidomain supervision. In *2019 international joint conference on neural networks (ijcnn)* (pp. 1–8).

Langenkämper, D., Van Kevelaer, R., Purser, A., & Nattkemper, T. W. (2020). Gear-induced concept drift in marine images and its effect on deep learning classification. *Frontiers in Marine Science, 7,* 506.

Levy, D., Belfer, Y., Osherov, E., Bigal, E., Scheinin, A. P., Nativ, H., . . . Treibitz, T. (2018). Automated analysis of marine video with limited data. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1385–1393).

Li, X., Shang, M., Qin, H., & Chen, L. (2015). Fast accurate fish detection and recognition of underwater images with fast r-cnn. In *Oceans 2015-MTS/IEEE Washington* (pp. 1–5).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740–755).

Måløy, H., Aamodt, A., & Misimi, E. (2019). A spatio-temporal recurrent network for salmon feeding action recognition from underwater videos in aquaculture. *Computers and Electronics in Agriculture, 167,* 105087.

Marini, S., Fanelli, E., Sbragaglia, V., Azzurro, E., Fernandez, J. D. R., & Aguzzi, J. (2018). Tracking fish abundance by underwater image recognition. *Scientific Reports, 8*(1), 1–12.

McEver, R. A., & Manjunath, B. (2020). Pcams: Weakly supervised semantic segmentation using point supervision. arXiv:2007.05615 .

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32,* 8026–8037.

Pedersen, M., Bruslund Haurum, J., Gade, R., & Moeslund, T. B. (2019). Detection of marine animals in a new underwater dataset with varying visibility. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 18–26).

Rashid, A. R., & Chennu, A. (2020). A trillion coral reef colors: Deeply annotated underwater hyperspectral images for automated classification and habitat mapping. *Data, 5*(1), 19.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Richards, B. L., Beijbom, O., Campbell, M. D., Clarke, M. E., Cutter, G., Dawkins, M., . . . Williams, K. (2019). Auto- mated analysis of underwater imagery: Accomplishments, products, and vision. Retrieved from https://repository.library.noaa.gov/view/noaa/20234 (Technical Memorandum).

Salman, A., Jalal, A., Shafait, F., Mian, A., Shortis, M., Seager, J., & Harvey, E. (2016). Fish species classification in unconstrained underwater environments based on deep learning. *Limnology and Oceanography: Methods, 14*(9), 570–585.

Shester, G., Enticknap, B., Kincaid, E., Lauermann, A., & Rosen, D. (2017). Exploring the living seafloor: Southern California expedition. Oceana Report.

Šiaulys, A., Vaičiukynas, E., Medelytė, S., Olenin, S., Šaškov, A., Buškus, K., & Verikas, A. (2021). A fully-annotated imagery dataset of sublittoral benthic species in Svalbard, Arctic. *Data in Brief, 35,* 106823.

Siddiqui, S. A., Salman, A., Malik, M. I., Shafait, F., Mian, A., Shortis, M. R., & Harvey, E. S. (2018). Automatic fish species classification in underwater videos: Exploiting pre-trained deep neural network models to compensate for limited labelled data. *ICES Journal of Marine Science, 75*(1), 374–389.

Taylor, J., Lovera, C., Whaling, P., Buck, K., Pane, E., & Barry, J. (2014). Physiological effects of environmental acidification in the deep-sea urchin *Strongylocentrotus fragilis*. *Biogeosciences, 11*(5), 1413–1423.

Wicksten, M. K. (1989). Ranges of offshore decapod crustaceans in the eastern Pacific Ocean.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., . . . Wang, X. (2021). Bytetrack: Multi-object tracking by associating every detection box. arXiv:2110.06864 .

Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems, 30*(11), 3212–3232.