Using Mask R-CNN for Underwater Fish Instance Segmentation as Novel Objects: A Proof of Concept

I-Hao Chen*[1] and Nabil Belbachir[1]

[1]NORCE Norwegian Research Centre, Bergen, Norway

Abstract

Instance Segmentation in general deals with detecting, segmenting and classifying individual instances of objects in an image. Underwater instance segmentation methods often involve aquatic animals like fish as the things to be detected. In order to train deep learning models for instance segmentation in an underwater environment, rigorous human annotation in form of instance segmentation masks with labels is usually required, since the aquatic environment poses challenges due to dynamic background patterns and optical distortions. However, annotating instance segmentation masks on images is especially time- and cost-intensive compared to classification tasks. Here we show an unsupervised instance learning and segmentation approach that introduces a novel class, e.g., "fish" to a pre-trained Mask R-CNN model using its own detection and segmentation capabilities in underwater images. Our results demonstrate a robust detection and segmentation of underwater fish in aquaculture without the need for human annotations. This proof of concept shows that there is room for novel objects within trained instance segmentation models in the paradigm of supervised learning.

1 Introduction

Underwater imaging can be challenging, due to a number of factors that affect taken images. Ranging from light reflection over water particles to contrast lost and light attenuation [10], labelling data may also be difficult for humans. At the same time, focusing on pure image instances and pan-optic segmentation usually requires an exhausting amount of high-quality labelled data. There is only a handful of available data sets which are frequently cited which suggests that the investment in labour and time to create data sets, for instance, semantic and pan-optic segmentation is high [2, 3, 5, 13]. Since supervised learning methods depend on pre-labelled ground truth data to learn, the objective of introducing a new novel class is not traditionally approachable without the further investment of resources. We try to alleviate that problem by introducing a method to fine-tune an existing Mask R-CNN model with its inference output to detect a novel object in the same environment.

In summary, we contribute an unsupervised system with minimal parameter settings that allows a Mask R-CNN to detect a novel class, i.e. fish.

2 Related Work

2.1 Instance Segmentation

Historically, the Region-based Convolutional Network method (RCNN) [7] was proposed to achieve an introduction to the concept of regions and therefore locality to object detection. The first step from image classification to object detection has been made using convolutional layers as the backbone. Fast R-CNN [6] alleviated some of the drawbacks of the RCNN, mainly the time component. The training of the hyperparameters of the region proposal component was trainable in the next extension, Faster RCNN [15]. Changing the k-means algorithm behind the region proposals to a fully connected network (FCN) and the convolutional neural network (CNN), as well as weight information sharing between the Deep FCN module and the Fast R-CNN, led to a decrease of manual hyperparameters and a better time performance of around factor 10. Up until that point, only bounding boxes could be assigned to image data. By the introduction of the library Detectron [9] implementing the Mask R-CNN,

*Corresponding Author: chen@norceresearch.no

https://doi.org/10.7557/18.6791

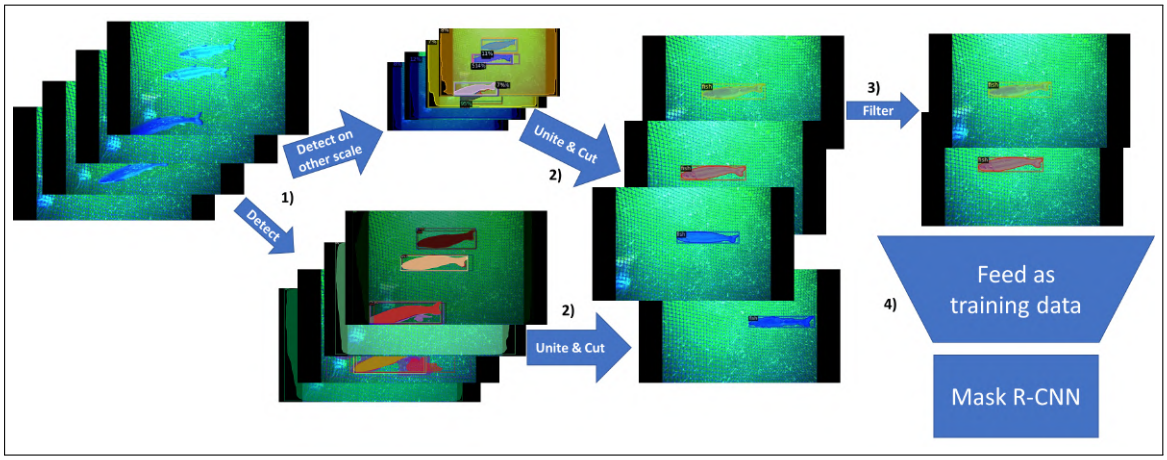


Figure 1: The proposed method: A four-step pipeline. Step 1: Extracting instance segmentation masks from baseline model. Step 2: Unification of the inference masks. Step 3: Filtering unified masks. Step 4: Train baseline model with instance segmentation masks with label *fish*.

masks are also generated. With a more concise model to hold onto local structure information through Region of Interest-Pooling (RoI-Pooling), the instance segmentation masks are calculated parallel with the classification. The Detectron2 [17] was introduced afterwards as a PyTorch-based modular object detection library as the successor of Detectron. There are also other approaches to the marine environment like instance segmentation in acoustic backscatter data to identify pelagic species [14].

2.2 Precision Fish Farming

The analogous principle of precision livestock farming in aquaculture has been defined with some adaptations as precision fish farming [4]. The concept divides the operation of precision aquaculture into sections: Observe, Interpret, Decide, and Act. While decision-making in aquaculture is often machine-aided due to the rise of artificial intelligence [8], observational challenges like estimating biomass and lice tracking remain challenging [12, 16]. Accurate mask measurements for fish contain more details about the object's shape compared to fish detection through bounding boxes [1] and can give rise to more complex analysis.

3 Material and Methods

3.1 Data Collection

The data collection took place at the Austevoll Research Station (Institute of Marine Research, Bergen, Norway) on 15.06.2021. Atlantic salmon smolts (20 cm/ 80 grams) in the sea cage (5x5m) were filmed using the ARV-i (Transmark Subsea, Bergen), an underwater drone with a navigation camera. We took a total of 3 videos with the navigation camera. The videos were recorded with a resolution of 6156x4509 pixels, respectively and varied in length ranging from 160 frames to 637 (368$\pm$199). All frames were being used in the proposed method. The recordings were taken at a depth of 2-3.5m.

3.2 The Mask R-CNN model

A baseline of COCO Instance Segmentation with Mask R-CNN from the library Detectron2 [17] was used, i.e. the model named R50-FPN with the model id 137849600. The model had been trained on COCO `train2017` and evaluated on COCO `val2017` [13] and uses a residual neural network (ResNet) and feature pyramid network (FPN) backbone. If not denoted with other attributes, we mean with Mask R-CNN always the pre-trained R50-FPN model. We note that COCO `train2017` and COCO `val2017` [13] contain no class "fish".

3.3 Proposed Method

The algorithmic pipeline for unsupervised training on instance segmentation is sketched in Figure 1 and consists of four main steps:

1. Extracting instance segmentation results from baseline model on images with novel objects
2. Unify extracted instance segmentation masks
3. Filter the unified masks
4. Train baseline model with unified masks as labelled with class "fish"

3.3.1 Extracting Masks

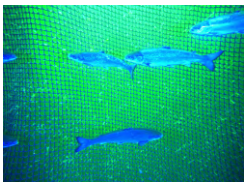
Mask R-CNN inference gives us three kinds of results: Bounding box, segmentation and classification. The output of the classification module was ignored. Bounding box predictions made by the Fast R-CNN module of Mask R-CNN are kept if their confidence score is above 5%. We call this parameter "bounding box confidence threshold". Lower-scoring bounding boxes were discarded. Theoretically, we could also choose lower (or even 0%) values as bounding box confidence threshold and thus allow each bounding box prediction to be kept, but this would increase the inference time significantly. Figure 2 and Figure 3 illustrate the difference in lowering the bounding box confidence threshold from 5% to 0.05% which lead foremost to many not-usable predictions, but also more instance segmentation of the novel object. Since this step (see step **1)** in Figure 1) just needs to produce at least one mask per fish individual for the proposed method to work, we chose 5% as the bounding box confidence threshold.

In order to increase the number of detection masks, we resized copies of the input images by scale factor $1:10$ before running Mask R-CNN inference on them as well. The resized images had thereby a resolution of 615x450 pixels.

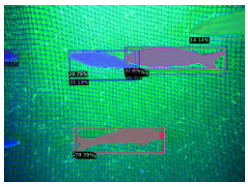
3.3.2 Unify Instance Segmentation Masks

We ran similarity checks on each bounding box (and its respective segmentation mask) from step 1 by iteration:

1. Collect all bounding boxes that have at least an Intersection over Union (IoU) of 80% with the iterating bounding box.



(a) original image



(b) segmentation output

Figure 2: Original image (a) and output of Mask R-CNN at 5% bounding box confidence threshold (b). The coloured boxes are bounding box predictions by the Fast R-CNN module of the Mask R-CNN, the coloured areas inside each bounding box illustrate the instance segmentation masks, respectively.

2. Use the instance segmentation masks of the kept group of bounding boxes to calculate an IoU.
3. If the IoU value of the instance segmentation masks is higher than 50%, apply unification of masks. Otherwise, continue with the next bounding box.
4. The final mask consists of pixel positions covered in at least 55% of the instance segmentation masks. For more insight into the choice of this hyperparameter, see Section 3.5.
5. Add the final mask to the set of final masks if not already present (no duplicates).

The pseudo-code is provided in Figure 4.

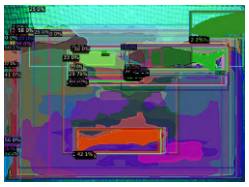
By selecting only the pixels which are present in at least 55% of segmentations, this voting strategy suppresses outliers. See Section 3.5 for further exploring the threshold number.

3.3.3 Filtering the Results

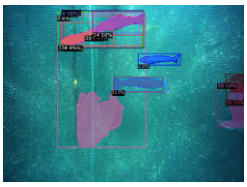
We filter novel object instance segmentation masks by parameters such as extent, solidity, equivalent diameter, mean value and aspect ratio which require human knowledge about the novel object to tune. At the same time, the quality of the resulting training data benefits from filtering.

3.3.4 Training the Network

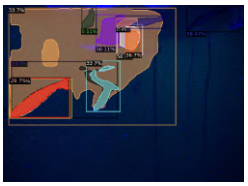
In the end, we fine-tune the Mask R-CNN model with the extracted training data, starting at the pre-trained weights of the R50-FPN model. The hyper-parameters



(a) high saturation



(b) unmodified



(c) blue light

Figure 3: Example output of Mask R-CNN at 0.05% bounding box confidence threshold for the three different recording modes (high saturation, unmodified, blue light). The coloured boxes are bounding box predictions by the Fast R-CNN module of the Mask R-CNN, the coloured areas inside each bounding box illustrate the instance segmentation masks, respectively.

are: We trained 750 epochs with a learning rate of 0.001, decaying the learning rate with the factor 0.1 at one quarter and at half of the total training, respectively. Training time was consistently below 15 minutes for each run under the usage of a Quadro RTX 4000 (NVIDIA, California, USA) graphics card.

3.4 Negative Background Detection

We try to emulate an automatic negative background detection. A prerequisite for that is that the initial video files actually contain frames where no novel objects are on. The detection as the background is the absence of detection of any kind. Otherwise, we use humanly pre-extracted backgrounds with no novel objects on them, since our research focus is not on background detection. Filtered segmentation will be pasted into those backgrounds and become images in our artificial training data set. Not pasting on the background without the novel object could otherwise lead to failed training and non-detection.

3.5 Overlap of Segmentations

One can decide on different parameters when a pixel does belong to the final masks or not. We propose the following strategy: If more than a certain percentage of masks have one pixel in common, this pixel will be included in the final mask.

Algorithm 1 Pseudo-code for mask unification

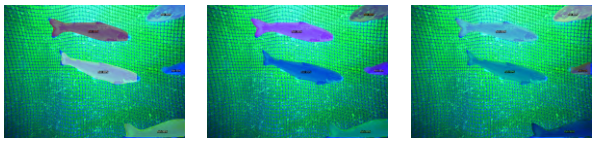
Data: Set of bounding boxes B
Result: Final mask set M
for $n \in B$ **do**
  for $k \in B$ **do**
    if $IoU(n, k) \geq 80\%$ **then**
      Add k to group G_n
    end
  end
  if $IoU(G_n) \geq 50\%$ **then**
    $m = take_{55\%}(G_n)$
    Add m to M
  end
end
Delete all duplicates in M.

Figure 4: Pseudo-code for mask unification with IoU thresholds. Let $IoU(\cdot)$ be the IoU function (calculates IoU of the input) and $take_{55\%}(\cdot)$ a function with a group of segmentation masks as input and a segmentation mask as output where every pixel position is covered in at least in 55% of the input mask group.

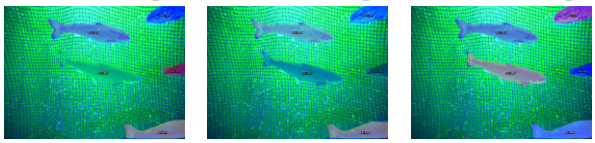
In Figure 5 and Figure 6 we compare the different mask results with different percentage thresholds, ranging from 5% to 90%. 100% of the masks is equivalent to the intersection of all masks for that novel object. Our findings indicate that the best percentage lies at 55%. Since we forbid self-intersecting and other invalid shapes (shapes with hull), the highest percentages of 100% and 95% are not yielding results. That means that needing all mask to contain certain pixels result in misfits for the final mask. On images as in Figure 5, where the initially segmented masks for different classes show low variability, the percentage threshold does not matter. Figure 6 shows that both over-detection on too-low bounds and under-detection on too-high ones can happen on more challenging images. Therefore, the percentage threshold has to be chosen with care.

3.6 Experimental Design

We ran our proposed method on three videos, respectively. One video was filmed with high saturation, one

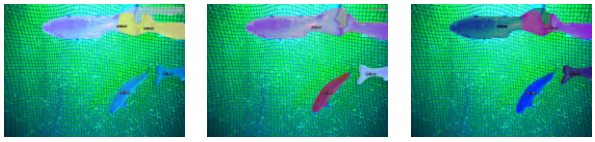


(a) 5% overlap (b) 25% overlap (c) 45% overlap

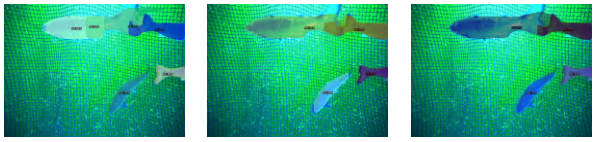


(d) 55% overlap (e) 75% overlap (f) 90% overlap

Figure 5: Detection results on the easier image on different mask overlap percentages.



(a) 5% overlap (b) 25% overlap (c) 45% overlap



(d) 55% overlap (e) 75% overlap (f) 90% overlap

Figure 6: Detection results on the more challenging image (fish overlap) on different mask overlap percentages.

in natural light without image processing and the last one with a blue light source. We use the term "recording mode" later to refer to the "high saturation", "unmodified" or "blue light" video respectively. We used bootstrapping for training: 10, 25 and 50 images were taken from the bag of frames of each video for the proposed method and training, respectively. In testing, a sample of 25 images was randomly chosen out of the out-of-bag sample frames. Training images, therefore, do not appear in the testing set, respectively. The confidence threshold for detection of 80% was chosen for testing (after training the network, see Section 3.3.4), and we counted the number of correct detections, overlaps and incorrect detections and the total number of fish on each test image (see Section 3.7). In total, we did the testing procedure $n = 5$ times per recording mode (high saturation, unmodified, blue light) and number of training images (10, 25, 50) in order to report the mean and standard deviation of correct, overlapping and incorrect detections as statistics.

3.7 Counting Strategy

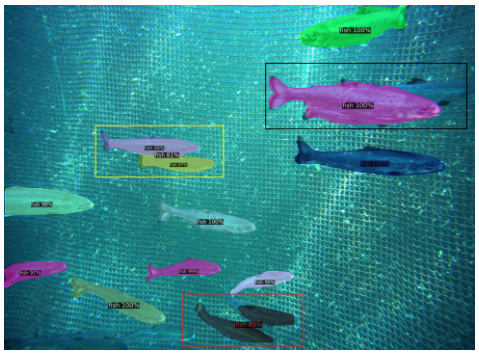


Figure 7: Test image from the recording mode unmodified with several overlaps. Yellow box: A masks containing two fish even though the two fish are respectively in an instance segmentation already. Black box: One part of one fish is mixed up in the instance segmentation mask of another fish, leading to a wrong mask. Red box: Two fish are in one overlapping instance segmentation mask even though they do not touch each other pixelwise.

We counted the total number of fish on the test images and categorized segmentation masks into different categories: Detection, overlap and incorrect. Fish with only the caudal fin (fin at the end of the fish) on the test images were not counted as fish. A mask fell under the category "detection" if the mask covers at least the whole main fish body (body without fins). The category "overlap" was used when a mask spanned over multiple fish since it is no longer an instance segmentation. Such masks can appear multiple times, as shown in Figure 7. "Incorrect" was the label for a mask if it did not contain the main body of the fish (e.g. head is not included in the mask) or other objects (not fish). By this counting strategy, the number of total instance segmentation masks on one test image equals the sum of the categories "detection", "overlap" and "incorrect". We note that the number of instance segmentation masks does not necessarily coincide with the counted fish in total on test images. One example is shown in Figure 7 (yellow box), where two correct instance segmentation masks contain one fish respectively, but another partially contains both fish. The overlap mask, therefore, increases

the number of total masks without increasing the number of fish on the image.

4 Experimental Results

Across the bootstrap tests (25 test images from the out-of-bag samples), an average of $70.8(\pm 7.3)$ fish in total were counted for the high saturation recording mode, $132.3(\pm 34.5)$ for the unmodified one and $232,5(\pm 49.1)$ for the blue light one.

4.0.1 Recording Mode as Factor for Detection

Figure 8 illustrates the testing results with bootstrap. The detection percentage varies across recording modes. While average detection results for the high saturation/unmodified recording mode mark around 80%/60% depending on the number of images, respectively, the average detection results for the blue light recording mode never get higher above 40%. Testing on the high saturation recording mode video frames yielded the best results in terms of detection, but we also report the highest percentage of overlaps among the recording modes across the number of images.

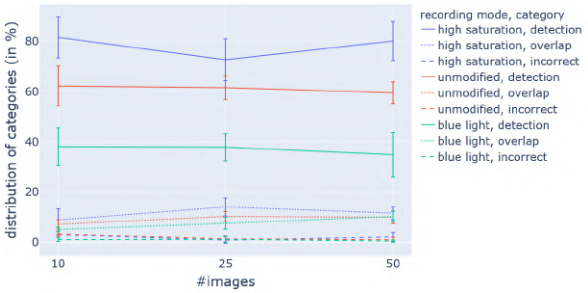


Figure 8: Bootstrap testing results with error bars divided in the categories detection (full line), overlap (dotted line) and incorrect (dashed line) for the recording modes high saturation (blue), unmodified (red) and blue light (green). The x-axis describes the number of training images, and the y-axis describes the occurrence percentage of each category relative to the number of fish on test images.

4.0.2 Number of Training Images

We calculated Pearson correlation coefficients between the number of training images and the different detection categories for every recording mode. We report two significant correlations ($p < 0.05$): For the blue light recording mode, the number of training images and the percentages of overlap masks have a strong positive correlation ($r = 0.78$). For the unmodified recording mode, the number of training images and the percentages of incorrect masks have a moderate negative correlation ($r = -0.62$). There is no significant correlation between the number of images and the percentages of masks in the category "detection" for the different recording modes, respectively. The analysis was based on a sample of $N = 5$ observations respectively (number of repetitions in the bootstrap testing).

5 Discussion

Our proposed method worked best in the high saturation recording mode, then in the unmodified, and then in the blue light recording mode as illustrated in Figure 8, which is revealing for the image quality needed. It seems to be beneficial to have a high gap in the colour values for instance segmentation.

Since the number of images is not correlated with the detection results, we interpret that the pre-trained Mask-RCNN model does not profit from additional images above 10 images as a training basis before being operational for the specific lighting recording mode in our study case. We want to note that other factors may play a role which would potentially increase the number of training images needed for training, such as rapidly changing environments in one video.

The relative number of incorrect is relatively low, suggesting that the method is more a "hit-or-miss" approach, meaning that we either get proper segmentation or not at all. In our testing neither background, marine snow, nor foreign objects seemed to have been fused with the fish instance segmentation masks, which we would have marked as "incorrect".

The biggest source of instance segmentation error originates from the category "overlap". We want to also emphasize that it seems to be a tendency from the trained model to seek proximity between multiple fish individuals, leading to multiple fish being marked as one fish via masks, even though they are separated by the background, as in Figure 7 (red box).

To counter non-detection, one could think to decrease the bounding box confidence threshold during the detection under testing. Lowering the bounding box confidence threshold is not practical though; it introduces

both instance segmentation masks which are sensible and not-sensible, which are not tolerable during testing. Therefore, we preferred to use the resized images as duplicates to produce more input images, leading to more masks for the proposed method.

5.0.1 Special Case Of Aquaculture

The case of aquaculture poses fewer challenges to objects when changing the background or lighting of the recorded footage. The recordings are on a sea cage farm, resulting in a quasi-non-changing and closed spatial environment per video, a controlled number of fish species and meta information about them (husbandry data). On the other hand, physical phenomena such as changing light conditions and changing light reflection, light attenuation, fish hiding behaviour and the fish physiology (reflective skin, blending in with the environment) pose challenges in data collection [11]. The background environment and colouring are subject to change between videos, as well as changing light conditions, which causes performance differences in our method (see Figure 8).

5.0.2 Limitations

We had consistent overlap across all three recording modes (see dotted line in Figure 8), suggesting a bigger underlying problem. Fish shoals or in general fish in higher density are bound to overlap from a 2D camera perspective, but our method only works if instances are cut off by the background. Otherwise, we can observe results as in Figure 7, where the two fishes on the left share a mask. This phenomenon is highly dependent on the image. If the fish are not crowded like in Figure 9, no overlap problems occur and all available instance segmentation masks are all separated and solely contain one fish.

Another limitation comes from the data collection: The lack of variation in the videos. If the environment in a video surprisingly changes which can occur in the real world, our model probably cannot keep up with its segmentation task, since the first extracted images may not mirror the environment correctly and corrupt model results. We are also prone to detect other foreign objects. Even in this sea cage example, we cannot guarantee that the novel object is a fish, it could also be waste or plastics/machinery. So, while we claim to have an algorithmic pipeline to segment fish, the knowledge that the desired object exists in a given video is needed.

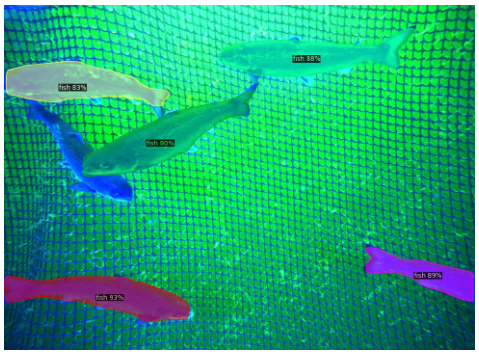


Figure 9: Mask R-CNN model prediction of a test image with after our method. All but one fish instance are marked with an instance segmentation mask, and no overlap occurs.

6 Conclusion

In this work, we proposed a low-cost method for unsupervised underwater fish instance segmentation using video frames. We showed varying results for different lighting recording modes and discussed the number of images needed for the proposed method.

One promising improvement could entail the implementation of a tracking algorithm for the fish masks to consider the temporal nature of the videos. This may lead to minimising overlap detection and significantly improving results.

All in all, we hope that this proof of concept shows that the current apparent paradigm “more annotated data, more knowledge” is not necessarily true; we would rather pledge for that that we already have enough annotated data but need to tickle out a little more info for our tasks.

7 Funding

We are grateful to the Norwegian Research Council, Project number 32330, for funding.

8 Acknowledgments

Acknowledgement to SubC3D AS for providing the original video/images taken with the underwater drone ARV-i (Transmark Subsea, Bergen).

References

[1] M. Buric, M. Pobar, and M. Ivašić-Kos. Ball detection using yolo and mask r-cnn. pages 319–323, 12 2018. doi: 10.1109/CSCI46756.2018.00068.

[2] 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. doi: 10.1109/CVPR.2016.350.

[3] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. doi: 10.1109/CVPR.2017.261.

[4] M. Føre, K. Frank, T. Norton, E. Svendsen, J. A. Alfredsen, T. Dempster, H. Eguiraun, W. Watson, A. Stahl, L. M. Sunde, C. Schellewald, K. R. Skøien, M. O. Alver, and D. Berckmans. Precision fish farming: A new framework to improve production in aquaculture. *Biosystems Engineering*, 173:176–193, 2018. ISSN 1537-5110. doi: 10.1016/j.biosystemseng.2017.10.014.

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. doi: 10.1109/CVPR.2012.6248074.

[6] R. Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587 TS – CrossRef. IEEE, 62014. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.81.

[8] E. Glikson and A. Woolley. Human trust in artificial intelligence: Review of empirical research. academy of management annals (in press). *The Academy of Management Annals*, 04 2020. doi: 10.5465/annals.2018.0057.

[9] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):386–397, 2020. doi: 10.1109/ICCV.2017.322.

[10] M. Jian, X. Liu, H. Luo, X. Lu, H. Yu, and J. Dong. Underwater image processing and analysis: A review. *Signal Processing: Image Communication*, 91:116088, 02 2021. doi: 10.1016/j.image.2020.116088.

[11] I. Kjerstad. Underwater imaging and the effect of inherent optical properties on image quality. 2014. URL `http://hdl.handle.net/11250/245550`. [Accessed Sep 08th, 2022].

[12] B. Kvæstad, T. Nordtug, and A. Hagemann. A machine vision system for tracking population behavior of zooplankton in small scale experiments: a case study on salmon lice (lepeophtheirus salmonis krøyer, 1838) copepodite population responses to different light stimuli. *Biology open*, 9, 06 2020. doi: 10.1242/bio.050724.

[13] 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 *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8693 LNCS, pages 740–755, 2014. ISBN 9781479951178. doi: 10.1007/978-3-319-10602-1_48.

[14] T. Porto Marques, M. Cote, A. Rezvanifar, A. Branzan Albu, K. Ersahin, T. Mudge, and S. Gauthier. Instance segmentation-based identification of pelagic species in acoustic backscatter data. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4373–4382, 2021. doi: 10.1109/CVPRW53098.2021.00494.

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE transactions on pattern analysis and machine intelligence*, 39 (6):1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

[16] M. Saberioon, A. Gholizadeh, P. Cisar, A. Pautsina, and J. Urban. Application of machine vision systems in aquaculture with emphasis on fish: state-of-the-art and key issues. *Reviews in Aquaculture*, 9(4):369–387, 2017. doi: 10.1111/raq.12143.

[17] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. [Accessed Sep 08th, 2022].