A segmentation-based approach for polyp counting in the wild

Vitjan Zavrtanik[a,*], Martin Vodopivec[b,c], Matej Kristan[a]

[a] *Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia*
[b] *Marine Biology Station, National Institute of Biology, Fornače 41, 6330 Piran, Slovenia*
[c] *Slovenian Environment Agency, Vojkova 1b, 1000 Ljubljana, Slovenia*

Abstract

We address the problem of jellyfish polyp counting in underwater images. Modern methods utilize convolutional neural networks for feature extraction and work in two stages. First, hypothetical regions are proposed at potential locations, the features of the regions are extracted and classified according to the contained object. Such methods typically require a dense grid for region proposals, explicitly test various scales and are prone to failure in densely populated regions. We propose a segmentation-based polyp counter – SegCo. A convolutional neural network is trained to produce locally-circular segmentation masks on the polyps, which are then detected by localizing circularly symmetric areas in the segmented image. Detection stage is efficient and avoids a greedy search over position and scales. SegCo outperforms the current state-of-the-art object detector RetinaNet (Lin et al., 2017) and the recent specialized polyp detection method PoCo (Vodopivec et al., 2018) by 2% and 24% in F-score, respectively, and sets a new state-of-the-art in polyp detection.

Keywords: Circular object detection, Semantic segmentation, Automated Counting, Jellyfish Polyp, Convolutional Neural Network

1. Introduction

Jellyfish (Scyphozoa) blooms have attracted significant interest of researchers over the recent years (Kogovšek et al., 2018; Brotz et al., 2012; Condon et al., 2013). During blooming, the jellyfish population rapidly increases in a relatively small geographical area, which importantly affects the local ecosystem. Bloom and jellyfish population dynamics prediction can be established by analyzing the population dynamics of polyps, which are one of the many forms in the jellyfish complex life cycle.

Widmer et al. (2016) analyzed polyp populations and experimentally established relation between the water temperature and salinity and the bloom magnitude. But their study was restricted to in vitro studies and the polyp density was manually estimated. Only a fraction of works focus on in situ studies. Polyp attachment substrates are scarce in natural environments, thus they usually cluster on small regions, like oyster shells. Using such regions, several studies (Hočevar et al., 2018; Vodopivec et al., 2018) have established important relations between polyp numbers in nature and the resulting jellyfish population dynamics. Most of these studies, however, rely on manual polyp counts.

Figure 1: Polyp detection is particularly difficult due to significant overlaps (top left) and a high degree of appearance variation between the polyps (top right). Images are often taken under challenging conditions which causes significant blurring in regions (bottom left) or poor lighting (bottom right).

*Corresponding author. Phone:+38640895838
Email address: vitjan.zavrtanik@fri.uni-lj.si (Vitjan Zavrtanik)

Manual polyp density estimation requires counting polyps in underwater images. These images might contain over a thousand polyps, which makes manual counting a time-consuming and laborious task. Furthermore, the annotation requires a high degree of expert knowledge to accurately classify ambiguous regions/objects as a polyp or a background structure (see Figure 1). This calls for automated polyp detection methods. The methods have to be particularly robust to deal with a large appearance variability of polyps and the various conditions under which the photographs are taken. These include lighting, camera focus, motion blur and background color.

Recently, an automated polyp counter was proposed by Vodopivec et al. (2018). The approach is based on the classical object detection architecture: a weak fast detector proposes potential polyp boxes and then computation-intensive features are extracted and classified. The approach suffers from several drawbacks. Polyps vary in size significantly within a single image, which requires a greedy search over several scales. This increases the computational cost as well as the potential for false positives. Application of non-maxima suppression to the detections that passed the final classifier results in missed detections in densely populated regions. The detection pipeline is composed of algorithmically non-homogeneous modules. Thus the opportunity to improve robustness by end-to-end training is lost. For practical applications, a streamlined end-to-end trainable detector is required that would address scale variability, retain a high detection rate, and allow re-training on alternative datasets obtained by different hardware.

Our main contribution is a new streamlined automated polyp counter. To deal with clutter and high appearance and scale diversity, we cast polyp detection as a semantic instance segmentation problem that capitalizes on approximate circular symmetry of the polyps (Figure 2). A convolutional neural network is designed to segment all pixels in the image into polyp and background pixels. The network is trained such to enhance the circular symmetry in the segmentation mask. A detector is then designed to extract approximately circular structures from the mask, using an approach based on distance transform. The detector simultaneously localizes polyps at several scales without explicit enumeration of the scales and naturally handles clustered polyps that partially overlap in the image. The approach is easily trainable, robust and fast.

The proposed segmentation-based polyp counter, SegCo, is evaluated on the recent challenging dataset of polyps in the wild proposed by Vodopivec et al. (2018). Results show that the approach is highly robust, it outperforms a slow but general state-of-the-art object detector RetinaNet Lin et al. (2017), by 2% in F-1 measure and outperforms the currently best method for polyp detection, PoCo Vodopivec et al. (2018), by 24% in F-1 measure. A licence-free Python implementation of SegCo will be made publicly available. We believe this will substantially contribute to the marine biology community as a trainable toolbox for polyp counting and similar applications, allowing to accurately process large datasets.

As an additional contribution we improve the annotations of the currently largest annotated polyp counting dataset Vodopivec et al. (2018) to allow a more accurate benchmarking of polyp counting methods. The revised PoCo dataset will be made publicly available as well.

2. Related Work

The field of object detection and segmentation has been very active in recent years. Current state of the art object detection methods (Ren et al., 2015; Lin et al., 2017; He et al., 2017) are often based on convolutional neural networks and achieve excellent results on object detection benchmarks such as (Lin et al., 2014; Russakovsky et al., 2015). These benchmarks strive to evaluate performance on general object detection problems but often do not accurately represent domain-specific problems such as polyp counting, that require detection of an extremely large number of densely positioned objects.

We focus on works that address counting of highly cluttered, overlapping objects. For an extensive overview of general object counting methods, we refer the reader to Heinrich et al. (2019). Polyp counting shares many characteristics with dense detection problems like nuclei segmentation (LaTorre et al., 2013), phytoplankton detection (Verikas et al., 2012), cell counting and detection (Xie et al., 2018). LaTorre et al. (2013) addressed segmentation of overlapping nuclei by finding concave regions in the segmentation mask. This method enables separation of near-by objects, but is not robust to significant overlaps between the objects. Ferrari et al. (2017) segment groups of bacterial colonies and then infer the number of colonies in each connected component using a convolutional neural network regressor. Schoening et al. (2012) estimate deep sea megafaunal densities by a confidence map obtained by per-pixel application of an SVM Cortes & Vapnik (1995) trained MPEG7 features. The approach is restricted to uniform backgrounds with little or no overlap between the objects.

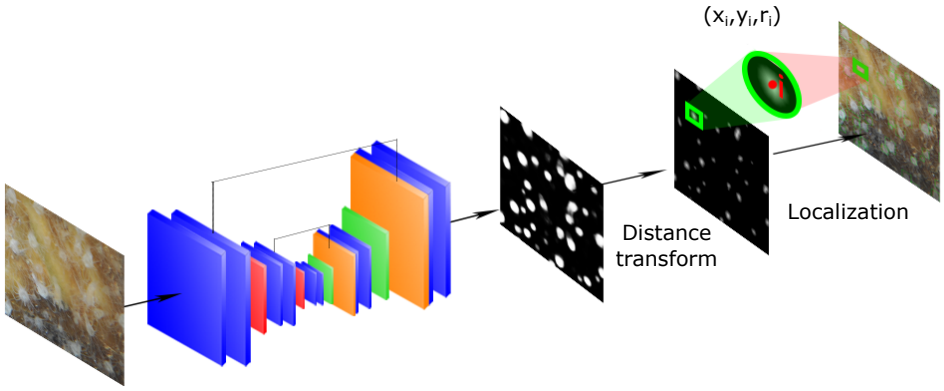

Figure 2: Our polyp counter SegCo consists of an encoder-decoder convolutional neural network (CNN) for image segmentation and segmentation mask post-processing steps for polyp localization. The CNN is trained to produce a mask image with individual polyps indicated by approximately circularly symmetric regions. A distance transform is computed and local maxima (x_i, y_i) are identified in the distance transform map, where x_i and y_i are the coordinates of the local maxima. The value of the distance transform r_i corresponds to the detection size in pixels.

Several approaches avoid explicit object detection by estimating the object density by numerically integrating over an object density map (Xie et al., 2018; Lempitsky & Zisserman, 2010). Perko et al. (2013) adapted this approach for counting people in crowds from aerial images. Their method estimates the object density map by learning feature importance from dense feature maps of object detection responses and SIFT features (Lowe, 2004). Foroughi et al. (2015) apply sparse representation classification in conjunction with dimensionality reduction to estimate the object density. These approaches, however, are impractical in many applications since count accuracy cannot be easily validated by a human supervisor.

The most closely related work to our own is the recent polyp counting method proposed by (Vodopivec et al., 2016, 2018). They propose a two-stage detection method that first generates candidate regions using aggregated channel features (Dollár et al., 2010) and extracts region feature vectors using the AlexNet (Krizhevsky et al., 2012) neural network, pretrained on ImageNet (Russakovsky et al., 2015). The method uses an SVM (Cortes & Vapnik, 1995) classifier to classify the extracted feature vectors as polyp or background regions. The method is algorithmically non-homogeneous and does not allow end-to-end training.

3. Segmentation-based polyp counter

We propose a semantic-segmentation-based polyp counting method SegCo (Figure 2). A segmentation convolutional neural network (Section 3.1) is applied to classify each pixel into a foreground (polyp) or background. The network is trained to produce locally circular segmentation masks on polyp images. The generated segmentation mask is then interpreted by a scale-invariant polyp localization method (Section 3.3). The steps of the method are detailed in the following subsections.

3.1. The polyp segmentation network architecture

An encoder-decoder network with a U-Net (Ronneberger et al., 2015) architecture is used in our segmentation network. The encoder is constructed from *convolutional blocks*. We define a *convolutional block* as two convolutional layers followed by a max-pooling layer (Figure 3). In many commonly used convolutional neural networks the number of filters in the convolutional layers doubles after each reduction in the spatial dimensions as in (Ronneberger et al., 2015; He et al., 2016; Simonyan & Zisserman, 2014). We define the base number of filters as the number of filters used on the first layer. As in (Ronneberger et al., 2015; He et al., 2016; Simonyan & Zisserman, 2014) the number of filters is doubled after each max pooling layer (Figure 3). The final convolutional block in the encoder is followed by a *bottleneck block*, which is composed of a dropout layer followed by two convolutional layers. The dropout layer serves as a regularizer to prevent overfitting.

The decoder uses the same number of blocks as the encoder, however, the decoder convolutional blocks are composed of an upsampling layer and two convolutional layers. A ReLu activation is applied on the output of each convolutional layer with the exception of the final layer where a sigmoid activation is applied instead. Skip connections are inserted between the encoder and

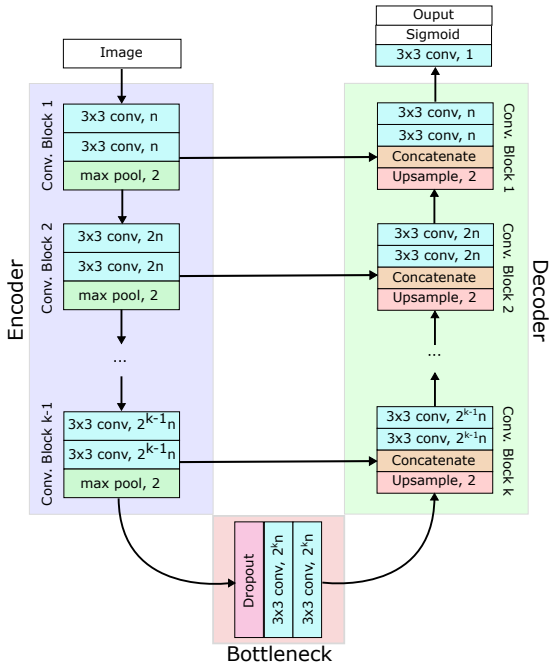

Figure 3: A U-net based segmentation network architecture used in SegCo. The encoder, decoder and bottleneck portions of the network are marked with purple, green and red rectangles respectively. Skip connections are denoted as straight horizontal arrows.

decoder convolutional blocks. These are implemented as a channel-wise concatenation of the output of the last convolutional layer of a convolutional block and the output of the upsampling layer of the decoder convolutional block. The combined features are the input to the first convolutional layer of the decoder convolutional block. We denote the model constructed from k blocks and n base layers as SegCo$^{(k,n)}$.

3.2. Enforcing circularity in segmentation masks

An accurate representation of the object shape and position is critical for training the polyp segmentation network. But manual per-pixel segmentation mask annotation is often infeasible in practice as segmenting polyp images requires annotation of several hundred individual polyps. The polyps are therefore usually annotated by bounding boxes (x_i, y_i, w_i, h_i), where (x_i, y_i) is the position and (w_i, h_i) are the width and height of the bounding box.

To facilitate polyp detection by detection of circularly symmetric regions, we convert the ground truth annotations into circles (x_i, y_i, r_i). The bounding box center is taken as the circle center, while the radius is estimated conservatively by half of the smaller rectangle edge, i.e.,

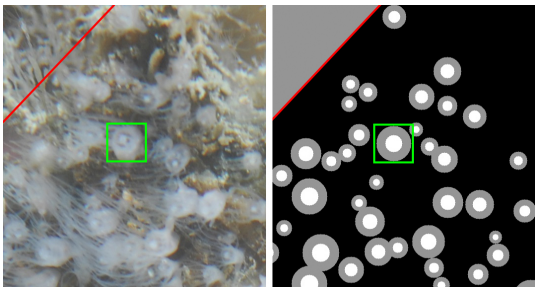

Figure 4: A training image (left) and the generated segmentation ground truth segmentation mask (right). An example of the annotated ground truth rectangle is shown in green. In the mask image, the polyp and non-polyp labels are denoted by white and black, respectively, while the ignore regions are denoted by gray.

$$r_i = \lfloor \frac{min(w_i, h_i)}{2} \rfloor. \quad (1)$$

In this approximation, the pixels immediately outside of the circle may also correspond to the polyp. Therefore an additional *ignore* label l_d is used to annotate pixels between radius r_i and a larger radius $\tilde{r}_i$ during learning ($\tilde{r}_i = max(w_i, h_i)$). Example of the automatically generated ground truth labels is shown in Figure 4.

The network is trained using a binary cross-entropy loss (L_{bce}), which is modified to account for the *ignore* labels:

$$L_{bce}(p_j, l_j) = \begin{cases} -log(p_j), & \text{if } l_j = 1 \\ 0, & \text{if } l_j = l_d \\ -log(1 - p_j), & \text{otherwise} \end{cases}, \quad (2)$$

where l_j denotes the training mask value assigned to each pixel j and p_j denotes the predicted probability of the pixel j belonging to the polyp class. The pixels located within radius r_i surrounding the location (x_i, y_i) of polyp i are assigned a label $l_j = 1$, meaning they represent a polyp, while those between radius r_i and $\tilde{r}_i$ are assigned a label $l_j = l_d$, which ensures these pixels are ignored. In our experiments the ignore label was set to $l_d = 0.5$.

With circular polyp masks, we intentionally introduce a bias for circular symmetry, while reducing the amount of noise by removing the surrounding area from the loss calculation. This encourages the network to infer circularly symmetrical regions at polyp locations.

3.3. Distance consensus point detection

The segmentation network from Section 3.1 assigns a probability of polyp location at each pixel. By thresholding this probability map at a threshold Θ_{th}, a binary segmentation mask is obtained. Polyp locations

and sizes are then determined from the segmentation mask. In densely populated regions, polyps may touch or mildly overlap, thus individual connected components in the mask image may potentially correspond to several polyps. The segmentation network is trained to produce locally circular segmentations on the polyps (see Section 3.2 for details). Thus we can exploit this property to detect polyps by identifying locations of local circular symmetry in the segmentation mask.

The centers of approximately circular objects may be determined by distance consensus points, i.e., points equally distant from the local mask edges. The inferred segmentation mask is thus first inverted (i.e., subtracted from 1) and a distance transform is computed according to Rosenfeld & Pfaltz (1968). Distance consensus points are identified as local maxima on the distance transform map. The polyp size can then be simply estimated as the value of the distance transform map at the detected polyp center, since this value is by definition the distance to the local edge in the binary mask. The polyp localization is thus scale-invariant in that it avoids the need for greedy search over a number of scales.

The proposed method allows accurate detection even when polyps are touching or overlapping. Figure 5 shows an idealized mask generated by a large polyp overlapping with a small one. The distance transform computed on this mask contains two clear local maxima corresponding to the smaller and the larger polyp centers. The value of the distance transform at the detected centers identifies the size accurately despite the significant overlap.

Example of the detection pipeline is shown in Figure 6. Even though some of the polyps are non-circular, they appear circular in the inferred segmentation mask, thus facilitating detection through circularity interpretation. Note that polyps are well localized even under substantial overlaps with other polyps and a large diversity in polyp size.

4. Evaluation

4.1. Implementation Details

The experiments were run on a PC with a *Intel(R) Core(TM) i7-6700K @ 4.00 GHz (8 CPUs)* processor, 32 GB RAM and *NVIDIA GeForce GTX 1070* graphics card. The network was trained by Adam with a 10^{-4} learning rate. The original images are too large to fit into our GPU memory. We thus divide them into 512x512 overlapping blocks and process each block individually.

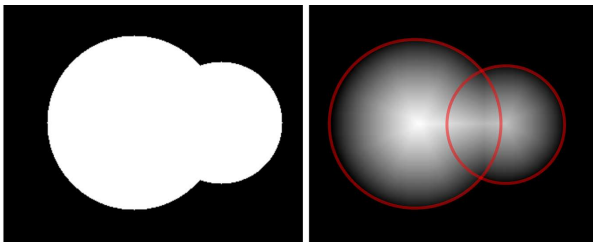

Figure 5: A synthetically generated mask of two overlapping polyps (left) and the corresponding distance transform (right). Circles are detected by the local maxima in the distance transform map along with the distance transform values at those positions (shown in red in the distance transform mask).

4.2. Dataset and evaluation measures

The recent polyp counting dataset from Vodopivec et al. (2018) was used to analyze the proposed segmentation-based polyp counter (SegCo). The images are recorded with resolution of 4288×2844 pixels and contain oysters with attached polyps (Figure 7). An oyster occupies a large central part of the image. Polyps are annotated only within a selected region on the oyster, thus a region of interest (ROI) polygon is defined for each image as well.

The training set was created from 30 images of oysters with polyps from Dataset B and other supplementary datasets from Vodopivec et al. (2018). These images were originally annotated in one pass by a single annotator which means some of the polyps might have been missed Vodopivec et al. (2018). The training dataset thus contains 32.685 polyp annotations. See Figure 7 for an example of the annotated image.

The test set was created from the Dataset A published in (Vodopivec et al., 2018). This is a carefully curated set of 7 images for accurate evaluation of polyp detection methods. Individual images of the test set are selected to include a variety of polyp appearances. Each image was annotated by several expert annotators multiple times in (Vodopivec et al., 2018) to minimize the annotation errors. In the following we refer to this data set as the PoCo test set.

We apply the polyp detection evaluation protocol from Vodopivec et al. (2018) in our experiments. The protocol pairs the detections with ground truth annotations by globally minimizing the distance between the pairs centers using the LAPJV pairing algorithm from Jonker & Volgenant (1987). An individual detection can only be paired with a ground truth annotation (and classified as a correct detection), if their centers are separated by less than a pre-defined distance threshold τ_d. This threshold is set in Vodopivec et al. (2018) as the

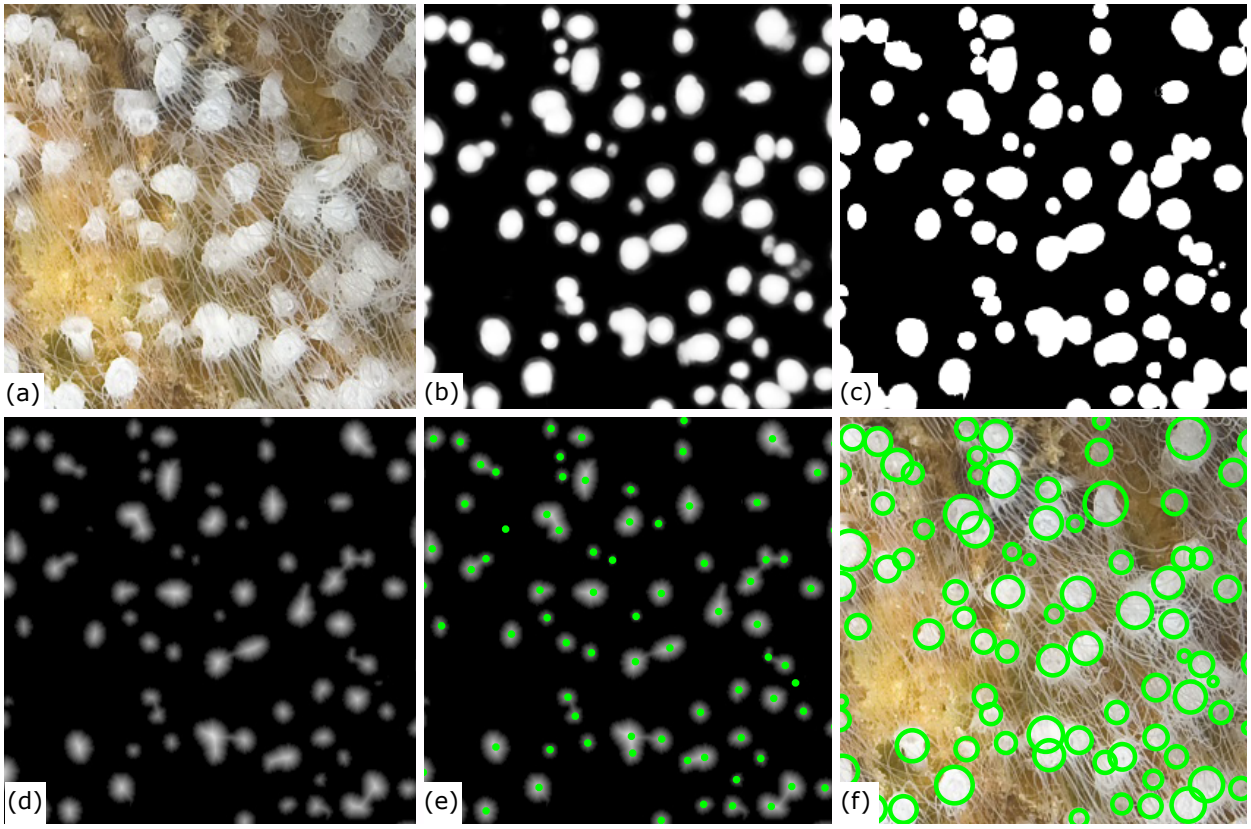

Figure 6: Intermediate steps of the SegCo detection pipeline. The image with polyps (a) is processed by the segmentation network to produce a segmentation map (b). The map is binarized (c) and a distance transform is computed (d). Local maxima on the distance transform map are shown in (e) and the detected polyps in (f).

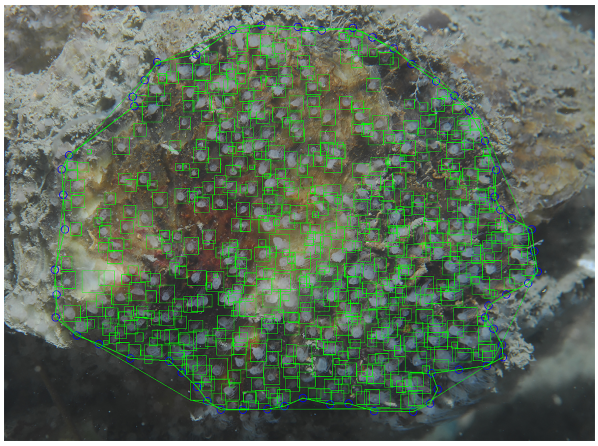

Figure 7: Annotated polyps on an oyster photo. Polyp bounding boxes are marked with green rectangles. The region of interest polygon is shown in thick green line, with blue circles marking the polygon vertices.

median of the annotation rectangle diagonals computed on the training set.

Detection performance is evaluated by the average recall (AR), average precision (AP) and F-1 measures. The count accuracy is evaluated by the count relative error (Rel. err.) and the ratio (Ratio) between the number of detections and the number of ground truth annotations as defined by Vodopivec et al. (2018).

4.3. Parameter Robustness Analysis

We first analysed the impact of the segmentation network depth and the number of filters used in each layer on the overall success of SegCo. In particular, the number of convolutional blocks and the base filter number was analyzed.

The results for the varying number of convolutional blocks and filters are shown in Table 1. We see that the performance generally improves with increasing the number of blocks and filters, but we also observe that similar results can be achieved with less complex networks. A partial loss in performance in networks with only $k = 2$ blocks could be explained by a smaller effective receptive field caused by the lack of pooling operations. This may be partially compensated by increasing the number of filters n as SegCo$^{(2,64)}$ performs significantly better than SegCo$^{(2,8)}$.

Next, we evaluated the inference time with respect to the number of convolutional blocks and filters. The time required for polyp detection on an single image is shown in Table 2. Inference speed is less affected by the increase in the number of blocks than the number of base filters since the pooling operation makes each

		Filters (n)			
Blocks (k)	**k**		**8**	**16**	**32**	**64**
	2	AR	0.84 ± 0.06	0.82 ± 0.13	0.83 ± 0.12	0.91 ± 0.05
		AP	0.77 ± 0.06	0.81 ± 0.05	0.83 ± 0.04	0.82 ± 0.04
		F1	0.80 ± 0.07	0.81 ± 0.07	0.83 ± 0.07	0.86 ± 0.04
	3	AR	0.89 ± 0.04	0.92 ± 0.06	0.90 ± 0.06	0.90 ± 0.06
		AP	0.86 ± 0.02	0.86 ± 0.02	0.89 ± 0.02	0.88 ± 0.02
		F1	0.87 ± 0.01	0.89 ± 0.02	0.89 ± 0.03	0.89 ± 0.03
	4	AR	0.88 ± 0.06	0.91 ± 0.05	0.90 ± 0.06	0.93 ± 0.03
		AP	0.87 ± 0.02	0.85 ± 0.02	0.85 ± 0.02	0.87 ± 0.02
		F1	0.88 ± 0.02	0.89 ± 0.02	0.88 ± 0.04	0.89 ± 0.02

Table 1: Average recall, accuracy and F-1 measures using a segmentation network with k blocks and n base filters with a fixed binarization threshold $\Theta_{\text{th}} = 0.44$. The performance generally improves with increasing the number of base filters and blocks, however similar results are achieved even with shallower networks.

successive block less computationally expensive. When compared to the number of parameters in Table 3, the inference times vastly varies between networks of similar complexities. We can see that the achieved detection performance measures of SegCo$^{(4,16)}$ and SegCo$^{(4,64)}$ are comparable even though the former has a significantly lower number of parameters (Table 3) and is consequently faster (Table 2).

	Filters (n)			
Blocks (k)	**8**	**16**	**32**	**64**
2	3.43 ± 0.32s	4.12 ± 0.47s	5.86 ± 0.51s	10.05 ± 0.89s
3	3.47 ± 0.36s	4.41 ± 0.49s	6.30 ± 0.60s	12.30 ± 1.04s
4	3.54 ± 0.43s	4.38 ± 0.53s	6.66 ± 0.63s	14.60 ± 1.23s

Table 2: Inference times per image using a segmentation network with k blocks and n filters.

	Filters (n)			
Blocks (k)	**8**	**16**	**32**	**64**
2	0.029×10^6	0.117×10^6	0.466×10^6	1.88×10^6
3	0.121×10^6	0.486×10^6	1.95×10^6	7.69×10^6
4	0.485×10^6	1.94×10^6	7.76×10^6	31.03×10^6

Table 3: Impact of k blocks and n filters on the number of network parameters. Even though SegCo$^{(4,16)}$ and SegCo$^{(3,32)}$ have a similar number of parameters, the network SegCo$^{(4,16)}$ has a much shorter inference time on test images.

An important parameter of the segmentation mask interpretation component of SegCo is the binarization threshold Θ_{th}. Figure 8 shows SegCo detection performance with respect to the parameter Θ_{th}. Results show that SegCo is largely robust to Θ_{th}, however F-1 measures start to decrease at very high or low Θ_{th} values. The performance is very stable for the interval $\Theta_{\text{th}} = (0.4, 0.6)$. In our experiments we use the threshold value $\Theta_{\text{th}} = 0.44$.

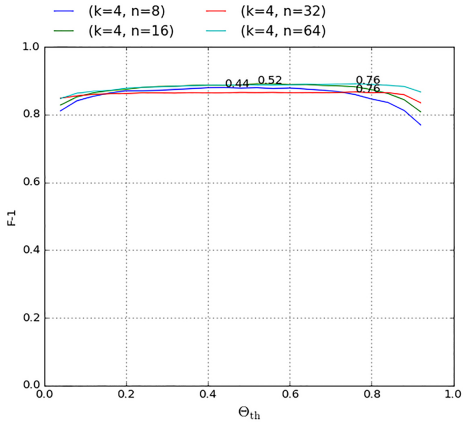

Figure 8: F-1 measure with respect to the mask binarization threshold Θ_{th}. SegCo performance is stable for a large range of Θ_{th} values.

4.4. Influence of data augmentation

We examined the impact of two types of data augmentation: (i) standard augmentation and (ii) extended augmentation. In *standard augmentation* horizontal and vertical image flipping and scale augmentation is applied to training data, where we increased or decreased the size of the image by 20%. The *extended augmentation* applies the standard augmentation and adds blurring by a Gaussian filter with a randomly chosen standard deviation from a uniform distribution on the interval $(12, 70)$, which corresponds to $(0.2\mu_{diag}, 1.2\mu_{diag})$, where μ_{diag} is the average diagonal length of polyp annotations.

The influence of data augmentation is quantified in Table 4. Standard augmentation improves detection on the majority of test images in comparison to not using the augmentation. It improves the AR measure by 4%. Extended augmentation further improves the performance on blurred regions which are poorly represented in the training set and often poorly annotated due to an increased level of ambiguity. Extended augmentation improves the AR measure by an additional 1% percent in comparison to standard augmentation.

4.5. Influence of the training set size

Since manual annotation is time consuming and error prone, achieving good results using a smaller data set is preferential. While each annotated image provides several hundred polyp learning examples, different images may not contribute to increasing the visual diversity of the dataset and might be redundant in this respect. In this experiment we measured the influence of the training set size to the overall detection performance.

We trained SegCo several times using a varying number of images in the training set. Figure 9 shows AR, AP and F-1 measures achieved by SegCo on the PoCo dataset with respect to the number of training images used in SegCo training. We can see that SegCo already performs reasonably well with a small number of images in the training set, provided that the variability of polyp visual features is well represented. We observe a slight trend of improving results with respect to the increasing number of images.

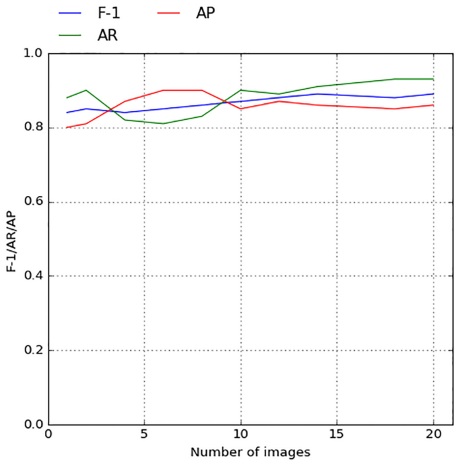

Figure 9: F-1, AR and AP measures achieved by SegCo with respect to the number of training images.

4.6. Comparison to the state-of-the-art

The proposed polyp counter SegCo was compared to the most recent state of the art polyp detector, PoCo (Vodopivec et al., 2018), and a state of the art method general object detector RetinaNet (Lin et al., 2017).

We used a ResNet-50 (He et al., 2016) pretrained on ImageNet (Russakovsky et al., 2015) as the backbone network for RetinaNet, which we then fine-tuned for polyp detection on our training set using the Adam optimizer and a learning rate of 10^{-4}. We use the same data augmentation protocol for all networks.

To ensure a fair comparison to PoCo results from Vodopivec et al. (2018), we evaluate our method using the median based threshold τ_d (median of bounding box diagonals) as proposed in Vodopivec et al. (2018). We compare two versions of our method SegCo$^{(4,16)}$ and SegCo$^{(4,64)}$ with the other methods. The results are reported in Table 5. SegCo$^{(4,16)}$ achieves a higher AP but a lower AR when compared to SegCo$^{(4,64)}$. SegCo achieves a 16% higher AR than PoCo and a 3% higher AR than RetinaNet, however it achieves a lower AP.

	S		E		N	
Image	rec.	prec.	rec.	prec.	rec.	prec.
# 1	0.93	0.85	**0.95**	0.83	0.92	**0.86**
# 2	**0.95**	0.88	**0.95**	0.86	0.92	**0.89**
# 3	0.93	0.86	**0.94**	0.84	0.91	**0.87**
# 4	0.92	0.89	**0.93**	0.88	0.87	**0.91**
# 5	**0.94**	**0.87**	0.92	**0.87**	0.93	**0.87**
# 6	0.86	0.87	**0.91**	0.86	0.75	**0.92**
# 7	**0.91**	0.88	**0.91**	**0.89**	0.85	0.88
μ	92 ± 3%	87 ± 1%	**93±2%**	86 ± 2%	88 ± 6%	**88±2%**

Table 4: Recall and precision values on individual images of the test set using varying methods of image augmentation. Standard augmentation (S) improves recall values over using no augmentation (N). Extended augmentation (E) mildly increases the detection performance in otherwise under represented blurred regions.

RetinaNet achieves a lower overall polyp count error, since it achieves approximately the same AP and AR measures (i.e., false negatives are compensated by false positives).

Compensation of false negatives by false positives may cause better results in terms of Ratio and Relative error in some cases, but it is otherwise undesireable in counting by detection, since the number of false positives and negatives are unpredictable. This reduces the reliability of the object counting method.

Note that the threshold τ_d from Vodopivec et al. (2018) is rather large and is well suited for counting performance evaluation, but does not reflect the localization accuracy. In fact, this threshold compensates for missed detections by nearby false positives.

4.7. PoCo Dataset A revisited

Vodopivec et al. (2018), manual annotation is error-prone even for experts, who achieve a relatively low recall (87%) and usually miss a portion of the polyps in an image. In fact, our inspection of the false positive detections of the evaluated detection methods revealed a non negligible number of missing annotations in the test set. See Figure 10 for examples of missing annotations. Due to high precision of human annotators, missing annotations are much more common than incorrectly annotated background regions, however ambiguous regions are also sometimes annotated as polyps (see Figure 11).

To more accurately evaluate the polyp detection methods, we revised the test dataset by manually inspecting each false positive detection of SegCo$^{(4,16)}$, SegCo$^{(4,64)}$, PoCo and RetinaNet and improved the dataset by adding detections that were marked as false positives but contained a polyp (i.e., were true positives). A significant portion of false positive detections are in regions where region (polyp) classification is highly ambiguous due to significant blurring or poor lighting. We only expand the dataset with detections where we could tell with a high certainty whether the detected region contains a polyp or not. The regions which can not be easily classified remain false positives, even though we can not claim with certainty that they do not contain a polyp.

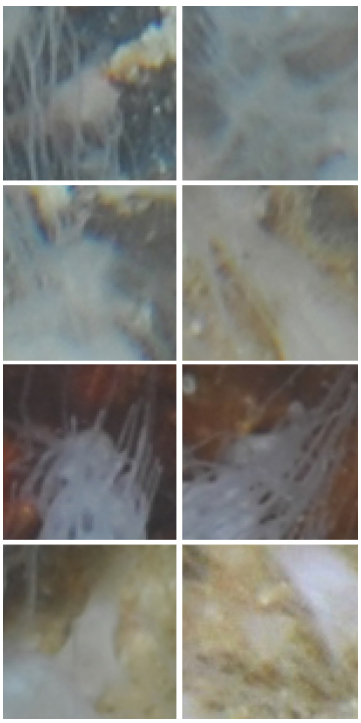

Figure 10: Polyps correctly identified by SegCo that were marked as false positives due to missing annotations. A high level of polyp occlusion, poor lightning and polyp transparency evidently make these regions difficult to annotate unambiguously.

Table 6 lists the changes in the revised test set. We can see that each image in the test set was missing 10% of polyp annotations on average, which demonstrates

Method	Ratio	Rel. err.	AP	AR	*F-1*
SegCo$^{(4,64)}$	1.09 ± 0.02	0.09 ± 0.02	0.91 ± 0.01	**0.99 ± 0.01**	0.95 ± 0.01
SegCo$^{(4,16)}$	1.05 ± 0.04	0.06 ± 0.03	0.94 ± 0.02	0.98 ± 0.02	**0.96 ± 0.01**
PoCo Vodopivec et al. (2018)	0.87 ± 0.15	0.17 ± 0.08	0.96 ± 0.06	0.83 ± 0.08	0.88 ± 0.03
RetinaNet Lin et al. (2017)	**1.00 ± 0.06**	**0.00 ± 0.05**	**0.96 ± 0.02**	0.96 ± 0.04	**0.96 ± 0.01**

Table 5: Detection results on the PoCo test set. The evaluation distance threshold τ_d equal to the median length of the annotation diagonals is used as in Vodopivec et al. (2018).

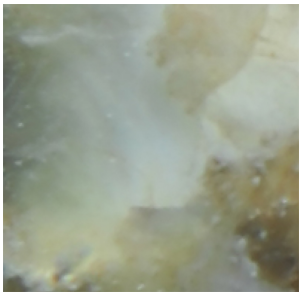

Figure 11: Ambiguous region that was incorrectly annotated as a polyp in the PoCo test set. Besides the color it contains little visual features that are typical for polyps and could easily be a similarly colored background structure.

Image	n_{gt}	n_{rev}	n_{add}	n_{rev}/n_{gt}
# 1	455	492	37	1.08
# 2	655	732	77	1.12
# 3	543	600	57	1.10
# 4	770	879	109	1.14
# 5	723	820	97	1.13
# 6	350	382	32	1.09
# 7	398	412	14	1.03

Table 6: Added annotations in the revised PoCo test set. n_{gt} lists the number of annotations in the PoCo test set, n_{rev} the number of annotations in the revised PoCo test set and n_{add} the number of added annotations.

how unreliable the manual annotation process can be, even with multiple manual re-annotations. A larger percentage of annotations were added in images #4 and #5 as they contain a high level of inter polyp occlusion, which seems to be a problem area for human annotators. On the other hand, very few annotations were added to image #7, due to poor lighting and camera focus, which increased the ambiguity of the image and made manual classification of false positives difficult.

4.8. Performance on the revised PoCo dataset

SegCo^{4,16},SegCo^{4,64}, PoCo and RetinaNet were evaluated on the revised PoCo dataset from Section 4.7 using a fixed τ_d. As argued in Section 4.6, the median bounding box diagonal length based τ_d value used in Vodopivec et al. (2018) is too permissive to accurately evaluate the localization ability of detection methods and allows false positive detections in crowded regions to compensate for false negatives. We have therefore applied a stricter threshold $\tau_s = 30$ pixels, which is approximately half of τ_d and is in the order of the smallest polyp diagonals in the dataset. The results are shown in Table 7.

The dataset correction gives the evaluated methods a healthy performance boost in precision due to the addition of missing polyps. This shows that AP is not very informative on the un-revised PoCo test set due to the relatively large amount of missing annotations. The results in Table 7 show that in comparison to the original PoCo test set results obtained using a stricter τ_d (Table 8), all evaluated methods suffer a slight drop in AR due to the additional annotations in the revised test set that they are not able to detect. We also see the drops in AR are not equal across the methods as RetinaNet suffers a 2% drop in AR, which is higher than the drop for SegCo with an AR drop of 1%. SegCo also has the lowest polyp count relative error on the revised PoCo test set, achieving an almost perfect polyp count ratio (*Ratio* = 0.99). One of the reasons for the RetinaNet performance drop could be due to the increased density of added annotations in already cluttered regions. RetinaNet suppresses non maximal detections which might be problematic in cluttered regions where the detection overlap is generally high. Another reason might be that the deep backbone of RetinaNet overfits the training set which could reduce its generalization ability and therefore render it unable to detect the additional polyps that are usually not well represented in the training set.

5. Conclusion

We addressed the problem of polyp counting in underwater images captured in the wild and proposed a segmentation-based detection method, SegCo, that achieves state-of-the-art results. A segmentation net is designed to infer a binary segmentation mask and the approximate circular symmetry of polyps is utilized to accurately interpret the generated segmentation mask. This is achieved by detecting distance consensus points,

Method	Ratio	Rel. err.	AP	AR	*F-1*
SegCo$^{(4,64)}$	**0.99 ± 0.02**	**0.01 ± 0.02**	0.95 ± 0.02	**0.94 ± 0.01**	**0.94 ± 0.01**
SegCo$^{(4,16)}$	0.96 ± 0.03	0.04 ± 0.03	**0.96 ± 0.02**	0.92 ± 0.03	**0.94 ± 0.01**
PoCo Vodopivec et al. (2018)	0.82 ± 0.16	0.23 ± 0.08	0.79 ± 0.08	0.63 ± 0.06	0.70 ± 0.03
RetinaNet	0.92 ± 0.05	0.08 ± 0.05	**0.96 ± 0.02**	0.89 ± 0.04	0.92 ± 0.01

Table 7: Detection results on the revised PoCo test set. A conservative evaluation distance threshold $\tau_d = 30$ pixels was used.

Method	Ratio	Rel. err.	AP	AR	*F-1*
SegCo$^{(4,64)}$	1.07 ± 0.02	0.07 ± 0.02	0.87 ± 0.02	**0.95 ± 0.02**	0.90 ± 0.01
SegCo$^{(4,16)}$	1.03 ± 0.04	0.05 ± 0.02	0.88 ± 0.02	0.93 ± 0.02	**0.91 ± 0.01**
PoCo Vodopivec et al. (2018)	0.90 ± 0.15	0.16 ± 0.07	0.76 ± 0.08	0.67 ± 0.05	0.71 ± 0.03
RetinaNet	**1.00 ± 0.05**	**0.05 ± 0.02**	**0.90 ± 0.03**	0.91 ± 0.03	**0.91 ± 0.01**

Table 8: Detection results on the PoCo test set. A conservative evaluation distance threshold $\tau_d = 30$ pixels was used. SegCo and RetinaNet achieve comparable results – note that due to annotation errors, precision and consequently F-1 measure differences are unreliable.

which simultaneously yield polyp centers as well as their sizes.

Several experiments were conducted to analyze the performance of SegCo. Results show that SegCo$^{(4,16)}$ which uses a shallow segmentation network with a low number of parameters, achieves similar polyp detection performance as SegCo$^{(4,64)}$, even though the latter is more complex and computationally more expensive. We identified additional difficulties in manual annotation by discovering a non negligible amount of errors in the PoCo test set (Dataset A) Vodopivec et al. (2018). The test set was revised and missing ground truth annotation were added. On average, 10% of annotations were added to each image.

We compared the proposed method SegCo with the state-of-the-art in polyp detection (PoCo Vodopivec et al. (2018)) and general object detection (RetinaNet Lin et al. (2017)). The evaluation results on the revised dataset show that SegCo outperforms both methods. The F-1 measure is improved by 24% in comparison to the PoCo detector (Vodopivec et al., 2018) and by 2% in comparison to RetinaNet (Lin et al., 2017).

Recently the use of hyperspectral imaging has gained popularity in biological image processing (Turra et al., 2015; Yoon et al., 2015; Arrigoni et al., 2017; Masschelein et al., 2012; Dumke et al., 2018). Although SegCo achieves a high accuracy on RGB images, inclusion of additional spectra is expected to increase the robustness of foreground-background distinction. Hyperspectral imaging thus presents an exciting new avenues for future work.

An easy-to-use Python application of SegCo is made publicly available [1]. We hope this will facilitate research in jellyfish bloom analysis by allowing to process vast amounts of image data, which has not been possible with the prior technology and manual annotations.

[1] `http://www.vicos.si/Projects/PoCo`

6. Acknowledgement

This work was supported by the Slovenian Research Agency (ARRS) projects J2-9433, J2-8175, Z7-1884 and program P2-0214.

References

Arrigoni, S., Turra, G., & Signoroni, A. (2017). Hyperspectral image analysis for rapid and accurate discrimination of bacterial infections: A benchmark study. *Computers in Biology and Medicine*, *88*, 60 – 71. URL: `http://www.sciencedirect.com/science/article/pii/S0010482517301828`. doi:`https://doi.org/10.1016/j.compbiomed.2017.06.018`.

Brotz, L., Cheung, W. W., Kleisner, K., Pakhomov, E., & Pauly, D. (2012). Increasing jellyfish populations: trends in large marine ecosystems. *Hydrobiologia*, *1*, 3–20.

Condon, R. H., Duarte, C. M., Pitt, K. A., Robinson, K. L., Lucas, C. H., Sutherland, K. R., Mianzan, H. W., Bogeberg, M., Purcell, J. E., Decker, M. B. et al. (2013). Recurrent jellyfish blooms are a consequence of global oscillations. *Proceedings of the National Academy of Sciences*, *110*, 1000–1005.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273–297.

Dollár, P., Belongie, S., & Perona, P. (2010). The fastest pedestrian detector in the west. *Proceedings of the British Machine Vision Conference*, *2*, 7–18.

Dumke, I., Purser, A., Marcon, Y., Nornes, S. M., Johnsen, G., Ludvigsen, M., & Søreide, F. (2018). Underwater hyperspectral imaging as an in situ taxonomic tool for deep-sea megafauna. *Scientific reports*, *8*, 12860.

Ferrari, A., Lombardi, S., & Signoroni, A. (2017). Bacterial colony counting with convolutional neural networks in digital microbiology imaging. *Pattern Recognition*, *61*, 629–640.

Foroughi, H., Ray, N., & Zhang, H. (2015). Robust people counting using sparse representation and random projection. *Pattern Recognition*, *48*, 3038–3052.

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

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

Heinrich, K., Roth, A., & Zschech, P. (2019). Everything counts: A taxonomy of deep learning approaches for object counting.

Hočevar, S., Malej, A., Boldin, B., & Purcell, J. E. (2018). Seasonal fluctuations in population dynamics of *Aurelia aurita* polyps in situ with a modelling perspective. *Marine Ecology Progress Series*, *591*, 155–166.

Jonker, R., & Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, *38*, 325–340.

Kogovšek, T., Vodopivec, M., Raicich, F., Uye, S., & Malej, A. (2018). Comparative analysis of the ecosystems in the northern adriatic sea and the inland sea of japan: Can anthropogenic pressures disclose jellyfish outbreaks? *The Science of the total environment*, *626*, 982–994.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LaTorre, A., Alonso-Nanclares, L., Muelas, S., Peña, J., & DeFelipe, J. (2013). Segmentation of neuronal nuclei based on clump splitting and a two-step binarization of images. *Expert Systems with Applications*, *40*, 6521–6530.

Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. In *Advances in neural information processing systems* (pp. 1324–1332).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 2999–3007). IEEE.

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

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*, 91–110.

Masschelein, B., Robles-Kelly, A., Blanch, C., Tack, N., Simpson-Young, B., & Lambrechts, A. (2012). Towards a colony counting system using hyperspectral imaging. In D. L. Farkas, D. V. Nicolau, & R. C. Leif (Eds.), *Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues X* (pp. 133 – 147). International Society for Optics and Photonics SPIE volume 8225. URL: `https://doi.org/10.1117/12.908041`. doi:`10.1117/12.908041`.

Perko, R., Schnabel, T., Fritz, G., Almer, A., & Paletta, L. (2013). Airborne based high performance crowd monitoring for security applications. In *Scandinavian Conference on Image Analysis* (pp. 664–674). Springer.

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

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Rosenfeld, A., & Pfaltz, J. L. (1968). Distance functions on digital pictures. *Pattern recognition*, *1*, 33–61.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*, 211–252. doi:`10.1007/s11263-015-0816-y`.

Schoening, T., Bergmann, M., Ontrup, J., Taylor, J., Dannheim, J., Gutt, J., Purser, A., & Nattkemper, T. W. (2012). Semi-automated image analysis for the assessment of megafaunal densities at the arctic deep-sea observatory hausgarten. *PloS one*, *7*, e38179.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, .

Turra, G., Conti, N., & Signoroni, A. (2015). Hyperspectral image acquisition and analysis of cultured bacteria for the discrimination of urinary tract infections. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 759–762). doi:`10.1109/EMBC.2015.7318473`.

Verikas, A., Gelzinis, A., Bacauskiene, M., Olenina, I., Olenin, S., & Vaiciukynas, E. (2012). Automated image analysis-and soft computing-based detection of the invasive dinoflagellate prorocentrum minimum (pavillard) schiller. *Expert Systems with Applications*, *39*, 6069–6077.

Vodopivec, M., Mandeljc, R., Makovec, T., Malej, A., & Kristan, M. (2018). Towards automated scyphistoma census in underwater imagery: useful research and monitoring tool. *Journal of Sea Research*, *142*, 147 – 156.

Vodopivec, M., Mandeljc, R., Malej, A., & Kristan, M. (2016). Polyp counting made easy: two stage scyphistoma detection for a computer-assisted census in underwater imagery. *Fifth International Jellyfish Bloom Symposium: Abstract book, Barcelona*, .

Widmer, C. L., Fox, C. J., & Brierley, A. S. (2016). Effects of temperature and salinity on four species of northeastern atlantic scyphistomae (cnidaria: Scyphozoa). *Marine Ecology Progress Series*, *559*, 73–88.

Xie, W., Noble, J. A., & Zisserman, A. (2018). Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, *6*, 283–292.

Yoon, S.-C., Lawrence, K. C., & Park, B. (2015). Automatic counting and classification of bacterial colonies using hyperspectral imaging. *Food and Bioprocess Technology*, *8*, 2047–2065. URL: `https://doi.org/10.1007/s11947-015-1555-3`. doi:`10.1007/s11947-015-1555-3`.