Title
Detection and Segmentation Using Less Supervision

Permalink
https://escholarship.org/uc/item/0914d98q

Author
McEver, R. Austin

Publication Date
2022

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Detection and Segmentation Using Less Supervision

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

R. Austin McEver

Committee in charge:

Professor B.S. Manjunath, Chair
Professor Shiv Chandrasekaran
Professor Kenneth Rose

December 2022

The Dissertation of R. Austin McEver is approved.

Professor Shiv Chandrasekaran

Professor Kenneth Rose

Professor B.S. Manjunath, Committee Chair

November 2022

Detection and Segmentation Using Less Supervision

Dedicated to my mother

Acknowledgements

First, I would like to thank my family for the support they have offered me to get me through school up to this point. In particular, I would like to thank my mother who has done more for me than anyone else ever has or ever will to ensure that I had the resources I needed to accomplish my goals.

My years at UCSB have been some of the best of my life, and I'll always remember them fondly. This is largely thanks to my friends and colleagues here who have made this time what it is. I want to thank members of the Vision Research Lab for their humor and support throughout this program. I want to thank the friends I've made through tennis over the more recent few years for keeping me company and helping me grow on and off the court. I want to also thank my friends from the Manley Co-op who brought laughter to my first few years of the program. I especially want to thank Maya for her humor, big heart, and support.

I'd also like to thank all of the educators who have impacted my life from kindergarten through graduate school who inspired me to continue on my academic journey for so long. I'd like to thank my PhD committee for their advice over the years. Among these educators, though, none have spent more time and effort on my work than Professor Manjunath, so I'd like to thank him in particular for his guidance over these past years as I worked to complete this dissertation.

Curriculum Vitæ
R. Austin McEver

Education

2022	Ph.D. in Computer Science (Expected), University of California, Santa Barbara.
2021	M.S. in Computer Science, University of California, Santa Barbara.
2017	B.S. in Computer Science, Minor in Engineering Entrepreneurship, University of Tennessee, Knoxville.

Experience

2018 - Present	Graduate Student Researcher, Vision Research Lab, University of California, Santa Barbara.
2021	Software Engineer - Machine Learning Intern, Facebook, Seattle, WA.
2018	Computer Vision Research Intern, Mayachitra Inc, Goleta, CA.
2017	Research Intern, OSIsoft LLC, Johnson City, TN.

Publications

R. Austin McEver, Bowen Zhang, and B.S. Manjunath. *Context-Matched Cut-and-Paste Collage Generation for Object Detection.* In review with IEEE Transactions on Multimedia.

R. Austin McEver, Bowen Zhang, Connor Levenson, A S M Iftekhar, and B.S. Manjunath. *Context-Driven Detection of Invertebrate Species in Deep-Sea Video.* Under revision with International Journal of Computer Vision.

Shafin Haque and **R. Austin McEver**. *Box Prediction Rebalancing for Training Single-Stage Object Detectors with Partially Labeled Data.* NeurIPS 2022 Workshop.

A S M Iftekhar, Satish Kumar, **R. Austin McEver**, Suya You, and B.S. Manjunath. *GTNet: Guided Transformer Network for Detecting Human-Object Interactions.* In review with SPIE.

R. Austin McEver and B.S. Manjunath. *PCAMs: Weakly Supervised Semantic Segmentation Using Point Supervision.* arXiv 2020.

Abstract

Detection and Segmentation Using Less Supervision

by

R. Austin McEver

Today's computer vision methods attempt to solve problems ranging from image classification to semantic segmentation. While some of these models are quite effective at their tasks, the most effective ones require a training set complete with a large set of heavily annotated data, but these large datasets and their annotations do not come without a cost. Dataset curators spend countless hours collecting data and even more data annotating it with semantic labels suitable for training today's methods. The expenses associated with data collection and annotation grow exponentially as the data becomes increasingly more scientific and difficult to annotate. While any lay person can take a photo of a dog and label it, collecting videos of the ocean floor and labelling the species in those videos can only be done with a budget sufficient to compensate a team of expert marine scientists. These costs motivate computer vision methods that can learn from less data, cheaper annotations, and less supervision. This thesis aims to provide some of these methods. We first introduce Point-supervised Class Activation Maps (PCAMs) to aid in semantic segmentation of images given only point level labels. Then, we introduce the Dataset for Underwater Invertebrate and Substrate Analysis (DUSIA), which comes with a limited set of partial labels. To address the challenges associated with learning from those labels, we train the Context Driven Detector with a Negative Region Dropping method, which enables better performance given partial labels. Finally, we introduce Context-Matched Collages as a means for generating additional training samples at a relatively low cost, leading to state of the art object detection performance on DUSIA.

Contents

Chapter 1

Introduction

The proliferation of imaging and video technology opens up many possibilities for scientific research. Scientists from all domains collect images and videos to aid in their research, but, to use these images and videos to their fullest extent, scientists must invest heavily in annotating the image and video data to distill the relevant information from the present pixel values. In many cases, this means hiring annotators to label images or videos, which can be time consuming, tedious, and expensive in many cases especially when the data requires a specially trained, domain-expert annotator.

This thesis first addresses semantic segmentation of natural images. That is, we aim to present a method that can label every pixel in an image as either a class of interest or background. In short, natural images are images where the objects of interest are common objects like cats, dogs, bicycles, bottles, or people. Labelling every single pixel by hand can be quite costly, but today's best semantic segmentation models require that human annotators label a large training set with pixel-level labels. These models then use that training set to learn how to label unseen images, often referred to as the validation and testing sets.

The first method presented in this thesis, Point-supervised Class Activation Maps

(PCAMs) aim to provide a means for training on images labelled with just a few points per image (i.e. using less supervision) to accomplish the semantic segmentation task. If segmentation can be accomplished with a training set containing just a few points per training image (rather than every single pixel in an image), then we may be able to automate segmentation using significantly less human annotation and resources associated with collecting those annotations.

This thesis then presents a more challenging, scientific dataset in the Dataset for Underwater Substrate and Invertebrate Analysis (DUSIA). Because these underwater videos are much more difficult to annotate than natural images, we tackle the problem of partially annotated object detection via the novel Context-Driven Detector (CDD). DUSIA's videos generally have less supervision than natural image datasets in that many individuals of species of interest in DUSIA may not have any label at all, and, alongside CDD, we present Negative Region Dropping as a method to address this partial supervision problem. Negative Region Dropping enables better object detection with less supervision.

To solve the object detection problem, a model should simply draw a bounding box around each object of interest rather than labelling every single pixel in an image. CDD addresses this problem by incorporating explicit context labels into the object detection process to enhance the model's prediction power. Using CDD and out of the box tracking algorithms, we are able to roughly count the invertebrates present in DUSIA, which is a major goal of the marine science projects that collect these videos.

Because the amount of data in DUSIA is somewhat limited compared to simpler datasets, we introduce Context Matched Collages that aim to generate new training examples. By increasing the size of our training set in this way, we can further improve object detection performance and better detect underwater invertebrates.

The remainder of this thesis aims to motivate and describe these major contributions in much more detail.

1.1 Motivation

While image and video technology enable new research and new forms of automation, the abundance of data generated by this technology can be overwhelming to deal with. Without metadata and annotation, images and videos are just more numbers to a computer. As a result, scientists and researchers allocate many resources to the collection of not only data but also to the annotation of that data so that the data itself can be of more use and so that future annotations might be automated.

The methods and contributions that follow generally aim to alleviate some of the burden associated with collecting annotations. We present methods that can learn to create pixel wise annotations from annotation of just a few pixels, methods that aim to detect every individual of a species in an image when only some are annotated, and methods to help make the most of existing annotations. These sorts of methods aim to reduce the expenses associated with the annotation process and aim to ultimately create a future where humans can spend less time and effort on tedious annotations and begin to rely more on computer automated annotation even for challenging, scientific data.

For example, semantic segmentation is one of the most challenging problems in computer vision, and collecting annotations for every pixel in a given image can take a long time for humans even when the image classes are relatively simple. Bearman et al [1] showed that collecting pixel level annotations on the VOC2012 natural image dataset [2] took on average 239.7 seconds per image. Collecting image level labels for the same images took on average 20.0 seconds per image, and point level labels where just a few pixels per image were annotated took only 22.1 seconds per image. Clearly, some types of annotation are much more difficult to collect than others, and annotation budgets are important to consider and allocate when taking on a new project.

As mentioned before, scientific data can be much more expensive to annotate, but much can be learned from scientific data. DUSIA's underwater videos are collected by the Marine Applied Research and Exploration (MARE) group who use them to advance their own marine research. They aim to survey the ocean floor to learn where in the ocean different species lives so that they can better influence policies to protect these underwater organisms. Beyond protection, MARE also simply wants to explore and learn more about life in the ocean, which can lead to new discoveries down the line. However, many of their resources must be dedicated to annotation. Currently, all of the video they collect must be reviewed by humans and annotated by hand, and it is done in multiple passes such that a video takes many times its length in human annotator time before it is fully annotated. Further, the humans who annotate this video must be trained scientist as laypeople have trouble distinguishing between the different species occurring in the video. For example, it can be difficult to distinguish between black coral and gray gorgonians as the remote operated vehicle (ROV) drives over and records the ocean floor.

This thesis aims to address some of those issues more via methods that can perform computer vision tasks with less supervision.

1.2 Challenges and Contributions

Creating computer vision methods to help create detailed annotations of relatively simple data, like images of dogs and cats, still presents a challenge for researchers. These days, many datasets like VOC2012 [3] and MS-COCO [4] (discussed in more detail in Chapter 2) contain many annotated examples of many common object classes typically found in so-called "natural images", but methods for automating annotation of this data are not perfect and still require a lot of data that is finely annotated. Chapter 3 will visit some of those applications in much more detail presenting a method that can enable the pixel-wise annotation of natural images after training on images that are annotated with just a few pixels each.

More specifically, Chapter 3 contributes a method for incorporating point supervision into a pipeline for weakly supervised semantic segmentation. This contribution enables the ability to train a semantic segmentation model with images labelled only with point or image level labels.

While collecting pixel level labels for natural images may take many resources because of the time required to make such detailed labels, almost anyone can annotate natural images because the classes of interest are common objects like cats, dogs, and people. However, many images and videos from scientific domains require specialized expertise to annotate. Requiring specialized knowledge makes the annotation process much more expensive because only trained scientists can annotate the data. As a result, creating methods that can automate annotation of this type of data can save a lot of resources, especially if these methods can learn to automate annotation with less training data.

As one might expect, though, creating models that can label scientific data is even more challenging than creating models that can label relatively simple data. In many cases, it is more challenging to distinguish among classes, the images and videos may be of lesser quality, and there may be less scientific data than natural data. Further, there may be less, noisy, or partial annotations when collecting annotations is more expensive.

Chapters 4, 5, and 6 address a specific instance of challenging scientific data, a solution to address the partial nature of many of this data's annotations, and a method for making the most of existing data via a collage generation method. Chapter 4 outlines the contribution of a new benchmark Dataset for Underwater Substrate and Invertebrate Analysis (DUSIA).

Chapter 5 contributes a method for counting invertebrate species in DUSIA's videos via a novel object detector, the Context-Driven Detector (CDD). CDD makes use of both explicit context labels and implicit context cues to enhance object detection performance on DUSIA. Chapter 5 also introduces Negative Region Dropping to enhance object detection performance given partial labels, as is the case with DUSIA's training set.

In addition to being partially labelled, DUSIA's training set also contains fewer annotated examples than many of today's natural image datasets due to the expenses associated with annotating DUSIA's scientific data. In order to enhance performance with less labels, Chapter 6 offers a method for generating additional samples from DUSIA's training set. Training with these additional samples leads to better object detection performance on DUSIA.

1.3 Organization

This dissertation is organized as follows

- Chapter 2 provides some background on the different tasks that will be discussed in future chapters as well as the types of supervision for those tasks.
- Chapter 3 presents a method for generating pixel-level annotations for natural images by training on a very limited number of annotated pixels per image in the training set.
- Chapter 4 illustrates the collection, curation, and annotation of the Dataset for Underwater Invertebrate and Substrate Analysis (DUSIA).
- Chapter 5 presents a method for demonstrating and leveraging the importance of context when detecting the invertebrates of DUSIA.

- Chapter 6 provides a method for generating synthetic, context-matched collages for expanding DUSIA's training set. By leveraging explicit context labels, more training samples can be generated that, when trained on, lead to better detection performance.

- Chapter 7 summarizes the main concepts of this dissertation and discusses potential future work that may further advance the field.

Chapter 2

Background

This chapter introduces several tasks and types of supervision for those tasks to lay a basis for the problems and methods that will be addressed in future chapters.

Common computer vision tasks and problems include different forms of image classification, object detection, and semantic segmentation. These tasks are typically addressed by training some type of machine learning model to perform the task using different training sets that have varying levels and types of supervision. This chapter aims to give a broad overview of those tasks and different types of supervision for those tasks.

2.1 Image Classification

Many computer vision tasks center around one of the field's most basic tasks: image classification. Image classification involves assigning one (single class) or more (multi-class) semantic labels to an image. These semantic labels typically simply describe what is in the image and are commonly referred to as image-level labels as they describe what is present at the broad image level.

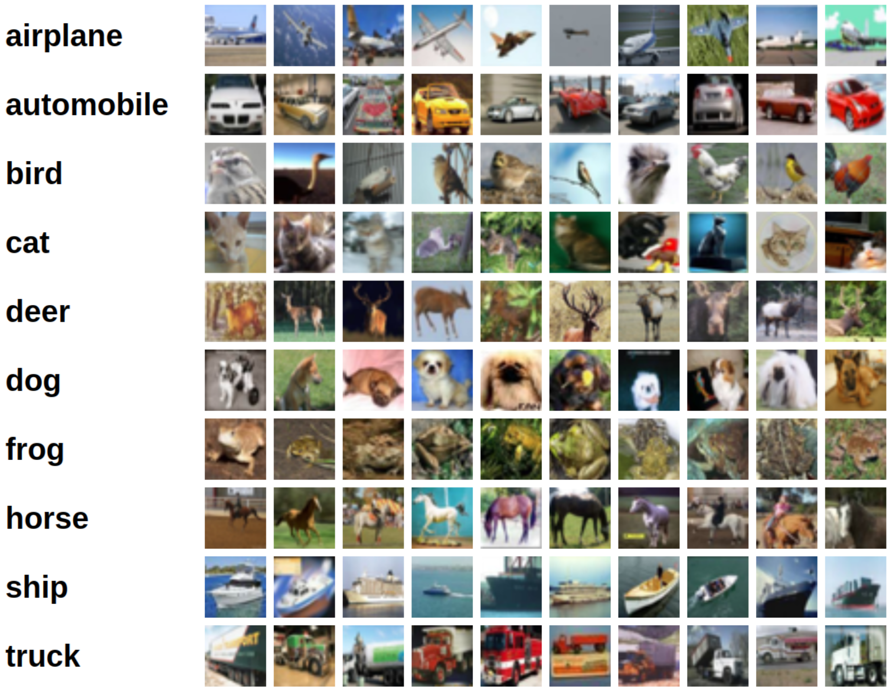

Figure 2.1: Examples of CIFAR-10 image classes taken from the CIFAR-10 website [5]

2.1.1 Common Datasets for Image Classification

The CIFAR-10 Dataset presents a simple, common example of an image classification task [5]. CIFAR-10 consists of 60,000 tiny images with ten, mutually exclusive, common object classes including cat, dog, and truck. Researchers commonly implement and test different image classification models that aim to classify each of CIFAR-10's images as containing an instance of one of the set's ten classes. Figure 2.1 shows example images for each of the ten classes.

Many well known image classifiers present results on the ImageNet dataset, associated with the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [6]. ImageNet includes a growing collection of images with currently over 14 million images. The primary benchmark has 1,000 image categories, and each category has at least 1,000 image examples. The dataset was originally introduced purely for the image classification (aka recognition) task, but they have since added annotations for object detection, a task which will be discussed next. The ImageNet dataset has also become the standard dataset for pre-training models. That is, many models that are provided for public use have weights from training on ImageNet, which allows faster convergence on other datasets.

As another example of image classification, Chapter 5 addresses a form of image classification where the image classes simply include the substrate present in underwater video captured near the ocean floor. Because multiple substrates may be present in a given video frame, this image classification tasks allows multiple labels.

2.1.2 Image Classification Evaluation

Image classification can be evaluated with several different metrics. Confusion matrices make up the basis for many of these metrics. Confusion matrices have four categories: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). True or false indicate whether a prediction is correct and positive or negative indicate whether the prediction is positive or negative.

For example, if an image contains a dog and not a cat, it is positive for dog and negative for cat. If the prediction is that both dog and cat are present, then we have a true positive for dog (dog predicted, dog present) and false positive for cat. (cat predicted, cat not present) If the prediction is that neither a dog nor cat is present, then we have a false negative for dog (dog not predicted, dog present) and true negative for cat (cat not predicted, cat not present).

Precision, recall, and accuracy derive from the confusion matrix.

$$precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

These three metrics can be used to evaluate image classification, though for many popular datasets, accuracy is the most common. Because models typically predict some confidence score for each class of interest, researchers commonly present both top-1 and top-5 accuracy, especially for single class classification problems. For example, consider the top predictions for an image include cat with confidence 0.4; dog with 0.2; car, 0.1; bus, 0.01; truck, 0.2. If the ground truth is cat, then this is TP example for cat and TN for all other classes. In a top-5 scenario, this prediction would be a TP if the ground truth is any of the top five predicted classes and incorrect otherwise. Most datasets do not include images of objects that are not classes of interest (i.e. there are no TNs).

Precision and recall can be plotted on a curve together at a range of different thresholds, giving more insight to the overall performance of a classifier. Confidence of predictions are the typical thresholds used for image classification. Figure 2.2 shows an example precision-recall curve. The area under the precision-recall curve, AUC, also effectively demonstrates the performance of a classifier and gives insight to how well its confidence values reflect the ground truth.

If a given image may contain multiple classes, mean Average Precision (mAP) reflects performance of a classifier. mAP aims to reflect AUC for each given class by taking the mean of average precision (AP) over each class. Average Precision is the mean of precision values at many points along the ROC curve.

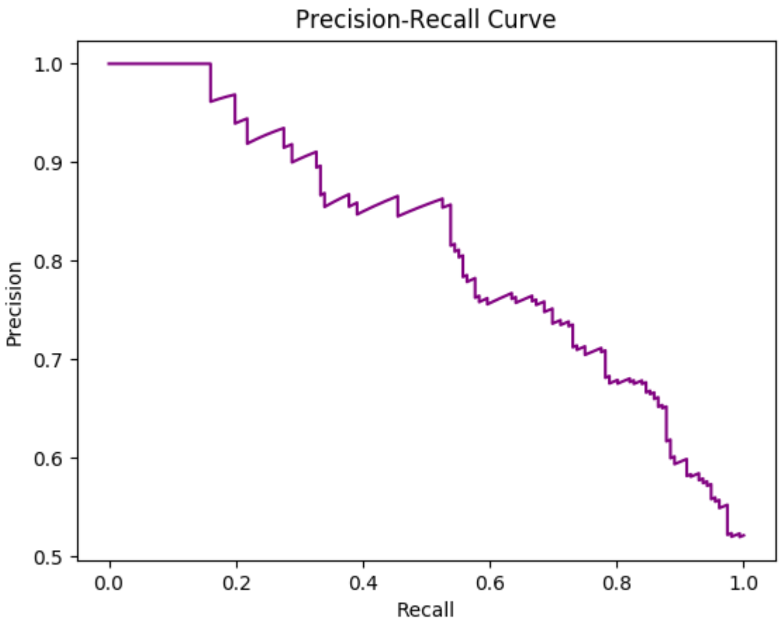

Figure 2.2: Example Precision-Recall curve for a classifier

2.2 Object Detection

Object detection consists of classifying objects into categories and goes beyond image classification by also localizing those objects in a given image. For the object detection task, bounding boxes typically provide a means of localizing objects by indicating the extremes of an object. The top of a bounding box should touch the topmost pixel of a given object, the left of the bounding box should touch the leftmost pixel of the object, etc.

Chapters 5 and 6 largely concern object detection and related tasks.

2.2.1 Common Datasets for Object Detection

The Visual Object Classes Challenge 2012 (VOC2012) [3] presents participants with multiple tasks including object detection for 20 common object classes as well as semantic

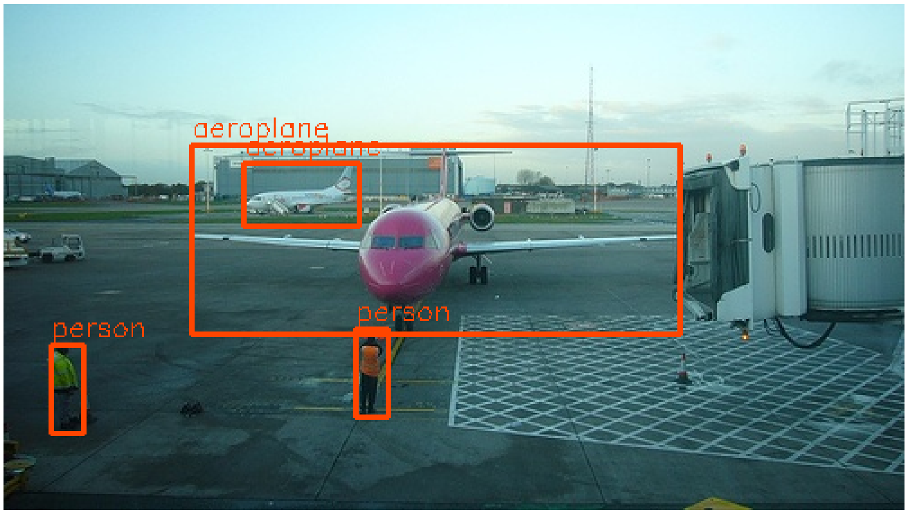

Figure 2.3: Image from the VOC12 dataset [3] with drawn on bounding box examples.

segmentation of images using those twenty classes, though semantic segmentation will be discussed in the next section. VOC's images also fall into the natural image category, much like CIFAR-10 because the classes of interest include common objects and classes like person, cat, dog, and car. Natural images, typically contrasted with more scientific images, are images of common objects people might see or interact with in daily lives. Natural images and their associated object categories can be labelled by any layperson. Figure 2.3 shows an example of an image from VOC2012 with drawn on bounding boxes. VOC2012 provides 11,530 images with 27,450 bounding box annotated objects and 6,929 segmentations.

As mentioned in the previous section, ImageNet also contains bounding box annotations for the object detection task [6]. Currently, the dataset includes over 400,000 images with bounding box annotations for 200 fully labelled object categories. ImageNet also provides annotations for 30 of those 200 categories for object detection from video.

Microsoft COCO (Common Objects in Context) contains 330,000 images with annotations for over 200,000 [7]. Like ImageNet, COCO has grown over the years and now includes 80 object categories and over 1.5 million object instances. A few of the

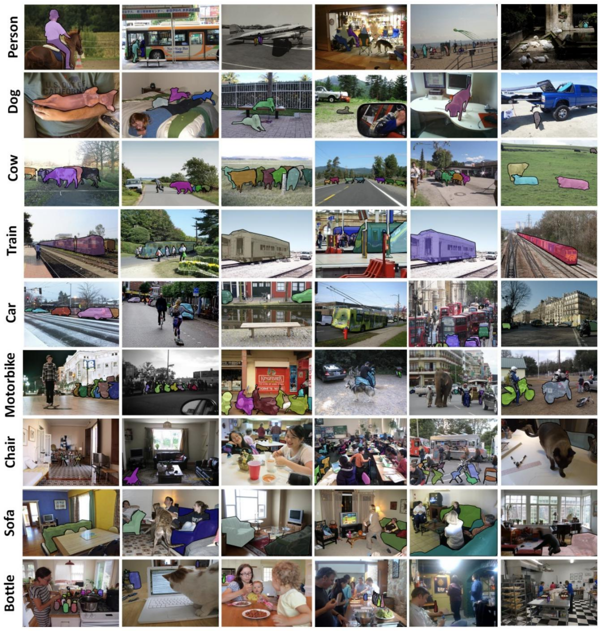

Figure 2.4: Examples from the COCO dataset taken directly from the original paper [7]

object categories are visualized in Figure 2.4. As the name suggests, COCO aims to present a collection of natural images of common objects found in their natural contexts. For example, there are images of trains on railways, sofas in living rooms, and cows in pastures.

2.2.2 Object Detection Evaluation

Researchers typically use the mean Average Precision (mAP) metric to quantitatively evaluate a model's performance on the object detection task. The major difference between using mAP to evaluate object detection and to evaluate image classification is that in image classification, each image has an entry in the confusion matrix, whereas, in object detection, each object has an entry in the confusion matrix.

To determine whether a given box is correct or not, researchers determine whether each box has a high enough intersection over union (IOU) with a ground truth box. Figure 2.5 illustrates an example of intersection over union for boxes. Dividing the intersecting

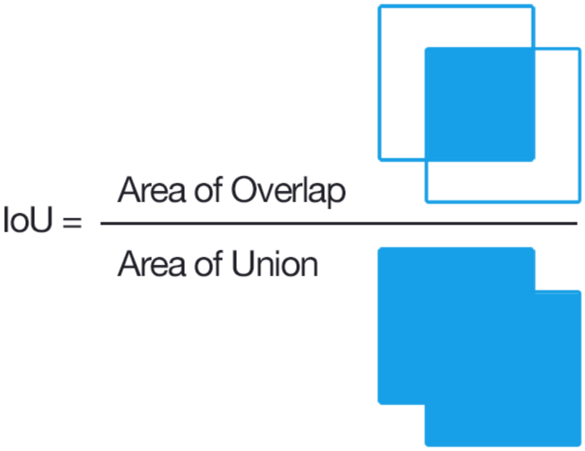

Figure 2.5: Intersection over union illustration from Adrian Rosebrock's PyImage-Search. Accessed September 2022. [8]

area of two boxes with the union of their area gives IOU. Typically, a box is labelled as correct if it has at least an IOU of 0.5 with a ground truth box. Additionally, each ground truth box can correspond to only one prediction. Therefore, if multiple predictions have high IOU with the same ground truth box, only the prediction with the highest IOU will count as correct, while the remaining predictions will be considered negative.

2.3 Semantic Segmentation

Semantic segmentation increases the granularity of object detection from a bounding box localization to a pixel-wise labelling. That is, semantic segmentation consists of labelling every pixel in an image as a given class of interest or background. Typically, the background class is used to cover any pixel that is not part of a class of interest. Some datasets, like VOC2012, also include an "ignore" class which is used at uncertain pixels at the edge of objects. "Ignore" class pixels are typically ignored during both training and evaluation. Figure 2.9 shows an example image and its associated semantic segmentation labels from the VOC 2012 dataset.

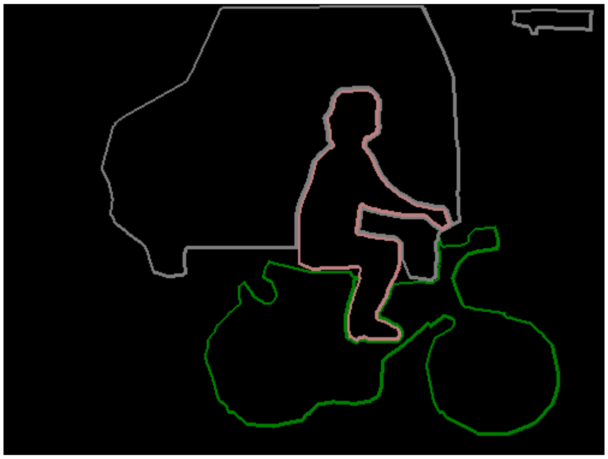

Figure 2.6: Example of SBD's semantic contour labels, taken directly from the original work [9]

Given the fine granularity of a pixel level semantic segmentation of an image, semantic segmentation is the most challenging of tasks presented here. Today's semantic segmentation models can generally segment large objects, but segmenting smaller objects, especially those with thin parts (e.g. chair legs, bicycle wheel spokes, etc) remains a difficult challenge. Recovering the fine details of objects is especially difficult when given limited training data or limited labels, but this challenge leaves room for further research into more advanced methods.

Chapter 3 introduces a novel method for generating semantic segmentation and illustrates its effectiveness on underwater images.

2.3.1 Common Datasets for Semantic Segmentation

As mentioned in Section 2.2.1, VOC2012 provides 6,929 segmentations for its 20 classes [2]; however, the Semantic Boundaries Dataset (SBD) [9] provides semantic contours for more of the VOC images. SBD provides precise labels for object boundaries, which they argue provides a more stringent metric for segmenting objects. Figure 2.6 shows an example of SBD's contour labels. These boundaries can trivially be converted to semantic

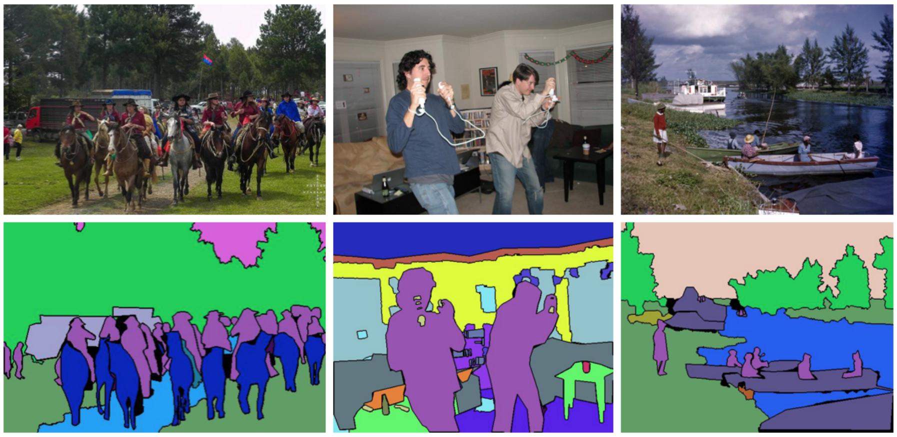

Figure 2.7: Example images and their associated semantic segmentation labels from the COCO Dataset. Each color corresponds to a given class [7].

segmentation labels. With the addition of SBD's labels, there are 10,582 training images and 1,449 validation images in VOC with semantic segmentation labels.

In addition to labels for object detection, COCO provides labels for 91 "stuff" classes in addition to their 80 object categories [7]. Whereas object categories include things like "person", "car", or "fork", "stuff" categories may have less defined boundaries for segmentation. Examples of the "stuff" classes include grass, wall, and sky. Figure 2.7 shows a few example images and their segmentation labels.

ImageNet also provides segmentation lables for 150 semantic categories across about 20,000 images [6]. The number of images and categories decrease significantly from the 14 million images available for image classification because of the expense associated with collecting semantic segmentation labels.

The Cityscapes Dataset provides a large-scale dataset with video recorded in street scenes. These videos include footagge from 50 different cities, and the dataset includes pixel-level annotations of 5,000 frames as well as less precise labels for another 20,000 frames. Learning to segment and understand Cityscapes' frames can help develop better urban scene understanding with important applications in self driving cars and robotics.

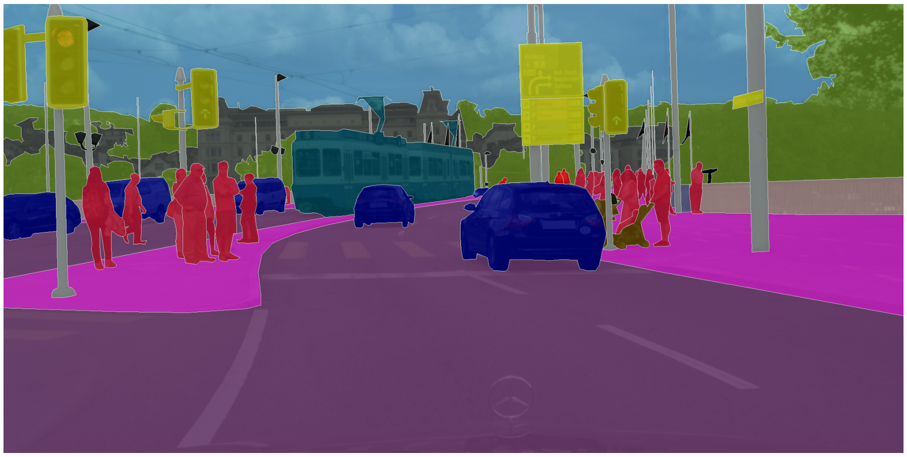

Figure 2.8: Example labelled image from the Cityscapes Dataset. Each color corresponds to a given class [10].

Cityscapes labels 30 classes including road, person, bus, tunnel, traffic sign, and vegetation. Figure 2.8 shows an example of a labelled frame from Cityscapes.

2.3.2 Semantic Segmentation Evaluation

Researchers typically use the mean Intersection over Union (mIOU) metric to quantitatively evaluate a model's performance on the semantic segmentation task. In this case, IOU is computed in the same way as for object detection; however, the ground truth and prediction shapes can be any shape rather than restricted to rectangles. This generalization does not change the computation for IOU. Taking the mean of IOU for each class gives the mIOU metric.

2.4 Types of Semantic Supervision

Each task presented thus far requires some sort of semantic labels. Semantic labels simply assign meaning, typically via a word or two, to some group of pixels. The group of pixels may be an entire image (such as as in image classification), just the pixels within

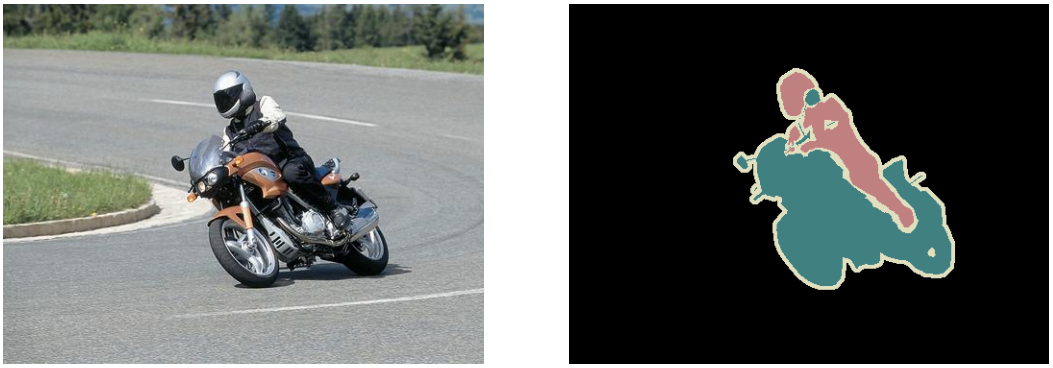

Figure 2.9: Example image and semantic segmentation label from the VOC2012 Dataset. Pink indicates the person class; white, ignore; blue, motorcycle; and black, background.

a bounding box (object detection), or a carefully segmented group of pixels (semantic segmentation). Image level semantic labels for Figure 2.3 might include “airport”, “airplane”, and/or “person”. The bounding box labels shown are also semantic labels.

To train a model to complete each of these tasks requires a training set. A training set consists of a collection of images and their semantic labels. During training, the model has full access to these images and labels so that the model can learn to complete the task on new images that are not in the training set. The model is supervised by the training set, and the type of semantic labels in the training set determines the type of supervision that the model receives.

In a fully supervised scenario, models that address a given task have access to a training set that contains labelled examples that match the granularity of the assigned task. For example, if tasked with semantic segmentation of a set of unseen images, a fully supervised model will be provided with a set of images that also contain semantic labels for every pixel of every image in the training set.

However, full semantic segmentation labels, like shown in Figure 2.9, can be expensive, difficult, or sometimes impossible to collect. For example, Bearman et al estimate that the semantic segmentation labels for VOC2012 took on average 239.7 seconds per image in contrast to the estimated 20.0 seconds per image required to collect image-level labels. In images collected in more technical, scientific scenarios, such as the segmentation of images of underwater, sessile organisms that will be described in the next chapter, pixel-level labels may be prohibitively expensive to collect.

The expenses associated with collecting semantic labels necessitate exploration of methods that can use fewer, weaker, partial, or less expensive labels. The work described in this dissertation largely aims to explore two of those cheaper forms of labels: weak and partial labels.

2.4.1 Weak Supervision

The amount of information present in a given type of label determines its strength. For the tasks and labels described so far, pixel level labels would be strongest followed by bounding box labels followed by image level labels. Somewhere in between lie squiggle and point level labels. Point labels provide the class of a single given pixel, provide less information than bounding box labels, and are easy and fast to collect. Squiggle level labels extend point level labels to more pixels but take longer to collect. Figure 2.10 provides examples of these types of labels.

In the most common weakly supervised scenario, a weakly supervised semantic segmentation model would be provided with a training set consisting of images with only their image level labels. During training, the model would be given access to the training images and their image level labels. During inference time, the model would be expected to output pixel level labels for unseen images. Along those lines, a weakly supervised model for semantic segmentation may be provided with bounding box or point level labels during training. While less common, this scenario may fit a given application, and will be described in more detail in Chapter 3.

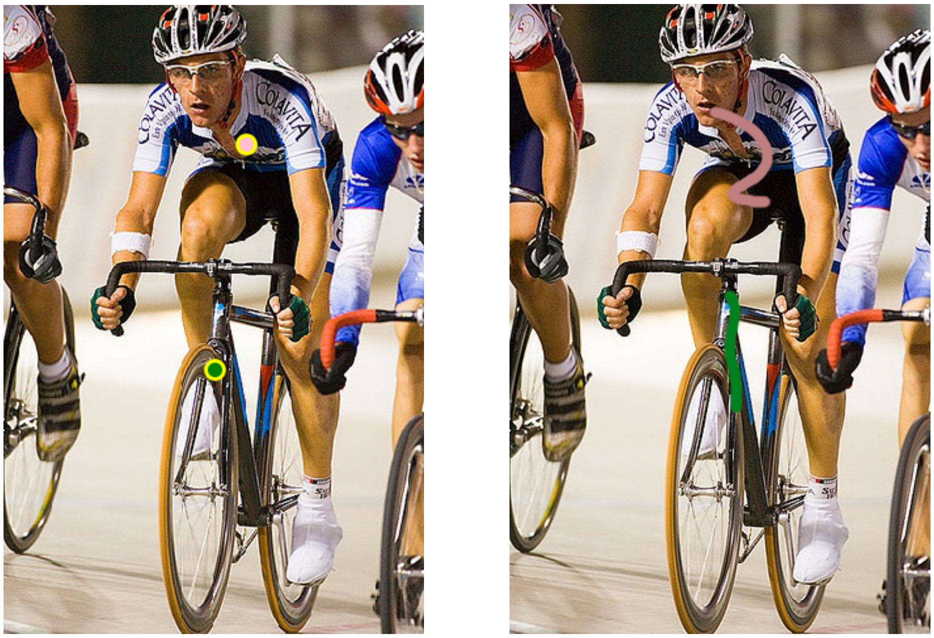

Figure 2.10: Point level (left) and squiggle level (right) labels. Pink indicates the person class; green, bicycle. Yellow highlights the point labels, which are inflated from a single pixel, for visibility.

2.4.2 Partial Supervision

A set's labels' *completeness* lie on a spectrum separate from the strength of a set's labels. The vast majority of datasets used to train and validate today's computer vision models aim to contain complete labels, meaning that each instance of a class of interest is labelled as such. Any missing labels are typically accidental noise in the set of labels. However, it can be prohibitively expensive to collect a collect labelling in some scenarios, especially with more technical, scientific data where annotator time is more expensive and limited.

Figure 2.11 demonstrates a complete and partial labelling for an image in the VOC2012 dataset. Because the classes of interest for its task include person and motorcycle, a complete labelling of this image must contain a bounding box for each instance of each motorcycle, person, and every other class of interest. A partially supervised image may contain bounding box labels for only some instances of each class of interest. Similarly,

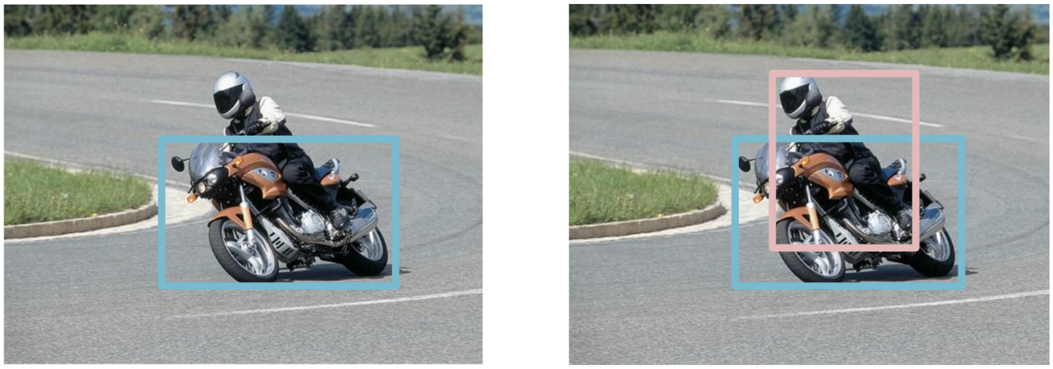

Figure 2.11: Partial (left) and complete (right) labels. Pink indicates the person class; blue, motorcycle.

Figure 2.10 demonstrates more examples partially labelled images because every person and every bicycle are not labelled.

Chapter 5 describes a scenario where collecting bounding box labels for every instance of every class of interest in underwater video is prohibitively expensive, so we present a method for object detection on partially supervised video frames.

2.4.3 Other Types of Supervision

While this dissertation primarily focuses on weak and partial supervision, researchers also explore other types of supervision, and a quick explanation may preempt any confusion about their relationship to weak and partial supervision. A brief description of a few different forms of supervision are as follows:

- **Mixed supervised** methods may learn from many different types of labels simultaneously. For example, some images in the training set may have a some point supervised images, some bounding box supervised images, and some images with labels only at the image level.

- **Semi supervised** methods may learn from some labelled data and some unlabelled data simultaneously. Typically, the labelled data will all have the same types of labels.

- **Unsupervised** methods may learn from data with no labels at all.

Chapter 3

Point Supervised Semantic Segmentation

3.1 Motivation

Semantic segmentation of an image typically involves labelling every pixel as a class of interest. This fine-grained labelling enables a plethora of applications in robotics, medicine, biology, and marine science. While the task may be easy for humans, it remains a huge challenge for computer vision algorithms, especially when the images requiring labelling contain difficult or scientific categories.

Our model to address this issue is trained and validated on the fully labelled VOC2012 dataset [3]. While training on the VOC2012 Dataset, the method only has access to point level labels created as part of the work done by Bearman et al [1].

Bearman et al [1] collected point level labels for VOC2012's images. They asked annotators to simply click once on each instance of a class of interest and give the label of that object. They showed that, using their method vs other state of the art methods, that point supervision resulted in the best performance given an annotation budget.

While only point level labels may be available to the model at training time, VOC2012 validation set's publicly available pixel level ground truth labels enable quantitative and qualitative evaluation.

Our method requires a few steps. First, we train a classification network to generate PCAMs, which are Point-supervised Class Activation Maps. We train the Inter-pixel Relation Network, IRNet [11], on our PCAMs to generate matrices of inter pixel realations. Next, we use these matrices to refine the PCAMs to better adhere to object boundaries. Finally, we train the fully supervised DeepLabv3+ [12] model on our refined PCAMs to predict semantic segmentation labels.

The contents of this chapter have been presented in PCAMs: Weakly Supervised Semantic Segmentation Using Point Supervision [13].

3.2 Related Work

3.2.1 Weakly Supervised Object Localization

Weakly supervised localization refers to the problem of finding areas of interest in an image given weak labels. This problem has also been referred to as saliency detection. These methods [14–16] attempt to find class generic regions that are likely to contain some class of interest using varying levels of supervision. Similarly, methods such as the one presented by Cholakka et al [17] work to localize areas that are likely to contain a specific classes of interest.

Following the path of several methods which use CNNs for weakly supervised object localization [18–21], CAMs as created by Zhou et al [22] have become the basis for many weakly supervised class specific localization problems including some of the methods that will be discussed in Section 3.2.2 [11, 23, 24].

In short, Zhou et al [22] propose a method for interpreting the activation of a CNN trained for classification that provides localization information for the classes of interest. This localization is done by adding a global average pooling (GAP) layer to a classification network and examining the activations of the final convolutional layer in the classification CNN. The CNN has class dependent weight vectors that learn which activations correspond with each class. By weighting the feature map of the last convolutional layer in the CNN and upsampling the weighted map to the image size, the authors can begin to localize which areas of an image correspond to different classes. Section 3.3.1 further discusses this method.

Ultimately, this process allows for coarse localization of classes of interest that can be derived from a network trained with only image level labels. Other methods have built on CAMs for creating even better class specific localization extended beyond the image domain [25,26]. Zhang et al [27] proposed a method for iteratively improving localization performance by erasing areas of high activation from training images, forcing the CNN to learn other features associated with classes of interest thereby expanding areas of localization.

Still, this line of work is only able to achieve coarse localization, and the resultant activation maps rarely cover the full extent of classes of interest. These activation maps also correspond poorly with object boundaries.

3.2.2 Image-level Supervised Segmentation

Given the ease of collecting image level labels, many efforts have recently been made to create effective segmentation algorithms that use only these labels [11,23,24,28–42].

Kolesnikov et al [40] propose a method for training a segmentation CNN with a loss that guides the network via its loss function to follow localization cues, expand around those cues, and adhere to object boundaries. Several methods follow a similar path including work done by Huang et al [34] which uses CAMs to generate its localization cues. Similarly, their loss function has a term for adhering to these cues, growing their regions according to some similarity criteria, and adhering to object boundaries.

The most relevant methods to this paper include recent works of Ahn et al [11, 23] which also generally rely on the method of Zhou et al [22] to create CAMs. Ahn's method generates CAMs, mines affinity labels from the CAMs, and presents a neural network that outputs information that can be used to generate a transition probability matrix. In one of their works [23], Ahn et al describe a method for carefully exploiting CAMs to generate positive and negative affinity labels for pixel pairs. Positive affinity labels should indicate a pair of pixels is of the same class while negative labels indicates differing classes. These labels can be generated automatically by giving thresholds for confidence in CAMs such that two pixels that have high confidence in the same class are assigned a positive affinity label.

Ahn et al [11, 23] present different networks and methods for learning from these mined affinity labels, but they output information that is ultimately used to generate a transition probability matrix. They then perform a random walk over CAMs using the computed transition probabilities to propagate class labels and refine CAMs into pseudo segmentation labels. Finally, these pseudo labels are used in place of ground truth to train a CNN that is designed to perform segmentation given real ground truth labels.

The method presented in this chapter follows this type of method with the novel introduction of point supervision during early stages that leads to a significant improvement in performance.

3.2.3 Point-level Supervised Segmentation

While image level supervision certainly has its place in domains where images can easily be collected for classes of interest, point supervision can significantly improve performance of segmentation algorithms and can easily be collected when annotating a new dataset. To our knowledge, no publications have tackled the problem of point supervised segmentation algorithms since Bearman et al [1] introduced the problem, collected and provided point level annotations for the PASCAL VOC 2012 dataset [3], and proposed a method for incorporating point supervision and localization cues into training a normally fully supervised network directly. This method computes a loss over supervised points that is based on a log softmax probability.

The work by Mainis et al [43] is sometimes mentioned in literature as using point level annotations to achieve semantic segmentation; however, this paper uses extreme points rather than more random points as collected in Bearman's work where annotators are simply asked to click on objects of interest [1].

Rather than point-level annotations, then, [43] more closely resembles bounding box level supervision. Arguably, this method uses even more supervision than bounding boxes in that extreme points give bounding box information in addition to four points that are certainly on the boundary of a given object. When given only a bounding box, it is uncertain which parts of the bounding box edges actually lie on an object boundary.

3.3 Method

This chapter introduces a method for training PCAMs: point supervised class activation maps as shown in Figure 3.1. Including point supervision in the training process in this way significantly improves the localization performance of CAMs. To achieve state of

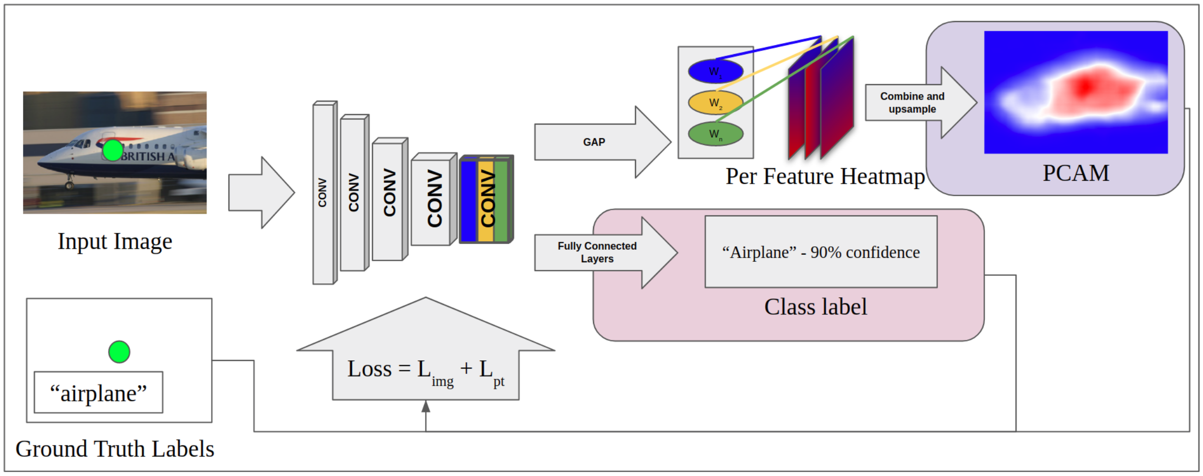

Figure 3.1: PCAM training overview: compute PCAM with a point supervised input image using the ResNet50 [44] backbone, produce the class labels, and compare the output with the supervised point and class labels to compute a loss used to update the PCAM network. Note that the last convolutional layer's feature maps are colored to match with their class dependent weight vector to produce per feature heatmaps, which are then combined and upsampled during training to generate the PCAM.

the art semantic segmentation results, we train IRNet [11] on our PCAMs and use its output to refine PCAMs to create pseudo semantic segmentation labels. Finally, we train the fully supervised segmentation network presented in [12], DeepLabv3+, on the pseudo labels and use this network for final predictions.

3.3.1 PCAMs

To train PCAMs, we present an addition to the work by Zhou et al [22] that allows the inclusion of point supervision during training as shown in Figure 3.1. The original method for producing CAMs adds a global average pooling layer to an image classification CNN. Following the work by Ahn et al [11], we use ResNet50 [44] as the backbone classification network and drop the stride of its last downsampling layer to prevent too much reduction in resolution. Training the image classification CNN is done with a simple multilabel soft margin loss.

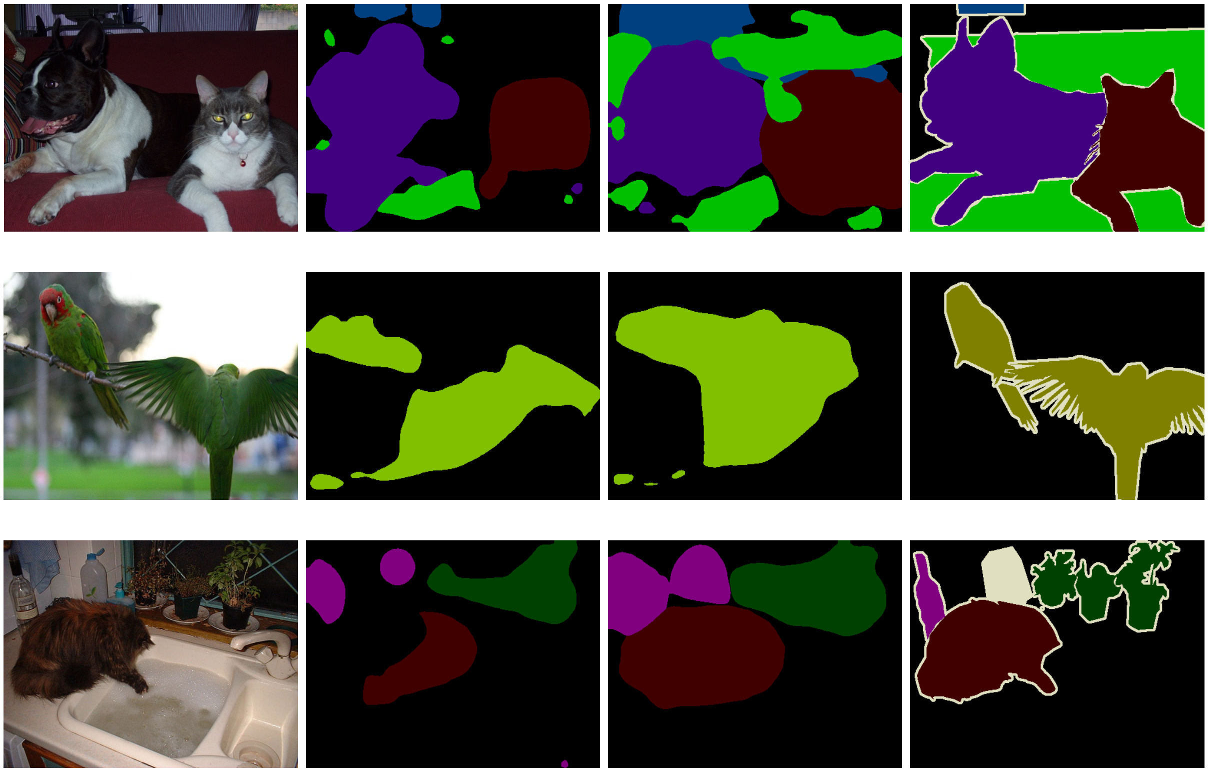

Figure 3.2: From left to right: original images, CAM labels, PCAM labels, and ground truth segmentation for each image

$$L_{img}(\hat{y}, y) = -\frac{1}{C} * \sum_{i=1}^{C} (y_i * \log((1 + \exp(-\hat{y}_i))^{-1}) + (1 - y_i) * \log(\frac{\exp(-\hat{y}_i)}{1 + \exp(-\hat{y}_i)})) \tag{3.1}$$

where y is the one hot encoded vector of classes in the images, $\hat{y}$ is the predicted class vector, y_i is the label of the ith class, and C is the number of classes. Once trained, CAMs of a given class c can be generated by

$$M_c(\mathbf{x}) = \frac{\phi_c^\top f(\mathbf{x})}{max_{\mathbf{x}} \phi_c^\top f(\mathbf{x})} \tag{3.2}$$

where ϕ_c represents the learned class-dependent weight vector for class c, f is the feature map from the final convolutional layer of the classification network backbone, and x is a 2D coordinate in f. As mentioned before, we use ResNet50 [44] as our backbone network with a reduced stride in the final downsampling layer. The dimensions of the CAMs are then 1/16 of the input image.

Our method for training the classification network with point annotations, however, requires that we generate the CAMs during training. To align the CAMs with the input image, we use bilinear interpolation to upsample the CAMs to the size of the input image to create U_c, the upsampled CAM for class c. This upsampling is only done during inference in previous works, but our method introduces this upsampling during training so that we can compare the upsampled CAM with any supervised points and guide the network for better activation mapping. We then compute the mean squared error loss over each of the supervised points as follows:

$$L_{pt}^{c} = \frac{1}{|S|} \sum_{s \epsilon S} (U_c(s) - G_c(s))^2 \tag{3.3}$$

where S is the set of supervised pixel locations, $U_c(s)$ is the predicted probability that location s is of class c, and $G_c(s)$ is the binary ground truth label for class c for the pixel at location s. For the point supervised term of the loss for training a network to generate PCAMs, we average the classwise losses for each class present in the training image.

$$L_{pt} = \frac{1}{|C'|} * \sum_{c \epsilon C'} L_{pt}^{c} \tag{3.4}$$

where C' is the set of classes in the training image. This allows us to precisely use any point level annotations to guide the network to activate at specific locations. This loss term also leads to more confident activation maps that cover a greater spatial extent of objects of interest. The total loss for training the PCAM network is then

$$L = L_{img} + \alpha L_{pt} \tag{3.5}$$

where α is a weighting term. We tested several

3.3.2 Refining PCAMs via IRNet's Semantic Affinity Predictions

We train the Inter-pixel Relation Network, IRNet, [11] using our PCAMs and use the refined maps to achieve more refined semantic segmentation. To train IRNet, we mine semantic affinity labels from our improved PCAMs, which use point level supervision, rather than mining labels from CAMs which use image level supervision, and use these mined semantic affinity labels for training IRNet.

In order to mine semantic affinity labels, we threshold each PCAM such that low values are considered background and high values are considered as the corresponding class. We ignore all pixels that have middling confidence values as their labels are uncertain and affinity labels must be mined reliably to optimize the performance of IRNet.

We then examine all pixels within a small radius of confident pixels. If a pair of pixels are both confident or background and have the same label, the pair is assigned a positive affinity label, and if it has a different label it is assigned a negative affinity label.

IRNet uses training images and these mined affinity labels to learn to predict a displacement vector field and class boundary map. The displacement field should indicate centroids of class boundaries, which aims to segment class instance but also aids in separating instances of adjacent differing classes. The class boundary map aims to indicate boundaries of classes.

The class boundary map can be synthesized to create an affinity matrix. If a pair of pixels have a positive class boundary pixel on the line between them, they have low semantic affinity. Otherwise, they are likely to be of the same class and have high semantic affinity. Next, the semantic affinities are used to compute a transition probability matrix for a random walk which is performed over instance-wise PCAMs.

Guided by this transition probability matrix, a random walk over PCAMs expands and refines activation areas. These refined PCAMs are then combined to create pseudo semantic segmentation labels. More details on IRNet and semantic affinity labels are provided in the source paper.

3.3.3 Training a Fully Supervised Segmentation Model on Refined PCAMs

The final step in our method takes the pseudo semantic segmentation labels generated via random walk propagation over PCAMs and uses them as ground truth for training DeepLabv3+ [12], a state of the art semantic segmentation model. We find that, despite training on imperfect labels, the network is still able to generate slightly better segmentation results on unseen images when compared to refined PCAMs.

Further, using a trained fully supervised network makes inference simpler. To infer a new image otherwise, we would need to first run inference using the trained PCAM network and run inference using IRNet before we could compute the transition probability network from IRNet's output. Finally, we would need to run the random walk algorithm on the PCAM using the generated transition probability matrix to refine the PCAM.

After we train DeepLabv3+ on our pseudo semantic segmentation labels, inference on an unseen image can be done simply by running inference with the trained DeepLabv3+ model.

3.4 VOC2012 Experiments and Results

The following section describes our experiments on the PASCAL VOC 2012 dataset [3] on which we achieve state of the art performance for point-supervised semantic segmentation.

After discussing the experiments on PASCAL VOC 2012, we explore the method's performance on the images collected by UCSB marine science researchers.

3.4.1 Dataset

All of our experimental results are reported on the PASCAL VOC 2012 dataset [3] and trained using the PASCAL VOC 2012 training images supplemented with the images from Semantic Boundaries Dataset, SBD [9], following common practice [1, 11, 23]. The PASCAL VOC 2012 dataset includes 1,464 training images and 1,449 validatation images. SBD contains annotations for 11,355 images from the PASCAL VOC 2012 dataset. In total, we train with 10,582 training images and test with 1,449 validation images.

For training PCAMs, we use the point level labels provided by Bearman et al [1]. These labels include one ore more annotated points per class in each training image. Overall, we use an average of approximately 2.4 points per image for training.

3.4.2 Hyperparameters

PCAM Network

The PCAM network uses ResNet50 [44] as its backbone network. The learning rate is initially set to 0.001 for the backbone parameters and 0.01 for the classification layers. The loss weighting term α is set to 0.1.

Activation Map	Sup.	*train*	*val*
CAM	I	48.3	46.0
PCAM	P	56.5	54.4
PCAM	F	64.0	55.5

Table 3.1: mIOU comparison of CAM, PCAM, and PCAM trained with all points on the PASCAL VOC 2012 *train* and *val* sets

Refined Activation Map	*train*	*val*
CAM	66.5	57.4
PCAM	70.8	68.5

Table 3.2: mIOU comparison of performance on PASCAL VOC 2012 *train* and *val* sets of CAMs and PCAMs after being refined by random walk via IRNet's transition matrix

IRNet

We generally use IRNet as presented by Ahn et al [11]. We set the CAM evaluation threshold for producing the labels for IRNet to 0.3. Similarly, we set the threshold for semantic segmentation at 0.3. These adjustments compensate for PCAMs being somewhat more confident than CAMs. For our setting, we had slightly better results setting IRNet's β parameter to 12. This parameter affects the generation of the transition matrix that is used for the random walk propogation of attention scores and is detailed in the paper by Ahn et al [11].

DeepLabv3+

We train DeepLabv3+ [12] using the labels generated from refining PCAMs with IRNet. We use a ResNet backbone, a learning rate of 0.007, weight decay of 4e-5, and a Nesterov momentum optimizer with a momentum of 0.9. We use an output stride of 16 during training and evaluation, and we do not implement multi-scale or flipped inputs during training. This is the same training setting as DeepLabv3+ shown in Table 3.3; however,

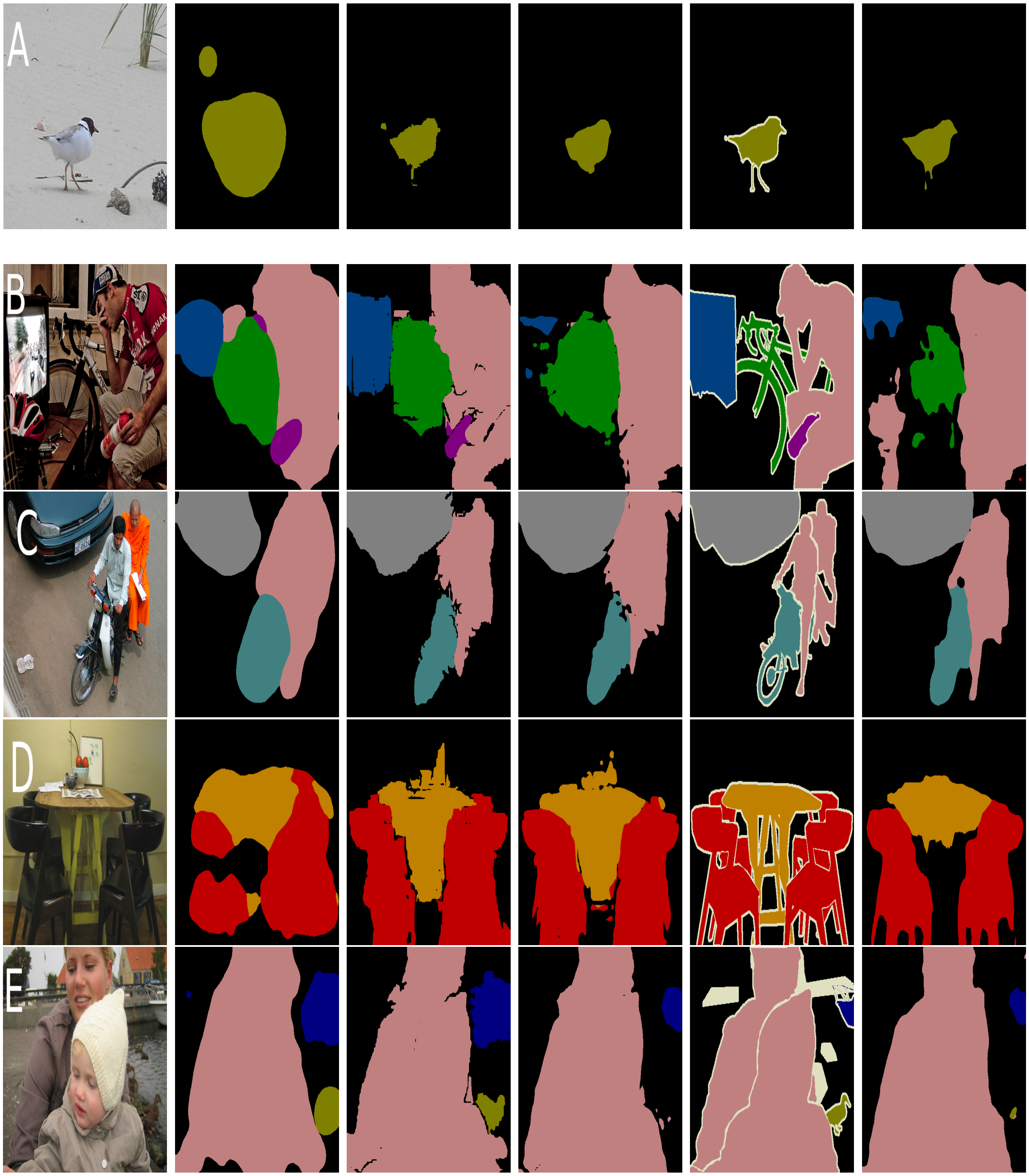

Figure 3.3: From left to right: original images, PCAM labels, refined PCAM labels, predictions from PCAM-supervised DeepLabv3+, ground truth segmentation, and predictions from fully supervised DeepLabv3+ for each image

Method	Sup.	mIOU
WhatsPoint [1]	*P*	46.1
MIL-FCN [45]	*I*	25.7
CCNN [41]	*I*	35.3
EM-Adapt [46]	*I*	38.2
MIL+seg [42]	*I*	42.0
DCSM [47]	*I*	44.1
BFBP [48]	*I*	46.6
SEC [40]	*I*	50.7
AF-SS [49]	*I*	52.6
Combining Cues [36]	*I*	52.8
AE-PSL [24]	*I*	55.0
DSRG [34]	*I*	61.4
AffinityNet [23]	*I*	61.7
IRNet [11]	*I*	63.5
FickleNet [33]	*I*	64.9
ScribbleSup [50]	*S*	63.1
NormCut [51]	*S*	65.1
BBox-seg [46]	*B*	60.6
SDI [52]	*B*	65.7
BCM [53]	*B*	66.8
Ours - refined PCAM	*P*	68.5
Ours - final	*P*	70.5
DeepLabv3+	*F*	78.9

Table 3.3: mIOU comparison of recent or related weakly supervised semantic segmentation methods on the PASCAL VOC 2012 *val* set. Sup. shows the level of supervision of each method where *P* is point level, *I* is image level, *B* is bounding box level, *S* is scribble level, and *F* is fully supervised. Ours - final uses DeepLabv3+ trained on refined PCAMs for inference.

Method	Sup.	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mkb	person	plant	sheep	sofa	train	tv	mean
EM-Adapt [46]	*I*	67.2	29.2	17.6	28.6	22.2	29.6	47.0	44.0	44.2	14.6	35.1	24.9	41.0	34.8	41.6	32.1	24.8	37.4	24.0	38.1	31.6	33.8
CCNN [41]	*I*	68.5	25.5	18.0	25.4	20.2	36.3	46.8	47.1	48.0	15.8	37.9	21.0	44.5	34.5	46.2	40.7	30.4	36.3	22.2	38.8	36.9	35.3
MIL+seg [42]	*I*	79.6	50.2	21.6	40.9	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
SEC [40]	*I*	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
AE-PSL [24]	*I*	83.4	71.1	30.5	72.9	41.6	55.9	63.1	60.2	74.0	18.0	66.5	32.4	71.7	56.3	64.8	52.4	37.4	69.1	31.4	58.9	43.9	55.0
What's Point [1]	*P*	80	49	23	39	41	46	60	61	56	18	38	41	54	42	55	57	32	51	26	55	45	46.0
AffinityNet [23]	*I*	88.2	68.2	**30.6**	81.1	49.6	61.0	77.8	66.1	75.1	29.0	66.0	40.2	80.4	62.0	70.4	**73.7**	42.5	70.7	42.6	68.1	51.6	61.7
BCM [53]	*B*	89.8	**68.3**	27.1	73.7	56.4	72.6	84.2	75.6	79.9	**35.2**	78.3	53.2	77.6	66.4	68.1	73.1	**56.8**	80.1	45.1	**74.7**	54.6	66.8
Ours - final	*P*	**89.9**	66.1	30.1	**85.2**	**62.5**	**75.8**	**87.1**	**80.4**	**87.1**	34.0	**85.1**	**60.0**	**84.4**	**82.4**	**77.4**	68.4	56.1	**84.0**	**46.1**	72.6	**64.2**	**70.5**

Table 3.4: per class mIOU comparsion of recent or related methods on the PASCAL VOC 2012 val set

the performance shown there for ”DeepLabv3+” is for the fully supervised setting trained on actual ground truth as opposed to ”Ours - final” which is DeepLabv3+ trained on refined PCAMs.

3.4.3 PCAM Performance

Figure 3.2 shows some localization performance of PCAMs compared to CAMs generated from the same network without point supervision. This figure shows that the point supervision generally leads to better localization. Further, PCAMs do a much better job of covering more of a given object’s extent.

Table 3.1 shows the quantitative performance difference of each activation map. In addition to the image level CAM, we also experimented with training the network with the PCAM loss given every ground truth point, i.e. full supervision, rather than the much smaller one point per class per image. Point supervision strongly increases the localization performance of activation maps, though using all points in an image with our method seems to have relatively little influence on performance on unseen images.

3.4.4 IRNet Label Refinement

Figure 3.3 shows several examples of our method’s performance as well as the performance of the fully supervised DeepLabv3+. We can see evidence that the label propagation via IRNet’s transition matrix usually helps refine boundaries more precisely, though we tend to see choppy edges as a result of the random walk propagation.

Table 3.2 shows the performance difference of each activation map after having been refined by IRNet. Unsurprisingly, the already better PCAMs have significantly better localization performance after being refined than do CAMs. The performance increase between PCAMs and CAMs only becomes amplified after refinement. Before refinement, PCAMs outperform CAMs by 8.4%, and after refinement, PCAMs surpasses CAMs by 11.1%.

3.4.5 Semantic Segmentation Results

Figure 3.3 shows the performance of DeepLabv3+ after being trained with refined PCAMs as well as its predictions after being trained with ground truth. The results are quite similar, with the PCAM-trained network generally making fewer predictions, which is helpful in the case of the image B where there is no false prediction in the bottom left corner. However, fewer predictions are not helpful in cases such as in the image E where the duck in the bottom right corner is not predicted at all.

While the PCAM-trained CNN makes fewer predictions and seems to sometimes miss smaller objects, it adheres to boundaries better and performs slightly better quantitatively than the refined PCAMs themselves.

Table 3.3 shows the performance of several recent methods on semantic segmentation of the VOC *val* set. Our method achieves state of the art performance in point supervision. Further, it surpasses the method from Tang et al [51], which uses stronger scribble level supervision, and it even outperforms the very recent method by Song et al [53], which uses stronger bounding box supervision for this task. Our method recovers an impressive 89% of the performance of its fully supervised counterpart using only point supervision.

Table 3.4 shows a number of recent methods, their levels of supervision, and their class-wise IOU performance. Unsurprisingly, the better supervised method from Song et al [53] performs better on the some of the challenging classes like chair; however, our method performs better on nearly every class and overall.

Chapter 4

DUSIA Dataset Collection, Curation, and Annotation

4.1 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.

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 [54, 55]. 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

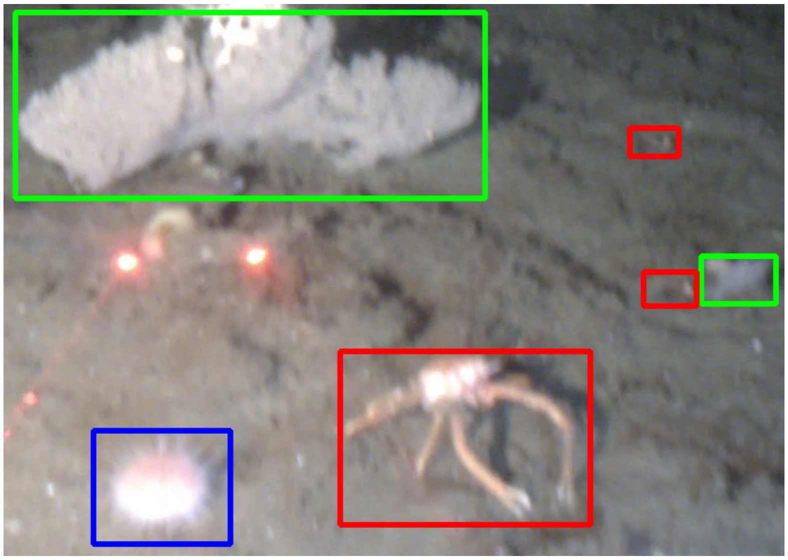

Figure 4.1: Cropped frame from DUSIA with examples of three classes of interest: fragile pink urchin (blue), gray gorgonian (green), and squat lobster (red). Variations in perspective, occlusion, and size can create large differences across appearances of species individuals and make some individuals, like small squat lobsters, almost invisible, especially in a single frame. Crop shown is 690x487 pixels from a 1920x1080 frame.

is vital to determining species hotspots and finding 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.

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. Most computer vision fo-

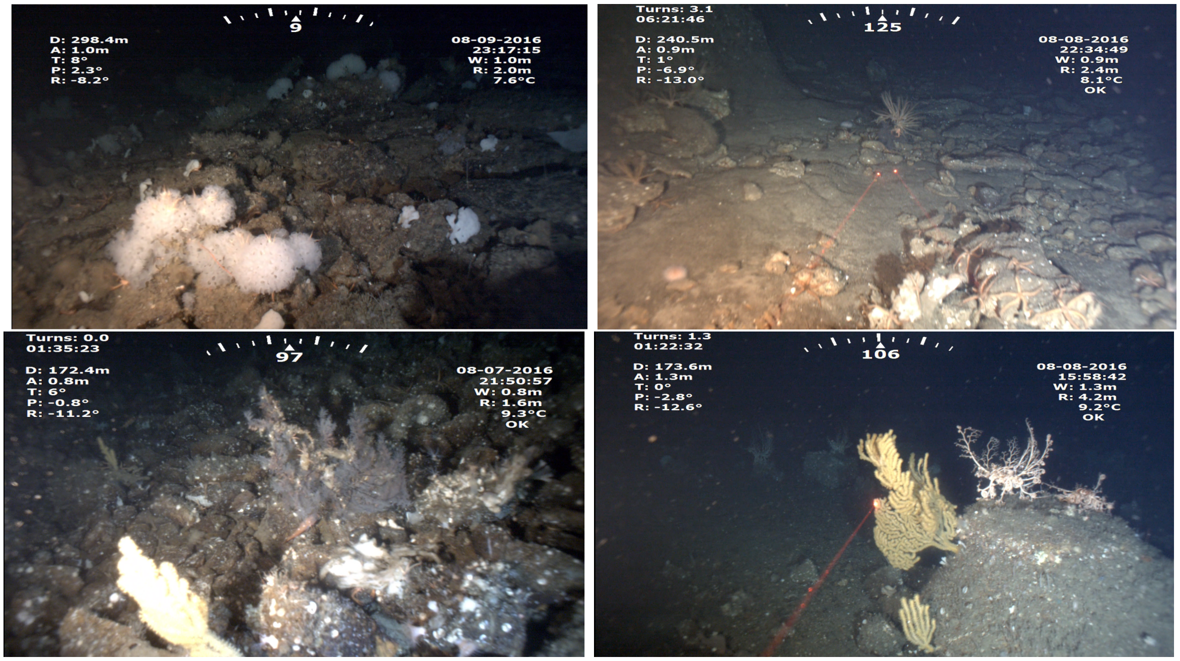

Figure 4.2: Full frame examples pulled from the DUSIA dataset

cused datasets collect the best types of annotations they can for the specific problem they want to solve. For example, to count objects, computer vision scientists might collect point or bounding box annotations. However, domain scientists, like marine researchers, have different problems in mind, and they may find spending their annotations budget on a different type of annotation may be more appropriate for addressing their problems efficiently.

As a step toward advancement in efficiently computationally analyzing videos from a marine science setting, we introduce a Dataset for Underwater Substrate and Invertebrate Analysis, 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.

Some contents of this chapter have been submitted for publication in the International Journal of Computer Vision in *Context-Driven Detection of Invertebrate Species in Deep-Sea Video.*

4.2 Related Works

Current achievements by deep learning-based vision models do not necessarily translate well when it comes to analyzing underwater animals and habitats as there 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 [56–60], it is hardly enough to tackle this daunting problem, 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. For the object detection task, for example, anyone with a mobile device can capture thousands of images and videos. For such a task, a layperson can then annotate different classes of objects in this data. Data-driven machine learning algorithms can leverage these annotations to achieve human-like performance. In contrast, collecting and annotating underwater images and videos requires specialized underwater cameras, equipment for navigating underwater, and experts with specific domain knowledge to then annotate the collected data. As a result, the whole data collection process becomes complicated and expensive. DUSIA aims to be a collaborative, comprehensive effort to guide the exploration and automated analysis of underwater ecosystems.

4.2.1 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 [58,60–63]. Numerous current works [61–64] 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, Maaloy et al [62] 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 [61] augments their dataset of underwater fish images with the underwater non-fish images from VOC2012 [2] by restricting their model to generating only binary (fish vs. no fish) predictions. In the same way, [63, 64] confined their models to do analysis only on one single fish. 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, [65] 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

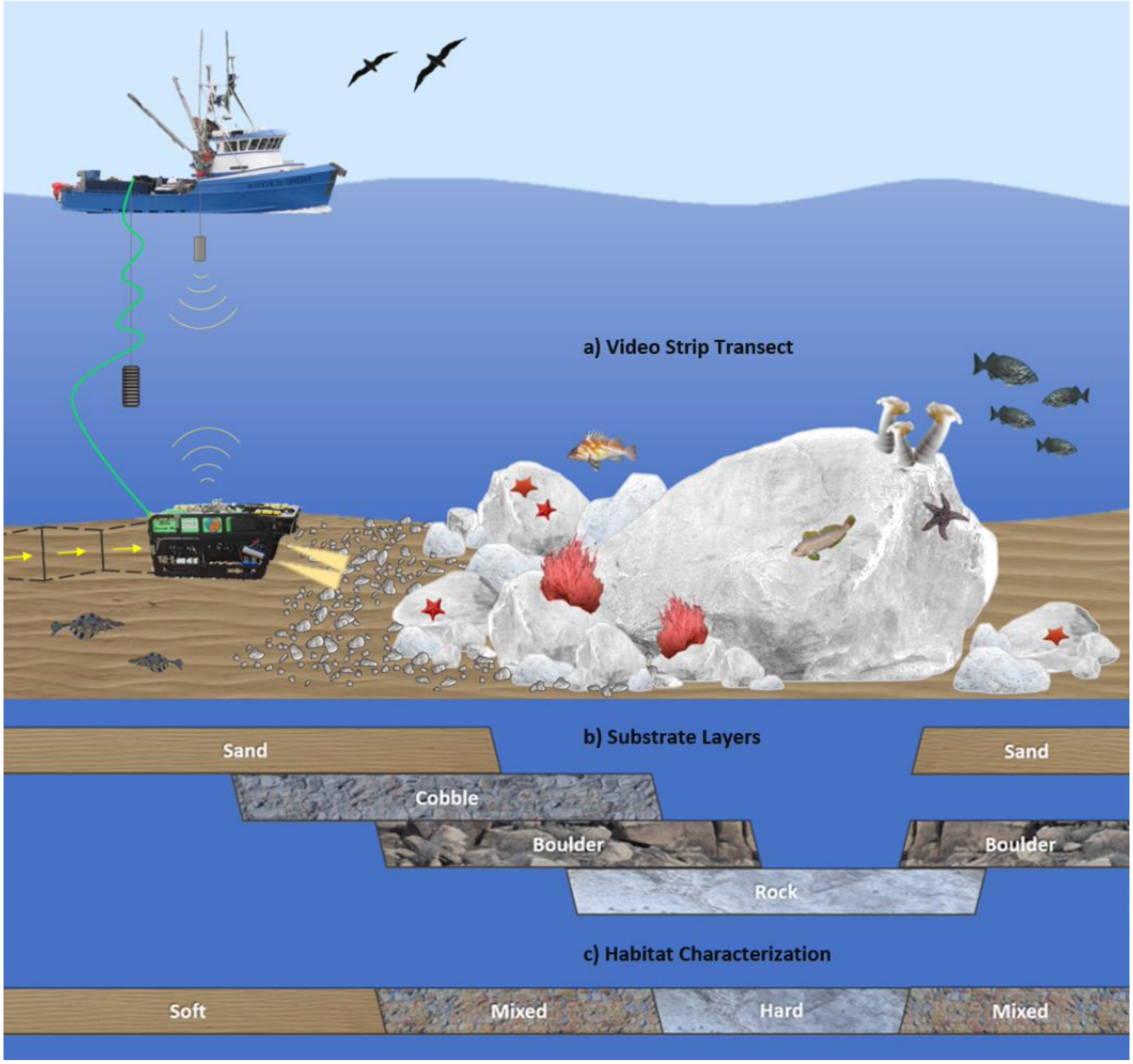

Figure 4.3: Illustration of the ROV attached to the catamaran, substrate layers, and habitat characterization. Substrates are divided into soft (mud, cobble and sand), hard (rock and boulder), or mixed (a combination of any soft and hard substrates). Illustration courtesy of Marine Applied Research and Exploration (MARE) Group.

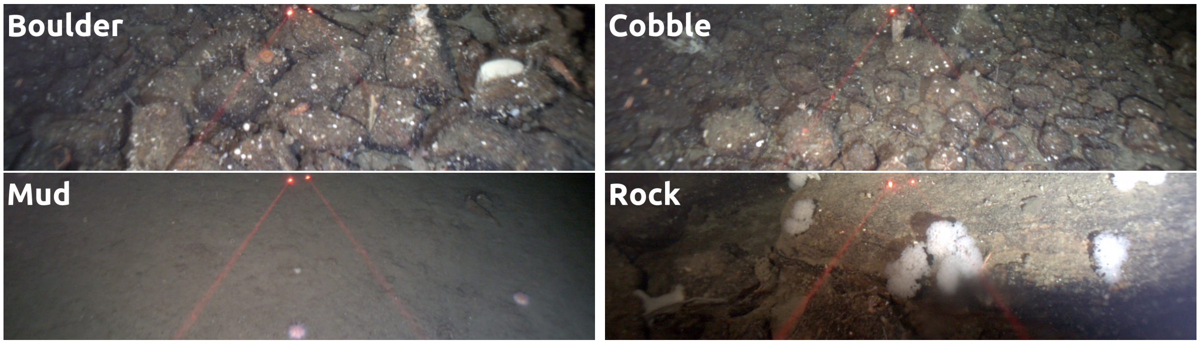

Figure 4.4: Example frames each containing just one substrate each, indicated by the in frame text

in this dataset. DUSIA, in contrast, is unique in this aspect, as it has both explicit substrate and invertebrate annotations.

4.2.2 Methodologies

As mentioned in the previous section, recently, different works have developed deep learning-based algorithms to detect marine species (mostly fishes). Li et al [66] uses a Fast-RCNN [67] based network to classify twelve different species of fish. Salman et al [68] present a deep network to detect fish in 32x32 size video frames. Siddiqui et al [69] 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. The method presented in the next chapter aims to alleviate some of these methods' shortcomings by using explicit substrate predictions to enhance species detections.

4.3 Dataset

DUSIA consists of over 10 hours 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 meters. DUSIA's videos are part of a large collection, and training on them may help annotation of similar videos from different excursions in the future.

4.3.1 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 [54]; 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 process, 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 4.3 roughly illustrates the ROV rig used for data collection.

4.3.2 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 Figure 4.3 and frames from the dataset are shown in Figure 4.4. 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

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 4.1: Description of the four substrates present in DUSIA

Table 4.1, and Table 4.2 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 computer vision methods including some described in the next chapter.

4.3.3 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.

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

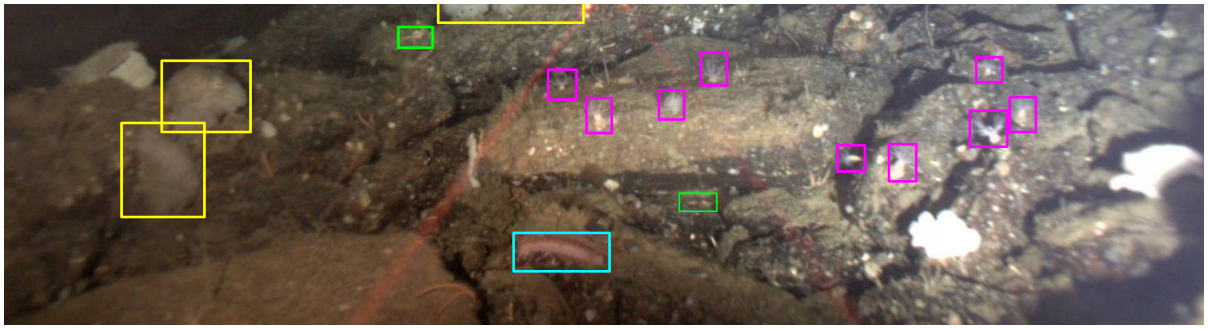

Figure 4.5: Fully annotated frame example. Color to species map is as follows: yellow: laced sponge, magenta: white spine sea cucumber, cyan: white slipper sea cucumber, green: squat lobster.

use these labels to learn to label further videos. However, current computer vision methods struggle with weak supervision, and count labels of this nature are unusual for current machine learning methods.

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	

Table 4.2: 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.

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

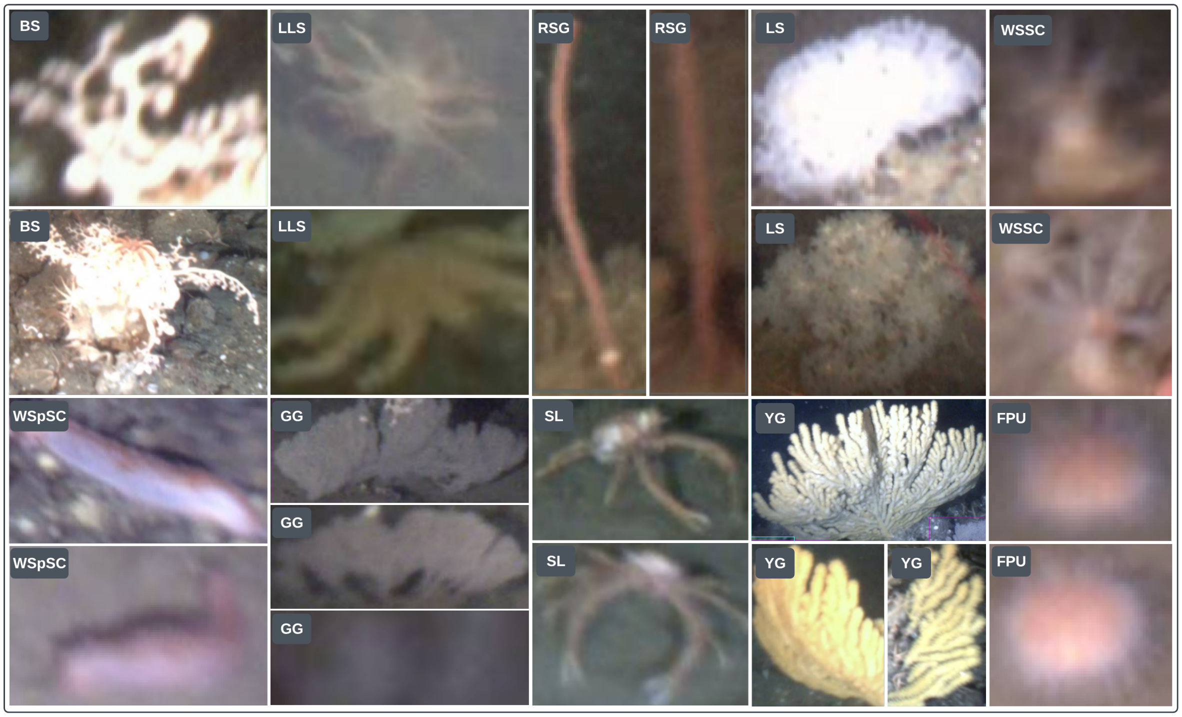

Figure 4.6: Cropped screenshots of each of the ten species of interest: basket star (BS), fragile pink urchin (FPU), gray gorgonian (GG), long-legged sunflower star (LLS), red swifita gorgonian (RSG), squat lobster (SL), laced sponge (LS), white slipper sea cucumber (WSSC), white spine sea cucumber (WSpSC), and yellow gorgonian (YG).

marine scientists' CABOF labels. First, we select a subset of species to annotate with stronger annotations. We choose ten species, each visualized in Figure 4.6 because they are some of the most abundant species in the dataset. Table 4.3 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), are 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

Species	Count	Species	Count
Fragile pink urchin	**3,402**	Spot prawn	18
Squat lobster	**2,593**	UI anemone	17
UI lobed sponge	1,753	Thorny sea star	16
White slipper sea cucumber	**1,313**	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		

Table 4.3: 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.

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 when 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 seconds per frame for trained individuals to annotate. For reference, it took annotators approximately 22.1 seconds per image to fully annotate with single point annotation and 34.9 seconds per image with squiggle supervision in the VOC2012 natural image dataset of 20 classes including cats, busses, and similar common object classes [1]. 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 Figure 4.5. Species shown in that frame without bounding box labels are not species of interest.

4.3.4 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 8,682 keyframes used for training the detector (described in detail in Section 4.3.3). 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 based on CABOF labels before labelling bounding boxes; however, preserving this distribution leads to a slightly uneven distribution of substrate occurrences.

4.3.5 Statistical Analysis of Data

Table 4.4 shows the frequency of each of the substrate classes present in our dataset.

Table 4.5 shows the frequency of bounding box labels for invertebrate species of interest represented in our dataset, and Table 4.6 illustrates the frequency of CABOF

	B	C	M	R	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 4.4: Distribution of number of frames containing each substrate across DUSIA and its splits

labels for invertebrate species.

Table 4.7 illustrates the distributions of CABOF labels for each species across the different substrates. While not weighted against the relative presence of each substrate, this 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.

	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	Total
Train	1,247	3,675	3,294	735	775	3,264	1,071	1,397	819	1,024	17,301
Val	61	394	259	20	85	594	91	439	51	38	2,032
Test	124	653	277	61	79	1,181	98	506	28	180	3,187
Total	1,432	4,722	3,830	816	939	5,039	1,260	2,342	898	1,242	22,520

Table 4.5: Distribution of bounding box annotations of each species across splits. Note that one species individual may be annotated with multiple bounding boxes as it occurs across multiple frames.

4.4 Tasks

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

	BS	FPU	GG	LLS	RSG	SL	LS	WSSC	WSpSC	YG	Total
Train	292	2,828	398	269	190	1,649	517	832	279	103	7,357
Val	17	154	80	8	19	208	40	164	22	9	721
Test	52	420	78	29	48	742	75	317	17	38	1,816
Total	361	3,402	556	306	257	2,599	632	1,313	318	150	9,894

Table 4.6: Distribution of CABOF labels across DUSIA and its splits. As described in Section 4.3.3, each species individual is counted only once when it touches the bottom of the frame.

	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

Table 4.7: Percentage of total species individuals occuring in each substrate according to CABOF labels. Note that a given frame may have multiple substrate labels, so a given individual may occur in multiple substrates at one time.

4.4.1 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 multi-label classification problem. Our dataset makes it possible to develop and test automated methods for this problem.

4.4.2 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 Section 4.3.3.

In the next chapter, we present a method for utilizing explicit context labels to detect, track, and count the ten species of interested that are annotated with bounding boxes in DUSIA's videos.

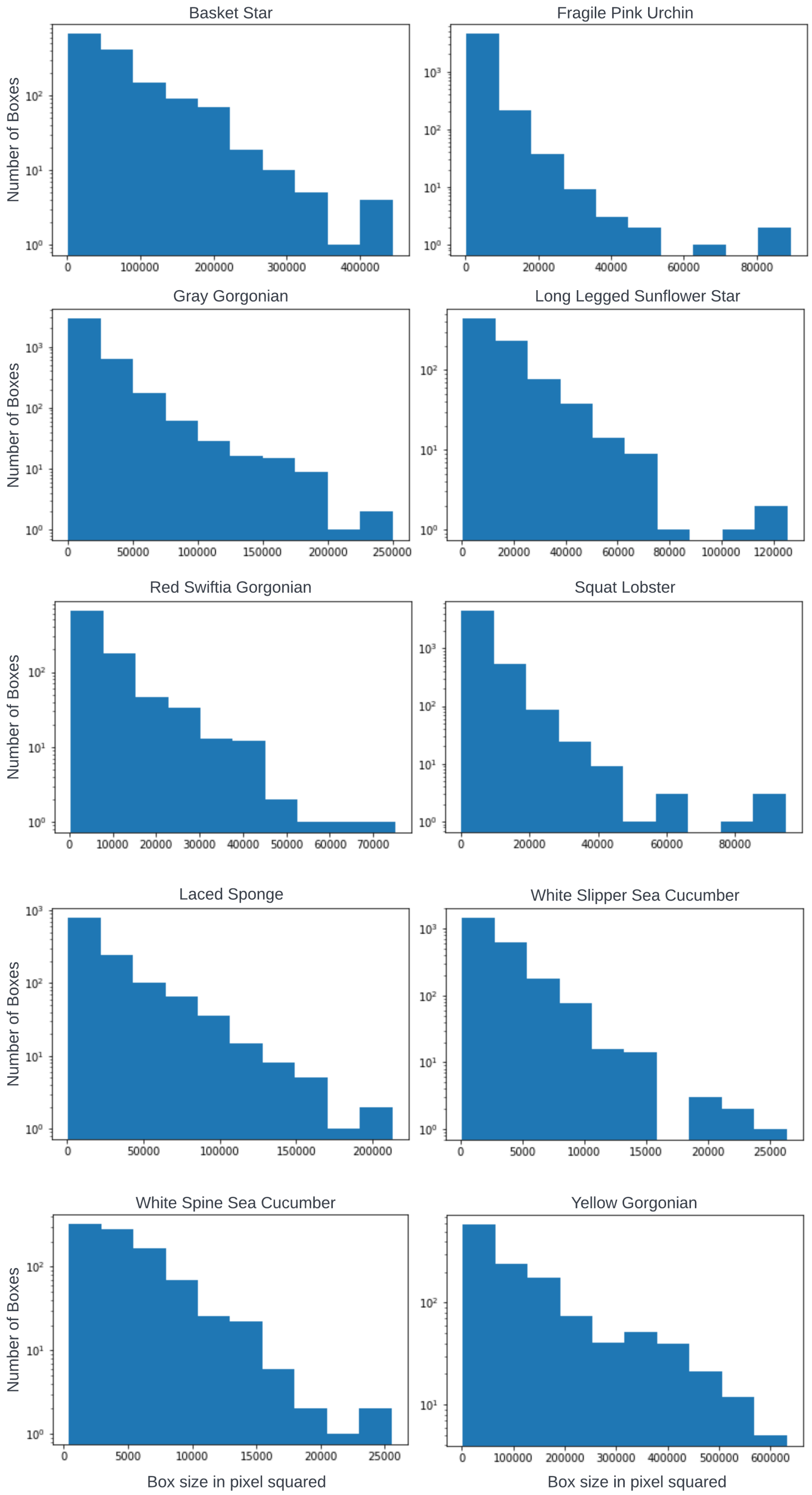

Figure 4.7: Histograms illustrating the distributions of box sizes (in pixels squared) for each species of interest.

Chapter 5

Context-Driven Detection, Tracking, and Counting of Underwater Invertebrate Species

5.1 Introduction

Counting underwater species individuals enables marine scientists and biologists to better understand the health of oceanic regions and to enact practices or propose legislation to ultimately protect life in the oceans, but, in practice, counting these organisms poses a challenging, expensive task. As a means to alleviating the cost of these ocean surveys, we present a method for detecting, i.e. classifying and spatially localizing, the underwater invertebrate species of DUSIA.

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. The context labels present in DUSIA include boulder, cobble, mud, and rock.

Additionally, 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 challenging setting of DUSIA's videos using a detection plus tracking pipeline.

5.2 Related Work

Object detection consists of classifying and localizing instances of classes of interest, typically by drawing a tight bounding box around each instance. Detection positions itself as one of the most popular problems in computer vision with much work being done on this problem across numerous domains, but object detection of commmon object classes (e.g. person, cat, dog) poses perhaps the most popular form of object detection. The vast majority of object detectors optimize for this problem.

State of the art object detection models follow three general architectures: single stage, two stage, and transformer-based models. Single stage models, like YOLO and its derivatives [70–72] make detection predictions in just one pass. Transformer based models like DETR and its derivatives [73, 74] pose object detection as a direcrt set prediction problem and use transformer based encoder-decoders to match box predictions with ground truths. Two stage models like the Faster RCNN [67, 75–77] line of models operate in two passes. A region proposal network does as the name suggests: proposes regions of interest that may contain objects of interest. Then, each of those region proposals are evaluated and refined to generate object detections.

Our novel Context Driven Detector (CDD) builds on Faster RCNN, introducing a

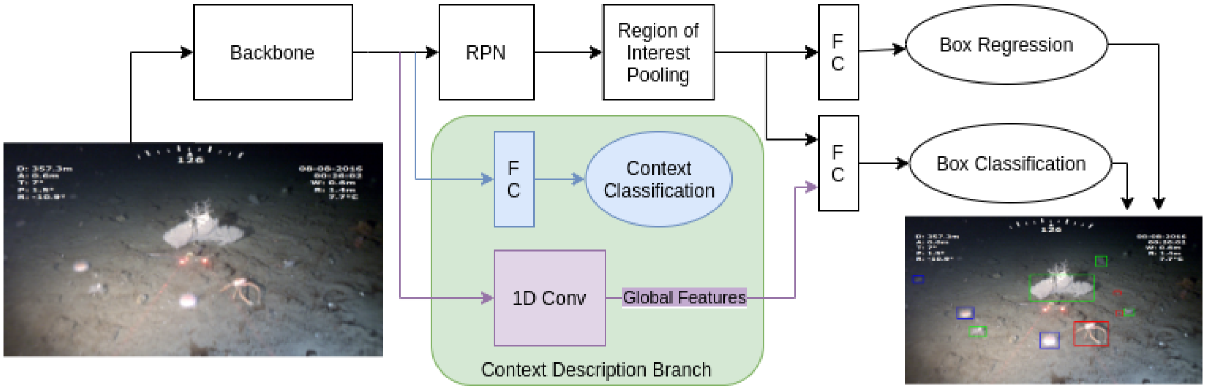

Figure 5.1: Context-Driven Detector: the Context Description Branch (green) takes features from the backbone, classifies context explicitly (blue), and feeds a global representation of context (purple) to the box classification layer to enhance detections. We show that using this branch enhances the detections overall indicating that learning from explicit context labels can enhance detections.

mechanism for using explicit context labels and implicit context representations to enhance object detection performance in underwater video frames where objects of interests are individuals of DUSIA's invertebrate species of interest. Additionally, we introduce Negative Region Dropping, a novel technique for training on DUSIA's partially supervised frames.

5.3 Methods

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 [78] to track invertebrate species and present a simple method for using these tracks to count invertebrate individuals.

5.3.1 Substrate Classification

Before diving into the Context Driven Detector, substrate classification experiments illustrate that deep learning models are capable of recognizing different substrates present in the underwater video. These models and experiments lay a foundation for CDD, suggesting that, if context can be classified with some degree of accuracy, context classifications might also be helpful toward the object detection problem.

For a baseline, we train two basic classifiers for substrate classification. First, we trained an out-of-the-box ResNet-50 based [44] classification CNN, pre-trained on ImageNet [79], 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.

5.3.2 Invertebrate Species Detection

We trained an out-of-the-box Faster RCNN model using our partially annotated keyframes (see section 4.3.3 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 Figure 4.7, many classes in DUSIA are made up of small boxes.

Figure 5.1 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 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 [75]:

$$L_v = L_d + L_p \tag{5.1}$$

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.

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 section 5.4, and show that dropping

			test_wv per class APs				
	val mAP	test mAP	B	C	M	R	test_wv mAP
Separate	**0.588**	**0.646**	**0.274**	**0.802**	0.750	**0.826**	0.663
Combined	0.551	0.572	0.259	0.777	**0.951**	0.781	**0.692**
CDD	0.517	0.596	-	-	-	-	-

Table 5.1: Substrate classifier performance. Per class APs are shown for the test_wv set. CDD shows the classification performance of the CDD with $\alpha = 0.0001$ and $\rho = 0.75$, which was not run on test_wv.

negative proposals in this way leads to significant improvement in detection performance on DUSIA.

Context Driven Detection

To improve invertebrate detection using context annotations, we introduce the novel Context Description Branch as shown in green in Figure 5.1. The first iteration of the context description branch (blue in Figure 5.1) 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 equation 5.1 with the additional loss for explicit context classification.

$$L = L_v + \alpha * L_c \tag{5.2}$$

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 equation 5.2, and the 1D convolution kernel is learned.

5.3.3 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 [78] 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.

5.4 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.

5.4.1 Substrate Temporal Localization

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 5.1 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_wv set which includes many more frames from the test videos. To generate test_wv we simply sample the test videos uniformly at one frame per second. We then classify each frame, and present the AP scores.

Combination of Multiple 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 5.1 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 changing perspective of the ROV makes it difficult to understand scale in the videos. That

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

Table 5.2: Performance of vanilla Faster RCNN with varying learning rates

said, a dedicated boulder detector out-performed the single classifier method overall due to its impressive performance classifying the mud class.

5.4.2 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. For each detection experiment, we initalize our models with weights pretrained on ImageNet and then train the network for up to 15 epochs. 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 [75] with a batch size of 8 and try several learning rates after initializing with weights pre-trained on COCO [7] provided by PyTorch [80]. The results are shown in Table 5.2.

We then perform hyperparameter searches for each of our method contributions described in section 5.3: 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

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 5.3: Performance of Faster-RCNN with varying Negative Region Dropping percentages

of our model since the best model weights are selected on best val mAP.

Negative Region Dropping Percent ρ

Table 5.3 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.

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 correctly, but concatenating a global feature representation with the local box features requires that the features come in at similar scales. Table 5.4 shows the effect of different scalar values for this fusion.

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

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 5.4: Performance of the Context Driven Detector given different β scalar 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 5.5: Performance of the Context Driven Detector given different context loss scaling α values

dedicated context classifier, but it leads to a more powerful object detector. Table 5.5 shows the effects of α on the detection performance.

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 5.6 highlights the best hyperparameter settings revealed during our search, and Section 5.6 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.

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 5.2 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 5.3 shows qualitative examples of success and failure cases of the best version of CDD.

5.4.3 Invertebrate Species Counting

There are some noteworthy differences between the detection and counting problems. As mentioned in Section 4.3.4, we partition DUSIA's videos into three sets: training,

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**

Table 5.6: Performance of best models from each hyperparameter combination

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 5.7: Relative errors of our counting method with no thresholding and the best threshold settings. Darker color indicates better performance. See Table 4.6 for ground truth counts for each species.

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

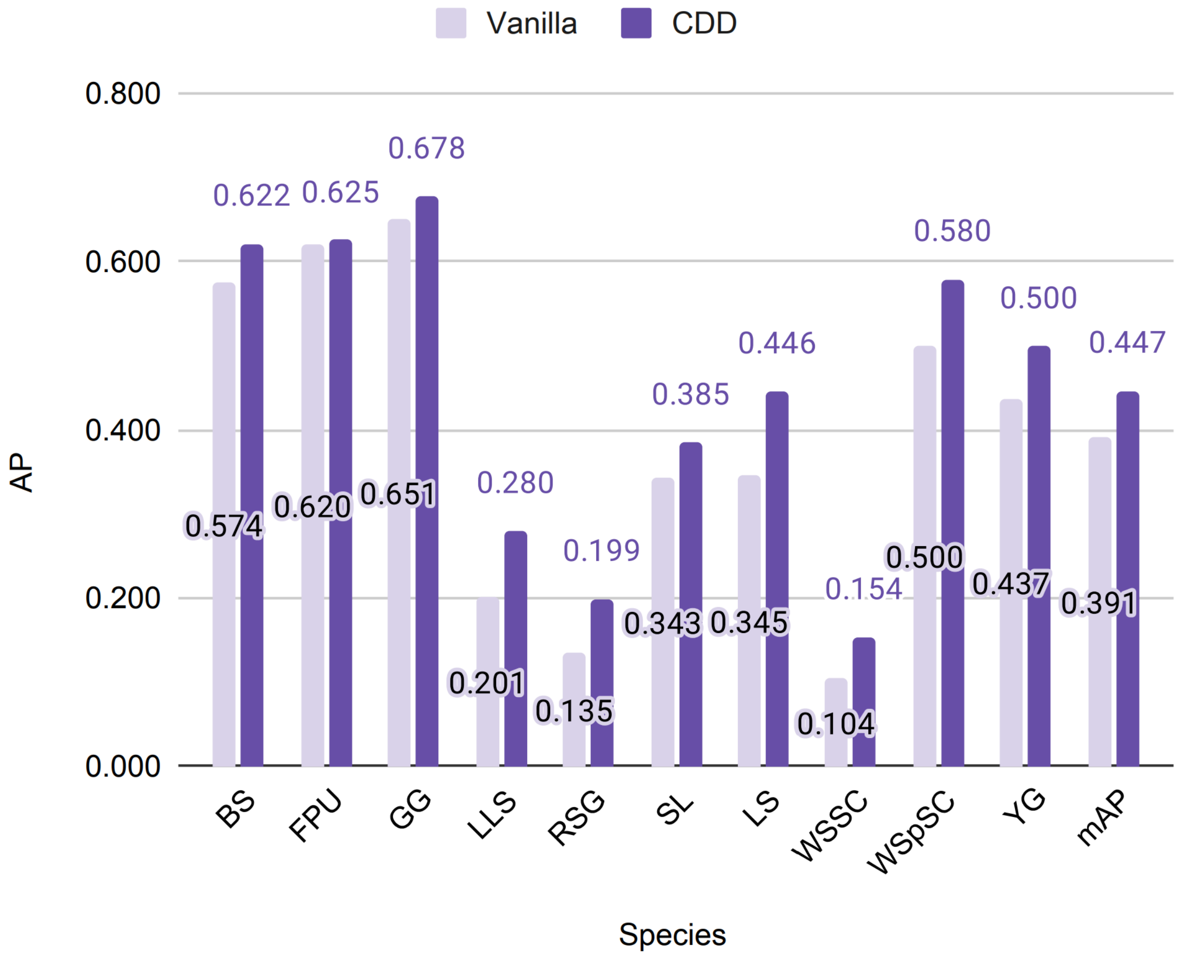

Figure 5.2: Per class test AP comparison of vanilla Faster RCNN and the best Context Driven Detector

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'

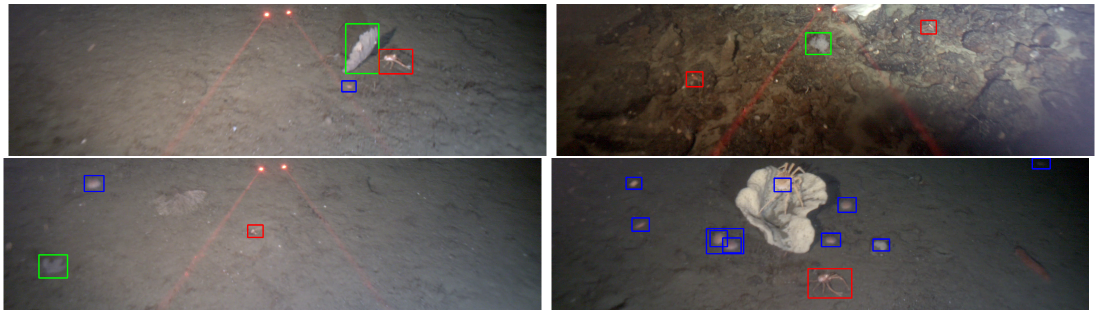

Figure 5.3: Detection examples from our dataset. Blue indicates fragile pink urchin; green, gray gorgonian; and red, squat lobster. We show the success of our detector with the exception of the bottom right image. A crab (not a species of interest) is mislabeled as a fragile pink urchin toward the top 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.

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 [78]. 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 5.3 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.

Table 5.7 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. Section 5.6 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.

5.5 Discussion and Future Work

Our baseline methods' detection and counting performance leaves plenty of room for improvement. Our detection methods 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.

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.

5.6 Hyperparameter Search Summary

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

Table 5.8: Results of hyperparameter search experiments on learning rate, α, β, and ρ

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 5.9: 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.

					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

Table 5.10: Relative errors of our counting method with different settings across the test set's videos

Chapter 6

Generating Context Matched Collages for Object Detection on Underwater Video

While we have achieved significant improvements in object detection performance on DUSIA's video frames via Context Driven Detection, the overall performance is far from perfect. Compared to many of today's benchmark datasets for object detection on natural images, DUSIA has a relatively small number of training frames, so we aim to show that increasing the size of the training set can further enhance object detection performance. In order to do so, we introduce a method for generating more training samples for DUSIA: Context-Matched Collages.

6.1 Introduction

Today's computer vision methods largely depend on enormous datasets with many annotated examples for each class. These sorts of datasets can be extremely expensive to collect, especially when the data is more scientific in nature. While any layperson can label a cat, dog, or human, the cost of labelling and differentiating between more specific, scientific classes grow exponentially. The cost of collecting, annotating, and analyzing scientific data is high, but that also means that any automation or streamlining of those processes can greatly benefit domain scientists who currently rely almost entirely on expensive human experts.

The Dataset for Underwater Substrate and Invertebrate Analysis (DUSIA) [81] provides an example of a challenging, scientific dataset. DUSIA contains 10 hours of video collected in 1080p using a remotely operated vehicle (ROV) that drives over and records the ocean floor at depths between 100 m and 400 m, and the data within DUSIA is part of a much greater, growing collection of hundreds of hours of unlabelled or weakly labelled videos. Marine scientists collect these videos as part of surveys that improve their understanding of habitats and organisms of the ocean floor. DUSIA's videos come directly from marine scientists working to study, understand, and survey the ocean floor.

Despite the rich content of the videos, DUSIA's annotations are limited due to the expense of hiring trained marine science experts to annotate video with the level of granularity of typical computer vision datasets. The dataset provides numerous, weak labels, which indicate timestamps at which 57 invertebrate species of interest are Counted At the Bottom Of the video Frame (CABOF), as well as a training set with frames partially annotated with bounding boxes for the the ten species shown in Figure 6.1.

CABOF labels are described in detail in the original work [81] and illustrated via a frame by frame representation of DUSIA's videos in Figure 6.2. In summary, as the ROV traverses the ocean floor, species come into view at the top of the frame and make their way to the bottom of the frame as the ROV and video moves forward. Cropped example frames are shown in Figure 6.2 with frames going forward in time from bottom to top. When a species individual first touches the bottom of the frame (like the yellow

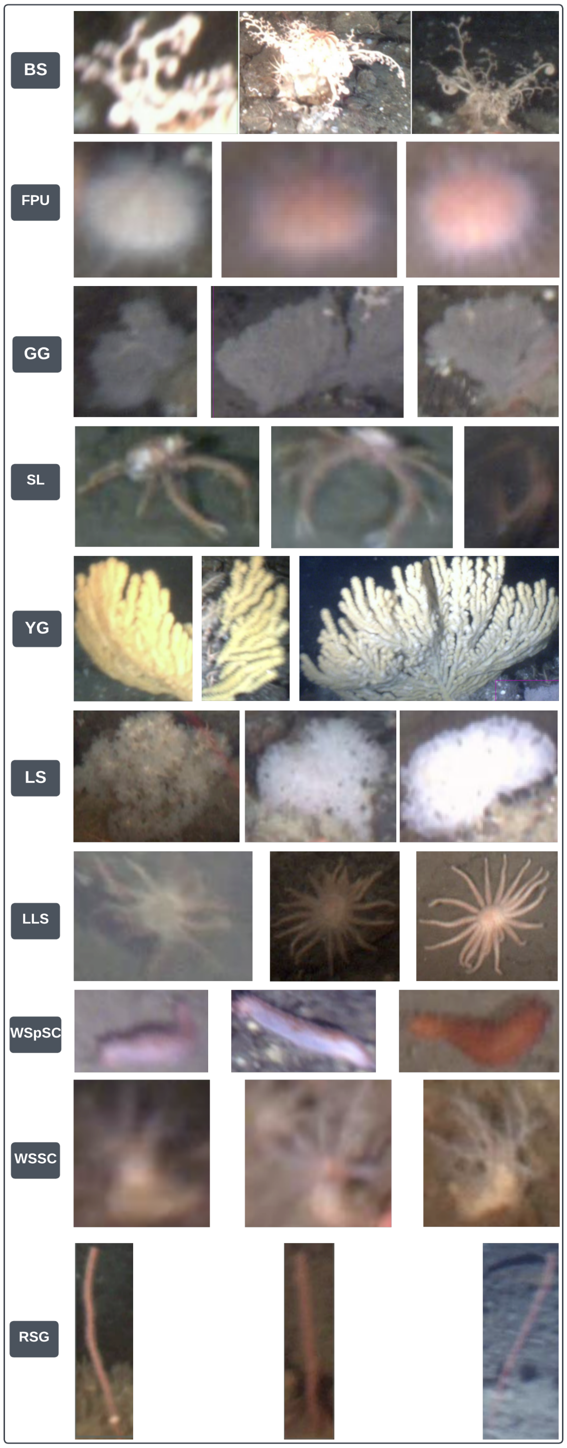

Figure 6.1: Examples of the ten species that are labelled with bounding boxes in DUSIA. YG stands for yellow gorgonian; BS, basket star; GG, gray gorgonian; LS, laced sponge; WSpSC, white spine sea cucumber; LLS, long-legged sunflower star; SL, squat lobster; FPU, fragile pink urchin; WSSC, white sea slipper cucumber; RSG, red swifita gorgonian.

gorgonian in Frame F of Figure 6.2), annotators create a CABOF label with that species name and the timestamp, which corresponds to collected GPS coordinates. This labelling gives marine science researchers a metric for counting the number of species individuals occurring along a narrow transect path. In Section 6.3, we present a new use for these CABOF labels and leverage them to try to find frames in DUSIA's videos where there are *no* species.

DUSIA presents partial bounding box labels for training because collecting full labels is preventatively expensive. These labels are partial in that every instance of a species of interest in the training set's frames may not be annotated. That is, there may be some unlabelled individuals of species of interest in the training frames.

DUSIA's partial annotations provide an interesting challenge for today's computer vision methods and require new methods to solve the object detection problems presented by the dataset. While unconventional, using computer vision on challenging, scientific datasets opens up new possibilities for computer vision applications. One new possibility may include generating synthetic data to supplement and enhance small, noisy training sets.

Our contributions are as follows:

- We present a novel variation on conventional "cut & paste" methods. Our method leverages explicit context labels available in DUSIA to generate new training samples that combine existing training samples with relatively empty background frames available in sparsely populated video areas, illustrating that cutting bounding boxes from the training set (as opposed to cutting more precise, segmented class instances) can serve as an effective basis for data augmentation.

- We introduce a computationally inexpensive method for leveraging DUSIA's CABOF labels, and by extension, empty, context-only, background images for generating ef-

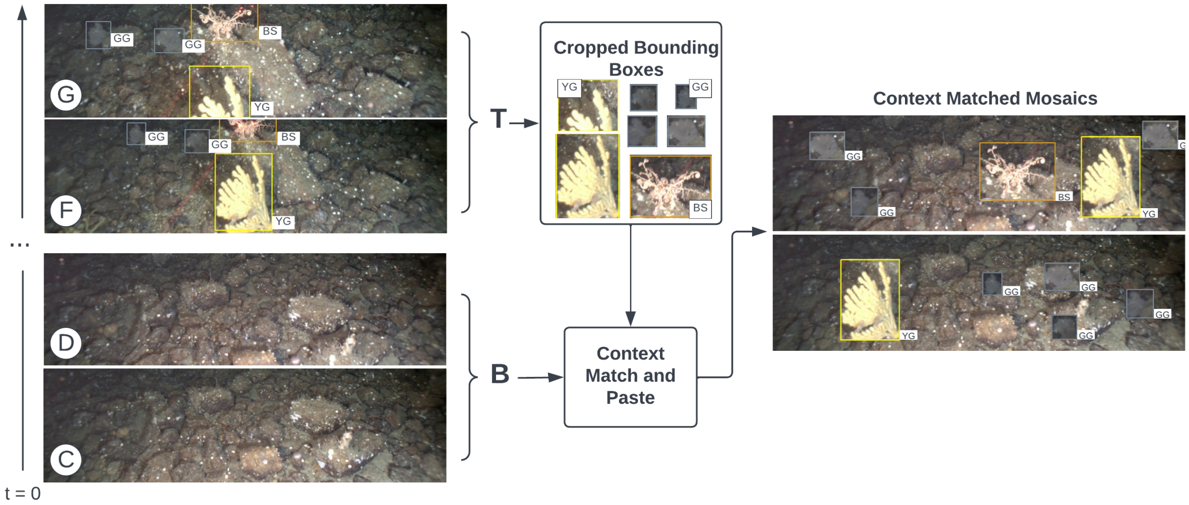

Figure 6.2: Diagram illustrating the method for generating Context Matched Collages. Mine bounding boxes from training set **T**, background frames from **B**, match the context, and paste boxes on to a context matched frame from **B**. See Figure 6.1 caption for species name abbreviations.

fective training samples.

- Using our method, we achieve state of the art detection results on DUSIA's validation and test sets using multiple different popular object detection models.

6.2 Related Work

Computer vision researchers have long been aware of the power of data augmentation methods for improving training object detectors and image classifiers, and recently, much work has gone into generating plausible training samples via cut/paste methods. Cut/paste methods take objects of interest, cut them from their original image, and paste them into another training image or other type of canvas (e.g. a blank background). DeVrires et al. [82] introduce Cutout as a method of cutting portions from images during training to help improve performance in the image classification task, and CutMix [83] builds upon Cutout by combining two training samples at a time by cropping a random part of one image and pasting it on to another image.

Cut, Paste, and Learn [84] leverages separate collections of common images of objects and typical indoor scene examples to cut object instances from the images available for training and paste them to random background scene images. Cutting object instances from images relies on an image segmentation model to separate objects from their backgrounds, and then those instances are randomly pasted on random indoor images.

Ghiasi et al. [85] perform an augmentation similar to Cut/Paste, and Learn where they cut object instances from one image to paste on to a different, randomly selected image. Interestingly, they argue that the context, on to which their cut instances are pasted, does not matter. However, they work with the COCO dataset which contains simple objects and does not have explicit context labels. Ghiasi et al. [85] consider indoor vs outdoor images, which they label based on COCO's panoptic labels. They then use these augmented images to train an image segmentation model. In the medical domain, TumorCP [86] leverages image segmentation labels to create additional training samples for a segmentation network, leading to better segmentation performance.

ObjectMix leverages instance segmentation labels to augment data for action recognition in videos by extracting object segments from two videos and combining them to create new video samples [87]. Similarly, Continuous Copy-Paste works to leverage instance segmentation labels to generate training samples for training models to solve the Multi-object Tracking problem [88].

Dvornik et al. show the importance of context in cut paste methods [89]. Their method involves training a network using both bounding box and instance segmentation labels to generate a notion of context where the bounding box includes pixels that are not included by the segmentation label. They train a network to then predict possible paste locations for cut out object instances so that objects are pasted on to images that their model predicts to be sensible.

Our method also employs a cut-paste based method and illustrates the importance of context in our scenario but in a few key different ways. For one, our method does not rely on any expensive segmentation labels, which label every pixel in an image, or segmentation models, which may segment objects unreliably. Section 6.4 shows that directly cutting a whole bounding box labelled as an object of interest from one frame and pasting it to another frame with matching context enables an improvement in object detection performance. This is important because segmentation labels are difficult and expensive to collect in many scenarios with challenging, scientific data like DUSIA [81], on which we present our results.

Additionally, we leverage explicit context labels for our background images. By following the method described in Section 6.3 and pasting boxes on to context-matched background frames, we maintain context clues that are relevant to detecting and determining the species of an underwater invertebrate.

Finally, we filter background frames from DUSIA's underwater video. We use the provided CABOF labels for the underwater invertebrates to find background images in underwater video. We leverage the vast number of video frames in DUSIA that are not labelled as containing invertebrate species and use those as empty background frames for our newly generated training samples.

6.3 Generating Context Matched Collages

Our method for generating synthetic frames from existing ones is a simple but powerful extension of typical cut-paste methods. We introduce novel changes to this method that allow us to generate better training samples for our specific DUSIA dataset. DUSIA contains 10 hours of video captured in 1080p at 30 fps, but across all of that video footage

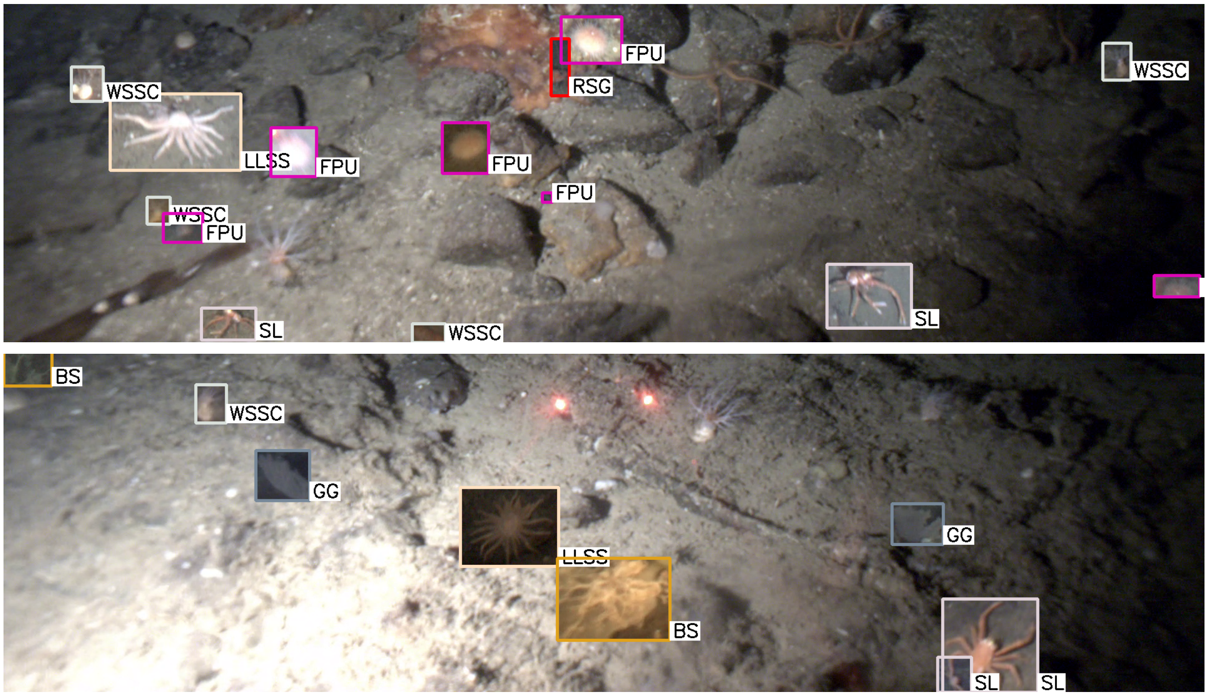

Figure 6.3: Examples of generated collage images. Top: bounding boxes from images labelled Cobble/Mud pasted on to an empty, background image with the matched Cobble/Mud label. Bottom: images and background labelled Mud. See caption of figure 6.1 for species abbreviations.

only 8,682 partially annotated frames contain bounding box labels suitable for supervising most of today's object detectors. These frames are considered partially annotated because they may contain individuals of species of interest that are not labelled with bounding boxes.

Still, DUSIA provides CABOF labels for the entirety of its videos. These CABOF labels indicate the first time at which a group or individual of a species of interest intersects with the bottom of a video frame (like the yellow gorgonian shown in Frame F of Figure 6.2) such that there is a single CABOF label for every single individual of a species of interest that touches the bottom of a video frame in DUSIA's videos. Our method aims to leverage these CABOF labels in a unique way.

Unfortunately, a training set of 8,682 frames limits the performance of the object detectors, but collecting additional bounding box annotations is expensive, especially given the challenging nature of DUSIA and its object classes. DUSIA's partial annotation scheme alleviates the annotation burden on expensive expert annotators, but the scheme makes it very difficult to train an effective object detection network. McEver et al. [81] demonstrate that Negative Region Dropping (NRD) can help train Faster RCNN based models on the partially annotated dataset, but the detector that they present is far from perfect. We propose combining DUSIA's original training set with Context Matched Collages as a complementary method that enables even better detection performance.

The first step in generating these collages is to find a set of frames from the videos that can serve as background for the synthetic collage training samples. Ideally, these background frames contain a minimal number of species individuals so that the resulting collages can have few unlabelled species individuals in them. Pasting on to these sort of background frames allows the object detector to see more of the video, helping it generalize on the test set. Further, by pasting known objects on to empty frames, we can generate frames that are better supervised than some of the partially annotated training set, as they contain fewer unlabelled species of interest.

To this end, we generate a set of frames, $\mathbf{B}$, that are unlikely to contain species of interest. In order to do so, we leverage the Count at Bottom of Frame (CABOF) labels, which indicate timestamps containing species individuals, provided for DUSIA's videos. We can therefore use frames that are far from all CABOF labels to find all the time spans that are unlikely to contain a species of interest.

We first initialize $\mathbf{B}$ to contain all frames in the training set. Then, we iterate through all CABOF labels removing frames within a certain range (e.g. a few seconds) of any CABOF label time stamp. For example, if a CABOF label indicates that there are three fragile pink urchins at time 00:10:30, we can remove all frames ranging from 00:10:20 to 00:10:40 from $\mathbf{B}$. Because not all species individuals that appear in the video touch the bottom of the video frame, they do not all get a CABOF label, but many individuals do. Additionally, since many species typically occur together, this step helps filter out very busy parts of the video with lots of species of interest in them, regardless of whether they touch the bottom of the frame because it is likely that their neighboring species individuals touch the bottom of the frame if they do not. The remaining frames in $\mathbf{B}$ may contain a few unlabelled species of interest, but there should be far fewer unlabelled species than in the original training frames, which are known to be partially supervised and typically come from busy parts of the videos that contain many intermingling groups of species of interest.

Given that nearly all frames in DUSIA have substrate labels that indicate the substrate present on the ocean floor, we also create a map $\mathbf{S}$ that maps each substrate combination, s to the frames from $\mathbf{B}$ that contain that substrate combination. Having this mapping allows us to match the context (i.e. substrate label) of potential background frames and the context of any bounding boxes we wish to paste into a new collage frame.

In order to generate the final collages, we cut bounding boxes from the original training set and paste them on to images from $\mathbf{B}$ that have matching substrate labels. To do so, we map all substrate combinations to a list of bounding boxes that exist on frames with that substrate combination label. For each substrate combination, we randomly sample boxes, and paste them to random locations in a randomly selected image from $\mathbf{B}$ that has matching context labels. In our case we randomly select between 1 and 15 boxes to paste on each image. We also ensure that boxes do not fully occlude one another, though we do allow partial overlap because species individuals often cluster closely together in the original videos.

While the generated collage images may be obviously manipulated to a human eye, they help train a stronger object detection model by providing better supervision, unique co-occurrences of species individuals, and more samples. We explore these improvements in Section 6.4

Algorithm 1 Pseudo code for generating Context Matched Collages

```
$b \leftarrow$ 10 seconds                                         ▷ buffer
$B \leftarrow$ all training frames
$C \leftarrow$ all CABOF labels
for $c \in C$ do                                                  ▷ Create set of potential background frames
    $T \leftarrow c.timestamp$
    $B.remove([T - b, T + b])$
end for
$L \leftarrow$ map of each substrate label to empty list
for $f \in B$ do                                                  ▷ map substrate combos to bg frames
    $L[f.substrate].append(f.timestamp)$
end for
$K \leftarrow$ map of each substrate label to bounding box labels
$M \leftarrow \emptyset$                                          ▷ list of generated frames
while not done do
                                                                  ▷ k is substrate label, O is list of boxes on substrate k
    for $k, O \in K.items()$ do
        $l \leftarrow L[k]$                                       ▷ list of bg w/ label k
        $m \leftarrow random.choice(l)$                           ▷ m can serve as bg
        $r \leftarrow random.randint([1, MAX\_BOXES])$
        for $i \in [0, r)$ do
            $o \leftarrow random.choice(O)$
            $O.remove(o)$
            $m.paste(o)$                                          ▷ paste box o randomly on bg
        end for
        $M.append(m)$                                             ▷ add Context Matched Collage
        if $len(M)$ ¿ MIN then
            $done \leftarrow True$
            break
        end if
    end for
end while
```

Figure 6.3 shows example Context Matched Collages generated by our method.

6.4 Experiments

We train multiple model architectures by combining DUSIA's 8,682 training frames with collages generated via our method. We refer to those 8,682 training frames as **T**. We paste a maximum of 15 boxes onto each background image. We use a buffer of 10 seconds around each CABOF label to ensure that our background images are sufficiently empty.

In order to illustrate the importance of context matching, we generate two collage sets. **M** contains approximately 2,000 frames generated as described in Section 6.3 with contexts matched properly. In practice, there are a small number of bounding box labels with substrate combinations not present in **B**. For those frames, we simply map to the nearest substrate combination. For example, if a box's substrate label is both mud and cobble, we paste it to a background frame that is either mud or cobble if there are no frames in **B** with the exact same substrate label of mud and cobble.

The second collage set, **R**, contains approximately 2,000 frames generated in exactly the same way as **M** except the context of the background frame and the pasted object frame are *not necessarily* matched. That is, the background frame is randomly selected from all of **B** rather than randomly selected from a subset of **B** with the matched substrate combination.

When training Faster R-CNN with negative region dropping, we saw success in first training on $\mathbf{T} + \mathbf{M}$ then lowering the learning rate to finetune on **T**. We also present results on that setting indicated by $\mathbf{T} + \mathbf{M}, \mathbf{T}$.

First, we tested the Context-Driven Detector (CDD) as proposed by McEver et al. [81]; however, we found that the detector performed better without the context description branch when trained with our collage augmented training sets. We theorize this may be due to the imperfect matching of context in some frames. Still, we found negative region dropping to help overall performance, and we set the negative region dropping percentage, ρ to 0.75 as in the original paper, and, also following [81], we used a learning rate of 0.01 for Faster R-CNN with Negative Region Dropping.

We also trained YOLOv5 [72] on each of our training sets to demonstrate the impact of including our Context Matched Collages. We trained the YOLOv5 large model from the author's provided weights, which were pre-trained on the COCO dataset [4]. After testing a variety of different hyperparameter settings, we found the best performance with most of the default settings; however, we changed the initial learning rate, lr_0, from 0.001 to 0.0001; the final OneCycleLR [90] learning rate, lr_f, from 0.1 to 0.01; the anchor-multiple threshold [91], $anchor_t$, from 4.0 to 2.0; the image rotation, $degrees$, from 0.0 to 30.0; and the image perspective [91], $perspective$, from 0.0 to 0.001. After training using the above settings, we finetuned on $\mathbf{T}$ lowering lr_0 to 1e-6.

Finally, we tested the DEtection TRansformer (DETR) [73] to give an additional example of a state of the art object detection framework. Starting from the author's provided pretrained weights, we trained DETR with minor changes to the default settings. We used ResNet-50 as the backbone and changed learning rate to 1e-5 and the learning rate drop to 40 epochs.

Table 6.1 shows the experimental results of all three detectors trained on the different training sets without any collages, including the Context Matched Collages, and the collages with random backgrounds. We evaluate our detection results using mean Average Precision (mAP) with an intersection over union (IOU) threshold of 0.5 as presented in [81]. We choose this metric, widely known as AP_{50}, because the detection tasks in DUSIA is not sensitive to exact localization as the detection tasks ultimately aim to aid in counting of invertebrate species. For more in depth analysis, we also present the popular full COCO suite [92] of evaluation metrics for our best detection model, Faster R-CNN with Negative Region Dropping trained on $\mathbf{T} + \mathbf{M}, \mathbf{T}$ in Table 6.2.

For our best model we also present some qualitative detection results in Figure 6.4.

Detector	**Train Set**	**val mAP**	**test mAP**
Context Driven Detector [81]	T	0.524	0.447
DETR [73]	T	0.534	0.416
DETR	T+R	0.541	0.426
DETR	T+M	0.541	0.446
YOLOv5 [72]	T	0.558	0.452
YOLOv5	T+R	0.518	0.437
YOLOv5	T+M	**0.558**	0.470
Faster RCNN [75] w/ NRD [81]	T	0.509	0.439
Faster RCNN w/ NRD	T+R	0.511	0.419
Faster RCNN w/ NRD	T+M	0.542	0.453
Faster RCNN w/ NRD	T+M, T	0.546	**0.482**

Table 6.1: Different object detectors, and their detection performance given different training sets. CDD results from [81]

metric	val	test
$AP_{50:95}$	0.264	0.221
AP_{50}	0.553	0.482
AP_{75}	0.224	0.174
AP_S	0.017	0.016
AP_M	0.168	0.165
AP_L	0.335	0.286

Table 6.2: Full COCO suite of metrics showing the performance of the best Faster R-CNN with NRD model on both the val and test sets

The results for this busy frame show that the detector performs quite well at finding objects at the edge of the frame, and it does a good job discriminating among species. It even leaves out a few sponge species that are not species of interest. The detector still struggles with the red swifitia gorgonian (RSG) class, which is one of the most challenging in DUSIA, showing some area for improvement.

For all three detectors, we see better generalizability in the model, as evidenced by test mAP, when the model is trained with Context Matched Collages. We also see that training with Context Matched Collages (**T**+**M**) consistently achieves better performance

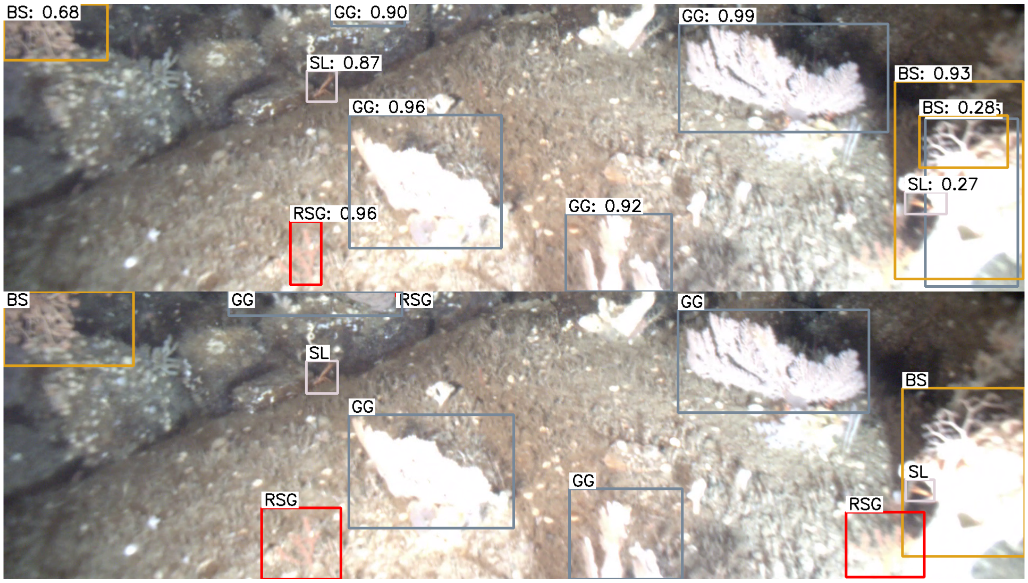

Figure 6.4: Detections from best model (top) and ground truth (bottom) for an example frame. See figure 6.1 caption for species name abbreviations.

than training with the the collages without context matching ($\mathbf{T} + \mathbf{R}$). YOLOv5 and Faster RCNN even see decreased performance when trained with collages in $\mathbf{R}$ illustrating the importance of context when detecting invertebrate species. Faster R-CNN achieves state of the art performance on the test set when trained with Context Matched Collages after finetuning on the original training set. Clearly, augmenting DUSIA's training set with Context Matched Collages leads to better overall performance.

6.5 Discussion

We introduced Context Matched Collages in this chapter. We mine frames containing few species of interest, cut bounding boxes from our training set, and paste those bounding boxes on to the mined images. This process leverages many unused video frames and produces unique training samples that aid in training object detectors to increase performance. We illustrate that cutting bounding boxes, as opposed to finely segmented object instances, and pasting them to create new training samples provides an effective augmentation for object detectors. Further, we introduce matching the explicit context labels of bounding boxes and the background to create collages. By augmenting the original training set of DUSIA with these Context Matched Collages, we are able to achieve state of the art object detection performance.

Even with these improvements, detection on DUSIA remains a challenging task, and the detection performance still needs improvement to alleviate the manual detection and counting of invertebrate species. The low performance on small objects, shown as AP_S in Table 6.2 shows reveals an area for improvement, and the qualitative results in Figure 6.4 show that measures may be taken to improve the performance on the red swiftia gorgonian class and perhaps other specific classes.

Chapter 7

Discussion

This dissertation aims to explore and provide methods for using less supervision, via weak and partial labels, to address common computer visions tasks in both natural and scientific settings. As image and video data and associated research becomes more abundant and more advanced, annotation of this data becomes more expensive. Creating methods that can automate annotation of data with less supervision can reduce the costs of all kinds of studies and accelerate research in computer vision and any field using machine learning to advance their studies.

PCAMs, introduced in Chapter 3 can enable semantic segmentation with only point level labels as supervision, and the method proposed in Chapter 3 can recover almost 90% of the performance of its fully supervised counterpart, showing that it may not be entirely necessary to collect pixel level labels for every project.

Chapter 5 shows that it may also be possible to achieve acceptable object detection performance given a partially labelled training set. Negative Region Dropping can help two-stage detection models perform better given a partially annotated training set.

The Context-Driven Detector from Chapter 5 also demonstrates how other types of labels, that is explicit context labels, may be important for object detection as well.

There may be cases where collecting these labels has already been done or is relatively cheap, and it may be worth incorporating them into training to enhance performance of an object detection system.

Chapter 6 provides a method for cheaply generating additional training samples for DUSIA. Training with those additional samples increases object detection performance on DUSIA, and we hope that this method can inspire similar approaches for other datasets as well.

7.1 Broader Impacts

The methods presented in this dissertation primarily focus on natural images and a specific type of underwater video. While applications for natural videos range widely among social media, self driving systems, in home robotics, and beyond, the broader impacts of work for specific types of underwater video may not be so obvious; however, when digging deeper, it is clear that the contributions to this domain can extend far beyond.

DUSIA provides a set of videos from a specific expedition to a specific place in the ocean, but these videos employ a generic method used by many marine science groups to conduct their surveys. As such, the methods produced for DUSIA may prove effective for expeditions using similar methods in other areas, and the models trained on DUSIA may even prove effective on videos from other areas. The substrate classifiers in particular may be effective in a wide range of underwater video because cobble, mud, rock, and boulder cover vast areas of the ocean floors.

Further, many of the invertebrate species found in DUSIA's videos are found in other areas of the ocean. As mentioned in Chapter 4, fragile pink urchins, squat lobsters, and other species annotated in DUSIA are spread widely across the ocean floor, so methods for detecting them may be applicable to future expeditions without retraining.

Along the lines of widely applicable methods, Negative Region Dropping may be an effective method for training on any datasets with unintentional noise. Few datasets may be intentionally partially annotated like DUSIA, but many datasets include noisy annotations with lots of annotations missing. Still, present annotations are generally quite reliable. For that reason, Negative Region Dropping may aid in object detection across a wide variety of datasets.

Similarly, it may be uncommon for video datasets to include to explicit labels for background; however, we have shown that collecting these labels may be worthwhile or that simply using the implicit context representation of CDD's context description branch improve detection performance in DUSIA and may be impactful in other domains where extra attention to context is important.

Extra attention to context may also aid in the generation of Context Matched Collages for a wide variety of datasets. Chapter 6 illustrates a method for generating collages using explicit context labels, but context may be matched in other ways as well. Extracting a context feature and employing some distance measure may allow clustering of images to generate context groups that may make collage generation possible without explicit labels.

7.2 Future Directions

The work presented in this dissertation may be extended in many ways. DUSIA may be expanded with videos from other expeditions, and more labels in DUSIA may lead to large improvements toward automating the tasks that interest marine science researchers.

Negative Region Dropping and CDD's context description branch may have wide uses beyond just underwater video, and exploring those may be worthwhile. Negative region dropping may have profound impacts on improving detection performance of noisily annotated data, and the context description branch may lead to improvements in object detection performance where context is particularly important.

The out of the box tracking method used to count invertebrates in Chapter 5 also leaves room for improvement. Extracting features from surrounding frames for each frame of interest may help improve the model's detection performance and may enable tracking in fewer stages. Further, the BYTE tracking model used may also benefit from training on image features rather than simply making predictions based on box locations over a series of frames.

Context matched collages may also lead to improvements in other datasets. While few datasets have explicit context labels and a plethora of background frames, generating pseudo context labels or creating background frames in other ways may be worth exploring.

Bibliography

[1] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, *What's the point: Semantic segmentation with point supervision*, in *European conference on computer vision*, pp. 549–565, Springer, 2016.

[2] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, *The pascal visual object classes challenge: A retrospective*, *International journal of computer vision* **111** (2015), no. 1 98–136.

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, *The pascal visual object classes (voc) challenge*, *International journal of computer vision* **88** (2010), no. 2 303–338.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft coco: Common objects in context*, in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 740–755, Springer International Publishing, 2014.

[5] A. Krizhevsky, G. Hinton, *et. al.*, *Learning multiple layers of features from tiny images*, .

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, *International Journal of Computer Vision (IJCV)* **115** (2015), no. 3 211–252.

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

[8] A. Rosebrock, *Intersection over union (iou) for object detection*, Apr, 2022.

[9] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, *Semantic contours from inverse detectors*, in *2011 International Conference on Computer Vision*, pp. 991–998, IEEE, 2011.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, *The cityscapes dataset for semantic urban scene understanding*, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] J. Ahn, S. Cho, and S. Kwak, *Weakly supervised learning of instance segmentation with inter-pixel relations*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2209–2218, 2019.

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-decoder with atrous separable convolution for semantic image segmentation*, in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.

[13] R. A. McEver and B. Manjunath, *Pcams: Weakly supervised semantic segmentation using point supervision*, *arXiv preprint arXiv:2007.05615* (2020).

[14] L. Itti, C. Koch, and E. Niebur, *A model of saliency-based visual attention for rapid scene analysis*, *IEEE Transactions on Pattern Analysis & Machine Intelligence* (1998), no. 11 1254–1259.

[15] X. Hou and L. Zhang, *Saliency detection: A spectral residual approach*, in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Ieee, 2007.

[16] B. Alexe, T. Deselaers, and V. Ferrari, *Measuring the objectness of image windows*, *IEEE transactions on pattern analysis and machine intelligence* **34** (2012), no. 11 2189–2202.

[17] H. Cholakkal, J. Johnson, and D. Rajan, *Backtracking scspm image classifier for weakly supervised top-down saliency*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5278–5287, 2016.

[18] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani, *Self-taught object localization with deep networks*, in *2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–9, IEEE, 2016.

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, *Is object localization for free?-weakly-supervised learning with convolutional neural networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 685–694, 2015.

[20] R. G. Cinbis, J. Verbeek, and C. Schmid, *Weakly supervised object localization with multi-fold multiple instance learning*, *IEEE transactions on pattern analysis and machine intelligence* **39** (2016), no. 1 189–203.

[21] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, *Learning and transferring mid-level image representations using convolutional neural networks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, *Learning deep features for discriminative localization*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[23] J. Ahn and S. Kwak, *Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4981–4990, 2018.

[24] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, *Object region mining with adversarial erasing: A simple classification to semantic segmentation approach*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1568–1576, 2017.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, *Grad-cam: Visual explanations from deep networks via gradient-based localization*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

[26] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, *Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks*, in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, IEEE, 2018.

[27] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, *Adversarial complementary learning for weakly supervised object localization*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1325–1334, 2018.

[28] R. Briq, M. Moeller, and J. Gall, *Convolutional simplex projection network for weakly supervised semantic segmentation.*, in *BMVC*, p. 263, 2018.

[29] X. Wang, S. You, X. Li, and H. Ma, *Weakly-supervised semantic segmentation by iteratively mining common object features*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1354–1362, 2018.

[30] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, *Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7268–7277, 2018.

[31] W. Ge, S. Yang, and Y. Yu, *Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1277–1286, 2018.

[32] S. Hong, D. Yeo, S. Kwak, H. Lee, and B. Han, *Weakly supervised semantic segmentation using web-crawled videos*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7322–7330, 2017.

[33] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, *Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5267–5276, 2019.

[34] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, *Weakly-supervised semantic segmentation network with deep seeded region growing*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7014–7023, 2018.

[35] T. Durand, T. Mordan, N. Thome, and M. Cord, *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, 2017.

[36] A. Roy and S. Todorovic, *Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3529–3538, 2017.

[37] S. Kwak, S. Hong, and B. Han, *Weakly supervised semantic segmentation using superpixel pooling network*, in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[38] Z. Lu, Z. Fu, T. Xiang, P. Han, L. Wang, and X. Gao, *Learning from weak and noisy labels for semantic segmentation*, *IEEE transactions on pattern analysis and machine intelligence* **39** (2016), no. 3 486–500.

[39] X. Liang, Y. Wei, L. Lin, Y. Chen, X. Shen, J. Yang, and S. Yan, *Learning to segment human by watching youtube*, *IEEE transactions on pattern analysis and machine intelligence* **39** (2016), no. 7 1462–1468.

[40] A. Kolesnikov and C. H. Lampert, *Seed, expand and constrain: Three principles for weakly-supervised image segmentation*, in *European Conference on Computer Vision*, pp. 695–711, Springer, 2016.

[41] D. Pathak, P. Krahenbuhl, and T. Darrell, *Constrained convolutional neural networks for weakly supervised segmentation*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1796–1804, 2015.

[42] P. O. Pinheiro and R. Collobert, *From image-level to pixel-level labeling with convolutional networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1713–1721, 2015.

[43] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, *Deep extreme cut: From extreme points to object segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 616–625, 2018.

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

[45] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, *Fully convolutional multi-class multiple instance learning*, *arXiv preprint arXiv:1412.7144* (2014).

[46] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, *Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1742–1750, 2015.

[47] W. Shimoda and K. Yanai, *Distinct class-specific saliency maps for weakly supervised semantic segmentation*, in *European Conference on Computer Vision*, pp. 218–234, Springer, 2016.

[48] F. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez, *Built-in foreground/background prior for weakly-supervised semantic segmentation*, in *European Conference on Computer Vision*, pp. 413–432, Springer, 2016.

[49] X. Qi, Z. Liu, J. Shi, H. Zhao, and J. Jia, *Augmented feedback in semantic segmentation under image level supervision*, in *European Conference on Computer Vision*, pp. 90–105, Springer, 2016.

[50] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, *Scribblesup: Scribble-supervised convolutional networks for semantic segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3159–3167, 2016.

[51] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, *Normalized cut loss for weakly-supervised cnn segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1818–1827, 2018.

[52] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, *Simple does it: Weakly supervised instance and semantic segmentation*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 876–885, 2017.

[53] C. Song, Y. Huang, W. Ouyang, and L. Wang, *Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3136–3145, 2019.

[54] G. Shester, B. Enticknap, E. Kincaid, A. Lauermann, and D. Rosen, *Exploring the living seafloor: Southern california expedition*, *Oceana Report* (2017).

[55] P. Drap, J. Seinturier, B. Hijazi, D. Merad, J.-M. Boi, B. Chemisky, E. Seguin, and L. Long, *The rov 3d project: Deep-sea underwater survey using photogrammetry: Applications for underwater archaeology*, *Journal on Computing and Cultural Heritage (JOCCH)* **8** (2015), no. 4 1–24.

[56] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, *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, 2019.

[57] A. King, S. M. Bhandarkar, and B. M. Hopkinson, *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, 2018.

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

[59] S. Marini, E. Fanelli, V. Sbragaglia, E. Azzurro, J. D. R. Fernandez, and J. Aguzzi, *Tracking fish abundance by underwater image recognition*, *Scientific reports* **8** (2018), no. 1 1–12.

[60] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, R. Planque, A. Rauber, R. Fisher, and H. Müller, *Lifeclef 2014: multimedia life species identification challenges*, in *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 229–249, Springer, 2014.

[61] D. A. Konovalov, A. Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves, *Underwater fish detection with weak multi-domain supervision*, in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

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

[63] D. Levy, Y. Belfer, E. Osherov, E. Bigal, A. P. Scheinin, H. Nativ, D. Tchernov, and T. Treibitz, *Automated analysis of marine video with limited data*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1385–1393, 2018.

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

[65] A. R. Rashid and A. Chennu, *A trillion coral reef colors: Deeply annotated underwater hyperspectral images for automated classification and habitat mapping*, *Data* **5** (2020), no. 1 19.

[66] X. Li, M. Shang, H. Qin, and L. Chen, *Fast accurate fish detection and recognition of underwater images with fast r-cnn*, in *OCEANS 2015-MTS/IEEE Washington*, pp. 1–5, IEEE, 2015.

[67] R. Girshick, *Fast r-cnn*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[68] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, *Fish species classification in unconstrained underwater environments based on deep learning*, *Limnology and Oceanography: Methods* **14** (2016), no. 9 570–585.

[69] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, and E. S. Harvey, *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** (2018), no. 1 374–389.

[70] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, *arXiv preprint arXiv:1804.02767* (2018).

[71] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, *arXiv preprint arXiv:2004.10934* (2020).

[72] G. Jocher, A. Stoken, J. Borovec, A. Chaurasia, and L. Changyu, *ultralytics/yolov5*, *Github Repository, YOLOv5* (2020).

[73] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[74] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, *Deformable detr: Deformable transformers for end-to-end object detection*, *arXiv preprint arXiv:2010.04159* (2020).

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

[76] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask r-cnn*, in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[77] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[78] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, *Bytetrack: Multi-object tracking by associating every detection box*, *arXiv preprint arXiv:2110.06864* (2021).

[79] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[80] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et. al.*, *Pytorch: An imperative style, high-performance deep learning library*, *Advances in neural information processing systems* **32** (2019) 8026–8037.

[81] R. A. McEver, B. Zhang, C. Levenson, A. Iftekhar, and B. Manjunath, *Context-driven detection of invertebrate species in deep-sea video*, *arXiv preprint arXiv:2206.00718* (2022).

[82] T. DeVries and G. W. Taylor, *Improved regularization of convolutional neural networks with cutout*, *arXiv preprint arXiv:1708.04552* (2017).

[83] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. J. Yoo, *Cutmix: Regularization strategy to train strong classifiers with localizable features*, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019) 6022–6031.

[84] D. Dwibedi, I. Misra, and M. Hebert, *Cut, paste and learn: Surprisingly easy synthesis for instance detection*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1301–1310, 2017.

[85] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, *Simple copy-paste is a strong data augmentation method for instance segmentation*, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021.

[86] J. Yang, Y. Zhang, Y. Liang, Y. Zhang, L. He, and Z. He, *Tumorcp: A simple but effective object-level data augmentation for tumor segmentation*, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 579–588, Springer, 2021.

[87] J. Kimata, T. Nitta, and T. Tamaki, *Objectmix: Data augmentation by copy-pasting objects in videos for action recognition*, *arXiv preprint arXiv:2204.00239* (2022).

[88] Z. Xu, A. Meng, Z. Shi, W. Yang, Z. Chen, and L. Huang, *Continuous copy-paste for one-stage multi-object tracking and segmentation*, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15303–15312, 2021.

[89] N. Dvornik, J. Mairal, and C. Schmid, *Modeling visual context is key to augmenting object detection datasets*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 364–380, 2018.

[90] L. N. Smith and N. Topin, *Super-convergence: Very fast training of neural networks using large learning rates*, in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006, pp. 369–386, SPIE, 2019.

[91] G. Jocher, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements." `https://github.com/ultralytics/yolov5`, Oct., 2020.

[92] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, *Object detection with deep learning: A review*, *IEEE transactions on neural networks and learning systems* **30** (2019), no. 11 3212–3232.