PLOS ONE

RESEARCH ARTICLE

MAIA—A machine learning assisted image annotation method for environmental monitoring and exploration

Martin Zurowietz[1]*, Daniel Langenkämper[1], Brett Hosking[2], Henry A. Ruhl[2,3], Tim W. Nattkemper[1]

1 Biodata Mining Group, Faculty of Technology, Bielefeld University, Bielefeld, Germany, **2** National Oceanography Centre, University of Southampton Waterfront Campus, Southampton, United Kingdom, **3** Monterey Bay Aquarium Research Institute, Moss Landing, California, United States of America

* martin@cebitec.uni-bielefeld.de

Abstract

Digital imaging has become one of the most important techniques in environmental monitoring and exploration. In the case of the marine environment, mobile platforms such as autonomous underwater vehicles (AUVs) are now equipped with high-resolution cameras to capture huge collections of images from the seabed. However, the timely evaluation of all these images presents a bottleneck problem as tens of thousands or more images can be collected during a single dive. This makes computational support for marine image analysis essential. Computer-aided analysis of environmental images (and marine images in particular) with machine learning algorithms is promising, but challenging and different to other imaging domains because training data and class labels cannot be collected as efficiently and comprehensively as in other areas. In this paper, we present Machine learning Assisted Image Annotation (MAIA), a new image annotation method for environmental monitoring and exploration that overcomes the obstacle of missing training data. The method uses a combination of autoencoder networks and Mask Region-based Convolutional Neural Network (Mask R-CNN), which allows human observers to annotate large image collections much faster than before. We evaluated the method with three marine image datasets featuring different types of background, imaging equipment and object classes. Using MAIA, we were able to annotate objects of interest with an average recall of 84.1% more than twice as fast as compared to "traditional" annotation methods, which are purely based on software-supported direct visual inspection and manual annotation. The speed gain increases proportionally with the size of a dataset. The MAIA approach represents a substantial improvement on the path to greater efficiency in the annotation of large benthic image collections.

Citation: Zurowietz M, Langenkämper D, Hosking B, Ruhl HA, Nattkemper TW (2018) MAIA—A machine learning assisted image annotation method for environmental monitoring and exploration. PLoS ONE 13(11): e0207498. https://doi.org/10.1371/journal.pone.0207498

Editor: Pinaki Sarder, State University of New York at Buffalo, UNITED STATES

Received: September 7, 2018

Accepted: October 31, 2018

Published: November 16, 2018

Data Availability Statement: The dataset of the results can be accessed at https://doi.org/10.5281/zenodo.1453836.

Funding: MZ and DL have received funding by Projektträger Jülich (grant no 03F0707C) under the framework of JPI Oceans. The collection and annotation of datasets used in this project was funded by the Managing Impacts of Deep-seA reSource exploitation (MIDAS) project funded by the European Union Seventh Framework Programme (FP7/2007–2013) grant agreement no.

1 Introduction

With the establishment of deep learning methods, the task of image classification seems to be solved for many computer vision scenarios, sometimes even outperforming human experts on

603418 and the European Community's ECO2 project grant agreement no. 265847. This work contributes to Natural Environment Research Council 486 "Autonomous Ecological Surveying of the Abyss" project (NERC grant NE/H021787/1) 487 and the NERC Climate Linked Atlantic Sector Science (CLASS) programme. Funding was also received from the Horizon 2020 projects EMSO-Link (No. 731036) and STEMM-CCS of the European Commission (No. 654462).

Competing interests: The authors have declared that no competing interests exist.

benchmark datasets [1]. However, automatic object detection is still below expert performance in most contexts and has gained more and more attention in recent years. Although applications for automatic object detection exist, they focus mostly on images that show everyday objects from scenes on land, neglecting the image domain of environmental monitoring data such as marine imagery.

Imaging in marine science is a noninvasive method to monitor and explore marine habitats (e.g. estimating biodiversity or ecological management). Due to advances in marine technology (marine digital imaging, autonomous underwater vehicles (AUVs), remotely operated vehicles (ROVs), etc.) vast amounts of image data can be acquired. The evaluation of these datasets is usually done by manual image annotation with dedicated tools like SQUIDLE+ (http://squidle.org) or the Bio-Image Indexing and Graphical Labelling Environment (BIIGLE) 2.0 [2]. While in computer science research areas such as media informatics or image databases, image annotation refers to the assignment of semantics to whole images, describing the content on a high level, image annotation in this context refers to the assignment of meaning (i.e. a class label selected from a given taxonomy) to a region of an image [3–5]. Even with dedicated tools, the purely manual approach to this kind of image annotation is still a time-consuming and error-prone task [3, 4]. In addition, owing to the complexity and diversity of organisms found in marine imagery, only domain experts are usually able to provide object detection and class labels with sufficient quality (i.e. sufficient inter-/intraobserver agreement and accuracy). This contrasts the situation of established computer vision benchmark image datasets like MS COCO [6], where crowdsourcing can be employed to generate large amounts of image annotations for everyday objects.

During manual image annotation, most of the time is spent on locating objects of interest (OOI) rather than assigning a correct class label for the object [3]. Moreover, OOI that were missed during image annotation are a key part of the differences in annotation quality between human observers [5]. Over the years, some methods have been proposed to assist in underwater object detection, like ranking images or image regions by saliency [7] or detecting interesting events in video streams [8]. An object detection method that efficiently and effectively generates annotation candidates (i.e. regions showing potential OOI) for marine images to increase the speed and volume of manual annotations, has yet to be developed.

In this paper, we present the Machine learning Assisted Image Annotation method (MAIA). The method aims to speed up manual image annotation by automating the process of object detection and instance segmentation of OOI in the images. Liberated from the task of object detection, human observers can concentrate purely on the classification of OOI. In the context of marine biology and environmental sciences, OOI could be particular species of interest or general megafauna like starfishes, holothurians or sponges. MAIA is based on a combination of two machine learning methods: Autoencoder networks (AEN) [9] and fully convolutional networks (FCN) [10].

Fully convolutional networks were introduced by Long et al. [10], who adapted a neural network for image classification into a fully convolutional network that performs semantic segmentation of an image. One of the more recent variants of FCNs is the Mask Region-based Convolutional Neural Network (Mask R-CNN) [11], which is capable of instance segmentation. Instance segmentation generates a bounding box, a class label and a pixel-accurate segmentation mask for each OOI in an image. Mask R-CNN can achieve impressive results, condition to the availability of a sufficient number of annotated training data samples, i.e. images with marked objects. Since the collection of training data in marine sciences is considerably expensive as it requires substantial experience and academic education, we developed MAIA to reduce time and effort on this end. In MAIA, Mask R-CNN is employed for instance segmentation and the resulting segmentation masks can be used as annotation candidates. To the best of our knowledge, we are the first to employ Mask R-CNN in the context of marine environmental monitoring and exploration.

As the supervised Mask R-CNN needs a sufficient amount of training data, we apply unsupervised AEN for novelty detection to efficiently and effectively generate training proposals, which are points of potential OOI in the images that could be considered for training a Mask R-CNN model. Based on the assumption that OOI are rare in the images, "background" pixels of the seabed can be regarded as common patterns and "interesting" pixels of objects are regarded as novel patterns. The concept of AEN was first presented by Baldi and Hornik in 1989 [9] and has been used in various contexts like dimensionality reduction [12, 13], human pose recovery [14] or cell nuclei detection [15]. AEN have been previously used for novelty detection as well [16, 17] but not in the context of FCN training data collection or environmental imaging.

The contributions of this paper can be summarized as follows: (1) We present a machine learning assisted method for image annotation that allows faster manual image annotation than methods that were used before, (2) we are the first to present the use of Mask R-CNN in the context of marine environmental monitoring and exploration, and (3) we present a detailed analysis of the manual annotation speed with three image collections featuring different types of background, imaging equipment and object classes.

In the following, MAIA is first described at the methodological level, dividing the method into four different stages to be performed one after the other (see Section 2). To assess the performance of the method regarding efficiency (i.e. time required per annotation) and accuracy (i.e. evaluation of statistics on correct and incorrect annotations), we have applied MAIA to three different datasets, referred to as JC77, SO242 and PAP, that are described in Section 3. The results of these applications are summarized in Section 4 and discussed in Section 5 regarding the dataset-specific performances and general observations. The dataset of the results can be accessed at [18]. Visual exploration of the results is possible in BIIGLE 2.0 at https://biigle.de/projects/139 using the login maia@example.com and the password `MAIApaper`. The manuscript ends with a short conclusion about the relevance of our results and the MAIA method for benthic image annotation.

2 Methods

MAIA consists of four consecutive stages (see Fig 1), all of which are described in the following sections. In Stage I, unsupervised novelty detection generates an initial set of training proposals T_p, which are image patches (i.e. regions of an image) showing patterns different from the background. These training proposals contain potential OOI that could be used in a training dataset for the instance segmentation. In Stage II, T_p is manually filtered to keep only those training proposals that actually show OOI relevant to the application context. In addition to that, the training proposals are manually refined with regard to their centroids and size, resulting in the dataset of training samples $T_s \subseteq T_p$. In Stage III, the set T_s is used to train a Mask R-CNN model for instance segmentation which is subsequently applied to produce a set A_c of annotation candidates for a whole image dataset. In Stage IV, these candidates are manually reviewed to remove false positives for the final set $A \subseteq A_c$ of detected objects and their bounding boxes.

Stage I: Novelty detection with AEN

A dataset of images $\{I_i\}$ from an AUV or ROV dive can feature different kinds of background patterns due to changes in sediment type, illumination or geological characteristics of the seabed (e.g. see SO242 in Fig 2). Thus, the images $\{I_i\}$ from one dataset are first grouped into

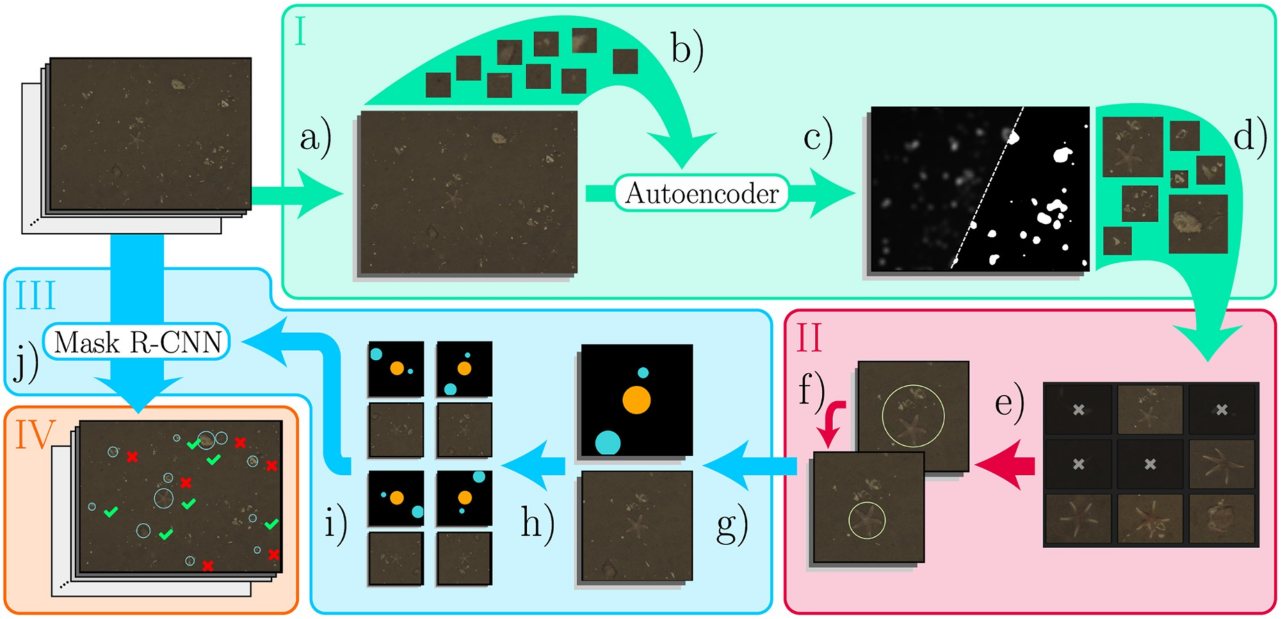

Fig 1. An overview of MAIA. Steps of the four stages are shown in different colors. a) Extraction of a random subset of the images. b) Training of the autoencoder network (AEN) with random image patches extracted from the subset. c) Application of AEN. d) Thresholding of the novelty maps and extraction of interesting regions as training proposals T_p. e) Manual filtering of T_p. f) Manual refinement of filtered T_p. g) Creation of the training dataset T_s. h) Boosting of the training dataset. i) Training of Mask R-CNN. j) Detection with Mask R-CNN on the whole image dataset and review of annotation candidates A_c.

https://doi.org/10.1371/journal.pone.0207498.g001

different clusters $\{U_k\}_{k=1,\ldots,K}$, each one representing images with a similar background. To this end, each I_i is mapped to a feature vector v_i which is used for k-means clustering of all images. The features were determined empirically as a combination of principal component projection coordinates and an entropy measure. Details are given in the supporting information (see S1 Text). Each resulting cluster U_k contains images featuring global similarities, mostly dependent on the background sediment. For each cluster, one AEN is trained, all sharing the same architecture.

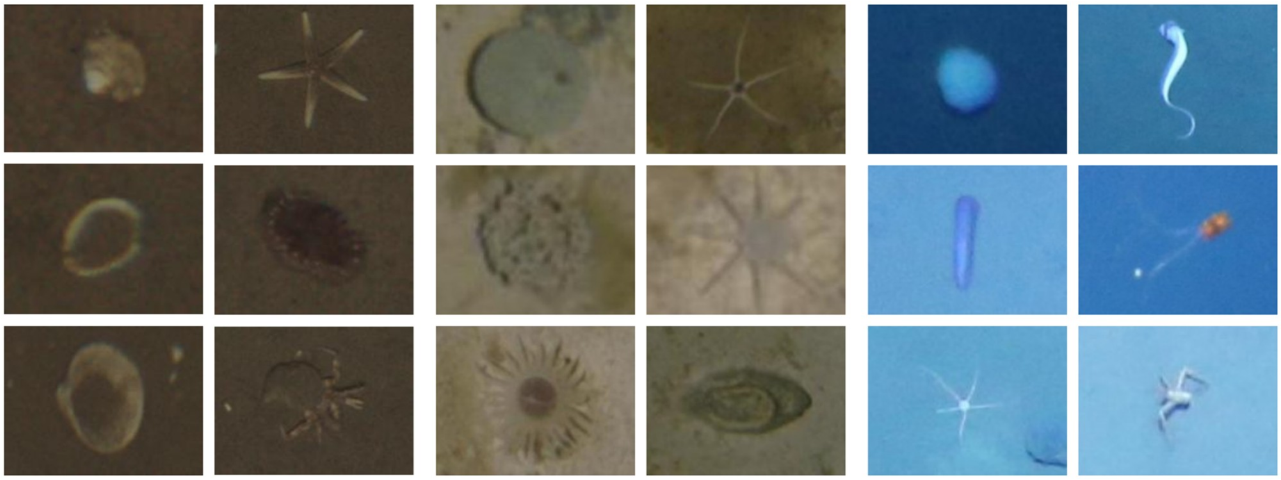

Fig 2. Examples of frequent OOI in the JC77 (columns 1-2), PAP (columns 3-4) and SO242 (columns 5-6) datasets.

https://doi.org/10.1371/journal.pone.0207498.g002

Each of the K AEN consists of three fully connected layers. From the first to the second layer (the latent layer), an input image patch $x \in \mathbb{R}^r$ is encoded to a lower dimensional latent representation $l \in \mathbb{R}^s$ with $s < r$, where W is a matrix of weights, b is a vector of biases and $\sigma(\cdot)$ is the softplus transfer function (see Eqs 1 and 3). If the RGB input image patch has a size of $r_e \times r_e$ pixels, the input layer dimension is $r = 3 \cdot r_e^2$. From the second to the third layer, the latent representation is decoded to the output patch $x' \in \mathbb{R}^r$, where W' is a matrix of weights and b' is a vector of biases (see Eq 2).

$$l = \sigma(Wx + b) \tag{1}$$

$$x' = \sigma(W'l + b') \tag{2}$$

$$\sigma(x) = \log(1 + \exp\ x) \tag{3}$$

The AEN is then trained to generate x' as a reconstruction of x by minimizing the reconstruction error $\mathcal{F}$ through backpropagation (see Eq 4).

$$\mathcal{F}(x, x') = (x - x')^2 \tag{4}$$

As motivated above, we consider image patches that show the pure seabed or background as common patterns and patches that show an interesting object (like a shell or a starfish) as novelties. This definition is based on the assumption that interesting objects are rare in deep-sea image datasets. Those rare objects would appear as novelties and patches featuring such a pattern also feature a large reconstruction error $\mathcal{F}$. By applying a threshold to the reconstruction error, background pixels can be separated from pixels that belong to an OOI.

To train the AEN for one cluster of images U_k, we sample 10^4 image patches, each of the form $x = (p_1^{(\mathfrak{r})}, p_1^{(\mathfrak{g})}, p_1^{(\mathfrak{b})}, \ldots, p_{r/3}^{(\mathfrak{b})})$ and of size $r = 39 \times 39 \times 3$ (i.e. a 39 pixel square of RGB values) randomly from U_k (see Fig 1b). The RGB values of each pixel p_j are "flattened" to the r-dimensional input x. The patch size was determined with a parameter search for good detection performance (see Section 4). We choose the dimension of the latent representation of the AEN as $s = [0.1r]$, with $[\cdot]$ as rounding operation. The "compression factor" of 0.1, again, was determined with a parameter search (see Section 4). To accelerate convergence during training, we use Xavier initialization [19] of the weights W of the encoding layer and Adam optimization [20] with a fixed learning rate of 10^{-3}. We train for 100 epochs with a minibatch size of 128. Training takes less than a minute per image cluster for all datasets, using TensorFlow [21] on a single NVIDIA Titan X. We refer to the trained AEN of an image cluster U_k with the term data-driven background model DBM$_k$.

Each of the K DBM$_k$ is applied to the images $I_i \in U_k$, processing each possible image patch of $r_e \times r_e$ pixels (see Fig 3). This can be compared to a convolution operation. We choose a stride of two for the convolution, since it had no impact on detection performance but a higher computational efficiency when compared to a stride of one. The higher stride results in an output of lower resolution than the original image, so the output is upscaled back to the original resolution of the image I_i using bilinear interpolation. Owing to GPU memory constraints, a DBM$_k$ is consecutively applied to horizontal image slices that are stitched together afterward. The reconstruction error $\mathcal{F}(x, x')$ of each image patch x is stored as the "novelty score" at the pixel coordinate of the center of the patch, resulting in the "novelty map" N_i for each image I_i. The novelty map is convolved with an all-ones kernel of the same size as the training patches, which acts as a dilation to smooth the boundaries of regions with a high novelty score (see Fig 1c).

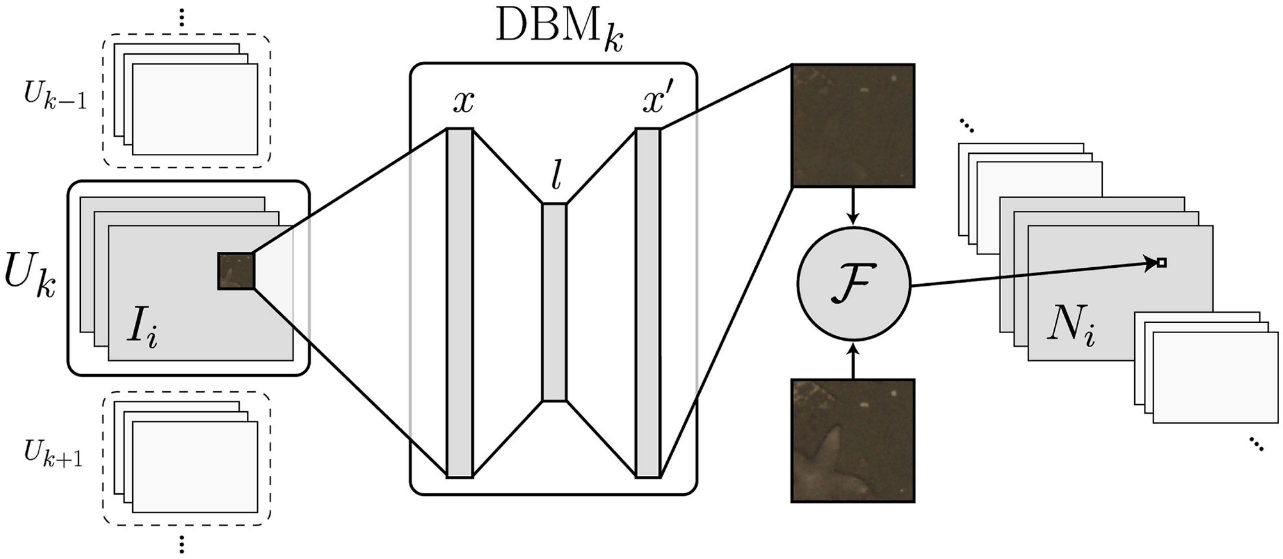

Fig 3. The data-driven background model (DBM_k) is applied like a convolution to the input image $I_i \in U_k$. The image patch is flattened to the input x. The input x is compressed to the latent representation l and then reconstructed to the output x'. The reconstruction error $\mathcal{F}(x, x')$ determines the value in the novelty map N_i at the position of each patch.

https://doi.org/10.1371/journal.pone.0207498.g003

To get a binary segmentation between background and interesting regions, a threshold has to be applied to the novelty map. We empirically determined the threshold t_k to be the mean of the 99-percentiles of the novelty maps of an image cluster U_k (see Eq 5), where $\mathrm{P}_{99}(N_i)$ is the operation that determines the 99-percentile of a novelty map N_i. Examples of thresholded novelty maps can be found in Fig 4.

$$t_k = \frac{1}{|U_k|} \sum_{i:I_i \in U_k} \mathrm{P}_{99}(N_i) \tag{5}$$

The binary segmentation of all novelty maps N_i is used to extract the set of training proposals T_p from all images. Each training proposal $y_j \in T_p$ is a part of an image, cropped to the minimum square bounding box of an interesting region (see Fig 1d). We set the minimum edge length of the bounding box to 30 pixels to ensure a certain size of the training proposals, even if the interesting region is smaller. We define the cumulated novelty scores of the pixels of an interesting region as the novelty score $\eta(y_j)$ of the corresponding training proposal.

Stage II: Manual filtering and refinement of training proposals

A considerable amount of the training proposals in T_p may show patterns that do not represent anything of interest for the domain experts. Thus, we apply a quick manual filtering step, where a human observer selects only those training proposals that contain OOI. The filtering has been implemented as single patch classification as defined in the RecoMIA guidelines [3], so only the isolated image regions of the training proposals are displayed to the human observer instead of the complete images. This saves the time the human observer would need to screen the complete images for multiple regions of interest. For each training proposal, the human observer has to determine if it contains (part of) an OOI or not. To accelerate manual filtering, the label review grid overview tool Largo of BIIGLE 2.0 is used (see S1 Fig in the supporting information). Training proposals y_j are displayed in a regular

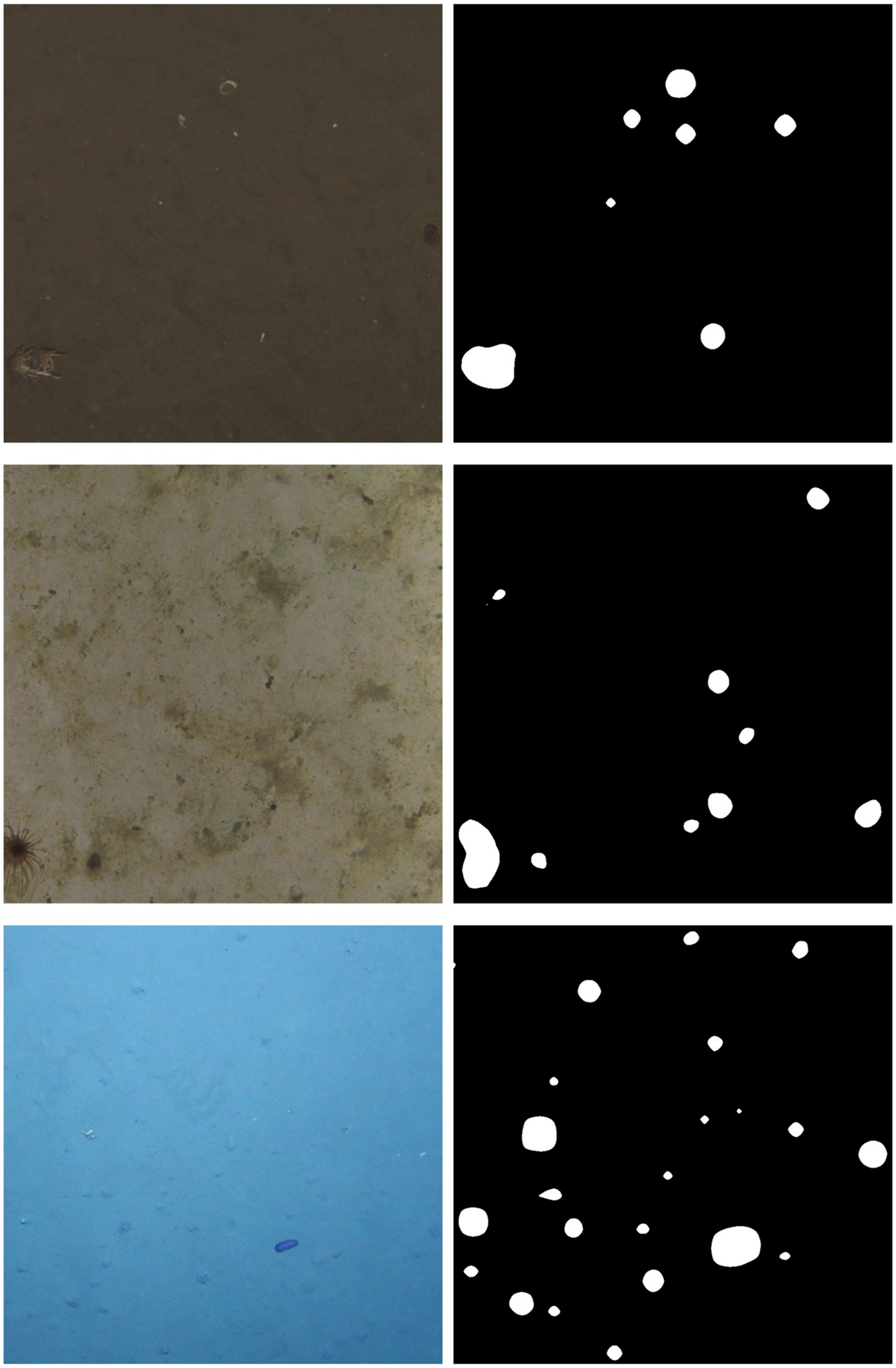

Fig 4. Crops of example images I_i (left column) and their thresholded novelty maps N_i (right column) for the three image datasets JC77 (top row), PAP (middle row) and SO242 (bottom row).

https://doi.org/10.1371/journal.pone.0207498.g004

grid in descending order of their novelty scores $\eta(y_j)$. This allows human observers to spot OOI very quickly and to review a large number of training proposals in a very short time (see Fig 1e). The sorting by novelty score is a similar technique than the saliency ranking described by [7]. Starting from the training proposals with the highest novelty score, a human observer can stop reviewing once a sufficient amount of training proposals have been selected. In this work we define 600 as the limit for the required number of selected training proposals for each object class.

The performance of an FCN-based instance segmentation method like Mask R-CNN is crucially dependent on the quality of the training dataset. If the samples in the training dataset are of low quality, i.e. with many discrepancies between interesting and non-interesting image regions, the performance of instance segmentation may be very poor. To obtain a set T_s with training samples of appropriate quality from the training proposals T_p, a manual refinement step is performed after the filtering. The filtered training proposals are shown to a human observer, each with a suggested centroid and size (i.e. a circle) that marks the OOI. The observer can modify the circle position or size so it closely fits the position and size of the OOI (see Fig 1f). To further accelerate the refinement, we use the volume label review tool Volare of BIIGLE 2.0 (see S2 Fig in the supporting information). With Volare, the viewport of the annotation tool jumps directly from one circle to the next, saving the time a human observer would need to look for and zoom in to each circle on an image.

Stage III: Instance segmentation with Mask R-CNN

The filtered and refined training proposals are used to build a dataset of training samples T_s for Mask R-CNN (see Fig 1g). This training dataset differs in fundamental aspects from datasets that are typically generated to train FCNs for instance segmentation. In the ideal case, all OOI are marked in all images of the training dataset. In our case, some OOI may have been missed in the novelty detection stage, so these would be falsely labeled as background. To reduce the probability of encountering such falsely labeled OOI during training, we crop a 500 × 500 pixel region around each filtered and refined training proposal y_j and use these crops instead of the whole images as training samples $\chi_j \in T_s$. Another difference are the circle shapes that are used for fast manual refinement. In a typical training dataset, the annotated segmentation of an OOI is not limited to a particular shape but aims to segment the object as accurately as possible. Here, we define all pixels inside the circle to belong to the OOI, which may include some pixels that actually do not belong to the OOI.

Due to the fixed number of images that are considered to generate a training dataset, there may be only a few hundred or less samples of a particular class of OOI. For comparison, in datasets like MS COCO [6] there are many thousands of instances for each object class. To increase the number of object instances that are available for training, we boost the training dataset (see Fig 1h). Details on the boosting we apply can be found in the supporting information (see S2 Text).

We utilize a freely available TensorFlow implementation of Mask R-CNN [22] in the archived version of [23]. This implementation differs in a few aspects from the original paper [11]. Input images are resized to support training in minibatches, training bounding boxes are generated on the fly and the learning rate is reduced to 10^{-3} (see [23] for details). In MAIA, the training is performed with the set of boosted training samples (see Fig 1i) and the default configuration of the Mask R-CNN implementation. The model is initialized with weights from pre-training with MS COCO [6]. First, all layers except the ResNet 101 backbone [24] are trained for 20 epochs with a learning rate of 10^{-3}, a batch size of 2 and 10^3 batches per epoch. Then all layers including the backbone are trained for another 10 epochs with a reduced learning rate of 10^{-4} but the same batch size and number of batches per epoch. With this configuration, training takes about eleven hours per dataset on a single NVIDIA Titan X.

Inference is performed by padding each image with zeros so that each dimension is divisible by 64. This is done to guarantee smooth scaling in the six levels of the Feature Pyramid Network [25] and still process the image in its original size. Based on the segmentation masks produced by Mask R-CNN, we define any pixel not belonging to the background class as "interesting". We take the minimum enclosing circle for each region of connected interesting pixels to get the set A_c of annotation candidates (see Fig 1j).

Stage IV: Annotation candidate review

To eliminate the false positive detections, which mark regions of the images that show no OOI, the annotation candidates A_c are manually reviewed in the last step. This is done analogously to the manual filtering of training proposals in Stage II. Each annotation candidate is shown as an image patch in a regular grid. A human observer then selects all candidates that are true positives, i.e. those that mark an OOI. This yields the final set A of annotations.

3 Datasets

We evaluated MAIA on three marine image datasets that were collected in different research projects. From each dataset $\Gamma \in \{\text{JC77, PAP, SO242}\}$, we extracted 500 random images as training subset T^{Γ}. For each subset, MAIA was performed to train a Mask R-CNN model. The detection performance of the model was evaluated on another 50 random images as validation subset V^{Γ} for each dataset. The images of the validation subset have been fully annotated using "traditional" methods.

JC77 dataset

The images of the first dataset were captured in the Central North Sea near the Sleipner CO2 storage site [26]. It comprises 6321 images of size 2448 × 2048 pixels (see Fig 5a). The images were captured at a depth of 77 m with a target distance to the seabed of 3.0 m. Traditional expert annotation took 166 min for a subset of 125 images, which are 79.68 $\text{s} \cdot \text{image}^{-1}$. The two different object classes "shell" (see Fig 2 column 1) and "animal" (see Fig 2 column 2) were annotated with a total of 278 annotations in the 50 images of the validation subset V^{JC77}.

PAP dataset

The second dataset contains images from the Porcupine Abyssal Plain (PAP), located southwest of the UK in international waters [27]. The dataset is composed of 3708 annotated images, each of which is a mosaic of ten single images. The images were captured in 4600-4900 m depth with a target distance to the seabed of 3.2 m. 57 different object classes were annotated by experts (for examples see Fig 2). Traditional expert annotation took 480 min for a subset of 243 images, which are 118.52 $\text{s} \cdot \text{image}^{-1}$. To eliminate the black background and mosaicing artifacts (e.g. the "sawtooth" boundary between images and background [27]), we extracted tiles of 1000 × 1000 pixels that contain no black background from the mosaics (see Fig 5b). The result is a dataset of 36238 tiles (9.77 $\text{tiles} \cdot \text{image}^{-1}$). The original mosaics contain a total of 33358 annotations. Owing to overlaps during tile extraction, the final dataset contains 41033 annotations. The 50 images of the validation subset V^{PAP} were chosen so that no overlapping tiles between training and validation subsets exist. The images of V^{PAP} contain a total of 57 annotations.

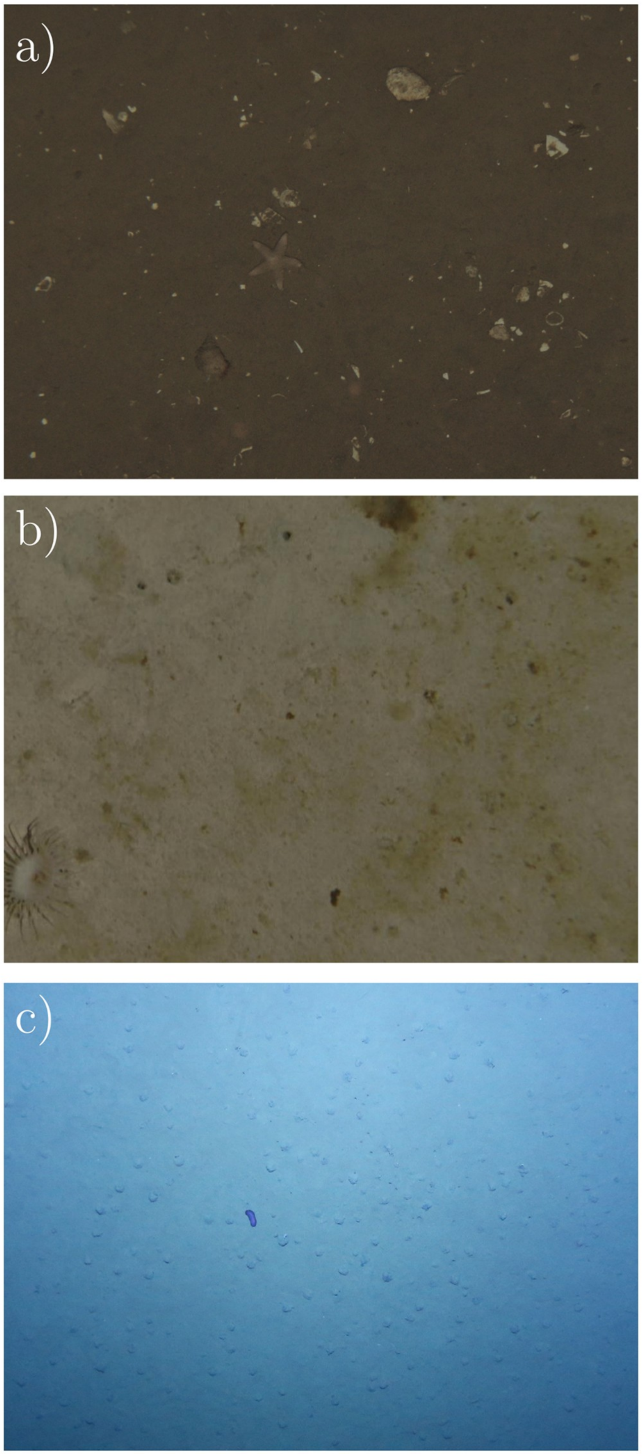

Fig 5. Crops of example images of the JC77 (a), PAP (b) and SO242 (c) datasets. Note the rocks and scattered shell fragments that are no OOI in JC77 (a) and the irregular illumination in SO242 (c).

https://doi.org/10.1371/journal.pone.0207498.g005

SO242 dataset

The third dataset consists of 809 images with a size of 4096 × 3072 pixels that were extracted from the SO242/1_83-1_AUV10 survey [28] (see Fig 5c). The images of the survey were captured at depths between 3420 m and 4140 m with a target distance to the seabed of 7.5 m. Traditional expert annotation took 87 min for a subset of 75 images, which are 69.6 $\mathrm{s} \cdot \mathrm{image}^{-1}$. The 50 images of the validation subset V^{SO242} contain a total of 350 annotations. Only a single "interesting" object class was used to annotate all megafauna in the images of V^{SO242}.

4 Results

Here we present the results of the evaluation of the MAIA stages I-IV. First, we show the detection performance of the unsupervised novelty detection using different AEN parameters. Next, we present the timings of the manual filtering and training proposal refinement steps. Third, we show the detection performance of the trained Mask R-CNN model on the validation subset V^{Γ} of each dataset. Finally, we determine the average time it took to annotate an image using MAIA compared to a "traditional" annotation method which employs a sophisticated annotation tool like BIIGLE 2.0 but is purely manual.

Stage I: Novelty detection with AEN

The AEN, as well as Mask R-CNN, produces a pixel-wise segmentation between background and interesting regions for each image. We evaluated the segmentation with **recall** = $TP_\theta(TP_\theta + FN_\theta)^{-1}$, ($TP_\theta$: number of OOI contained in interesting regions, FN_θ: number of OOI not contained in an interesting region) and **precision** = $TP_\rho(TP_\rho + FP_\rho)^{-1}$, ($TP_\rho$: number of interesting regions containing an OOI, FP_ρ: number of interesting regions not containing an OOI). From these values we determined the F_2-score [29] for the detection performance (see Eq 6). We chose the F_2-score to put a higher weight on the recall as this was considered reasonable in the research context at hand.

$$F_2 = \frac{5 \cdot \text{recall} \cdot \text{precision}}{(4 \cdot \text{precision}) + \text{recall}} \tag{6}$$

A parameter search was conducted for the number of clusters $K \in \{1, 5, 10, 50, 100\}$, input patch dimension $r_e \in \{29, 39\}$ and latent layer compression factor $s_c = s/r \in \{0.1, 0.2\}$ of the AEN. For each parameter triplet, we determined the interesting regions of the novelty maps $\{N_i\}$ and calculated the F_2-score for the validation subset V^{Γ} of each image dataset. The F_2-scores are visualized in Fig 6 for all three datasets. Detailed values can be found in the supporting information (see S3 Text). On average, the scores of the parameter triplets with $r_e = 39$ are higher than those with $r_e = 29$ and a cluster number of $K = 5$ is a good selection for all datasets. The scores of the parameter triplets with a larger s_c do not differ much from those with a smaller s_c and the same r_e but a smaller s_c has a lower computational cost.

Stage II: Manual filtering and refinement of training proposals

Novelty detection on the 500 images of the JC77 training subset T^{JC77} produced a total of 18826 training proposals. These were manually filtered in 61.02 min and refined in 60.17 min which resulted in 440 training samples for the "animal" class and 600 training samples for the "shell" class. Total manual processing of the images of T^{JC77} took 14.54 $\mathrm{s} \cdot \mathrm{image}^{-1}$ on average.

In the case of the 500 tiles of T^{PAP}, 2399 training proposals were produced by novelty detection. These were manually filtered in 23.88 min and refined in 22.20 min which resulted in a total of 200 training samples T_s^{PAP}. We decided to use the single class "interesting" for the

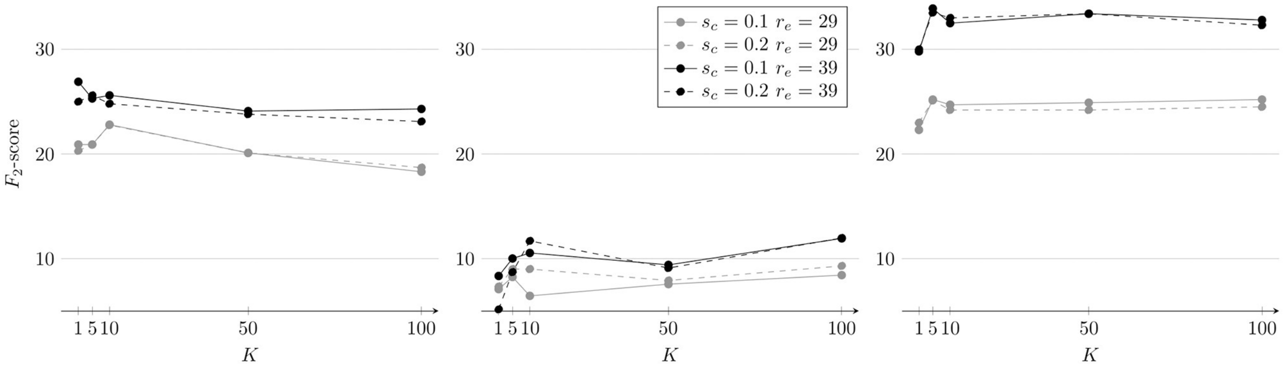

Fig 6. The F_2-scores of each parameter triplet of the AEN parameter search for V^{JC77} (left), V^{PAP} (center) and V^{SO242} (right).

https://doi.org/10.1371/journal.pone.0207498.g006

training samples of this dataset, since the number of training samples was too low for a finer taxonomic differentiation. Total manual processing of the images of T^{PAP} took 5.53 s · tile^{-1} which is extrapolated to 54.03 s · image^{-1} using the ratio of 9.77 tiles · image^{-1}.

For the images of T^{SO242}, novelty detection produced 30783 training proposals. Manual filtering took only 19.42 min because the limit of 600 training proposals was quickly reached. Refinement of the 600 training proposals took 25.53 min. Total manual processing of the images of T^{SO242} took 5.39 s · image^{-1}.

For all datasets, manual filtering and refinement took 24.65 s · image^{-1} on average, or 3.42 h for the 500 images of a training subset T^{Γ}.

The training sample datasets have a size of $|T_s^{\mathrm{JC77}}| = 1040$, $|T_s^{\mathrm{PAP}}| = 200$ and $|T_s^{\mathrm{SO242}}| = 600$. However, with the boosting applied, $6 \cdot 10^4$ unique training samples were used during the $3 \cdot 10^4$ training steps with batch size 2 for the Mask R-CNN model of each dataset.

Stage III: Instance segmentation with Mask R-CNN

The binary segmentation between background and interesting pixels produced by Mask R-CNN was evaluated in the same way as the novelty detection in Stage I. For each validation subset V^{Γ}, precision, recall and F_2-scores were calculated. The results are shown in Table 1. The values for V^{JC77} are highest, followed by those for V^{SO242}. The precision for V^{SO242} is almost only half of the precision for V^{JC77}. The values for V^{PAP} are lowest with a precision less than half of the precision for V^{JC77}. Overall, the instance segmentation achieved an average recall of 84.1% and an average precision of 30.3%. Example annotation candidates for each dataset are shown in Fig 7.

Stage IV: Annotation candidate review

The trained Mask R-CNN model produced 308 to 1094 annotation candidates A_c^{Γ} for the 50 images of each validation subset V^{Γ} (see Table 2). These were reviewed in 2.35 to 16.70 min with an average review time of 18.34 s · image^{-1}. The review time of A_c^{PAP} has been extrapolated

Table 1. Detection performance of the trained Mask R-CNN model on each validation subset V^{Γ}.

Dataset Γ	F_2-score	recall	precision	$\|T_s^{\Gamma}\|$
JC77	78.8	91.7	50.5	1040
PAP	41.1	77.2	14.3	200
SO242	58.0	83.4	26.1	600

https://doi.org/10.1371/journal.pone.0207498.t001

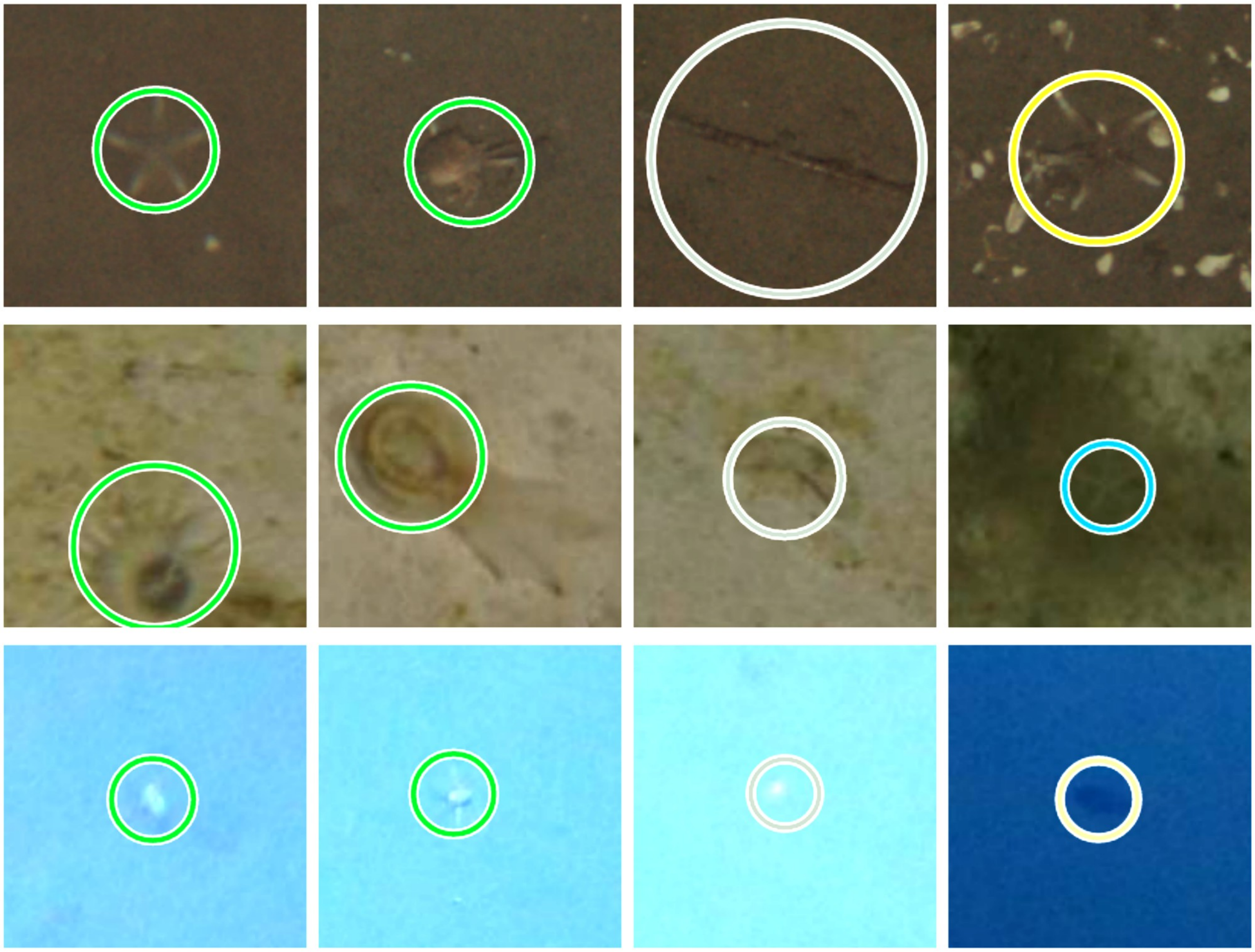

Fig 7. Example annotation candidates (column 1-2 true positives, column 3 false positives) from A_c^{JC77} (top row), A_c^{PAP} (middle row) and A_c^{SO242} (bottom row). Column 4 shows false negatives from the respective validation subset V^{Γ}.

https://doi.org/10.1371/journal.pone.0207498.g007

from 2.82 s · tile^{-1} to 27.55 s · image^{-1}. The average review time and the 3.42 h it takes to manually filter and refine training proposals of 500 training images allowed us to build a function $\tau_{\mathrm{MAIA}}(n)$ for the time it takes to annotate n images using MAIA (see Eq 7). Likewise, we were able to build a function $\tau_{\mathrm{trad}}(n)$ for the time it takes to annotate n images using a traditional method (see Eq 8). This function is based on the average annotation time of 89.27 s · image^{-1} that was measured for the three image datasets.

$$\tau_{\mathrm{MAIA}}(n) \quad = \quad 18.34\ \mathrm{s} \cdot \mathrm{image}^{-1} \cdot n + 12327\ \mathrm{s} \tag{7}$$

Table 2. Number of total annotation candidates A_c^{Γ}, manually selected true positive candidates A^{Γ} and the time it took to review them for each dataset. The review time in s · image^{-1} for A_c^{PAP} has been extrapolated.

Dataset Γ	$\lvert A_c^{\Gamma}\rvert$	$\lvert A^{\Gamma}\rvert$	Review time (min)	Review time (s · image^{-1})	Review time (s · candidate^{-1})
JC77	501	276	6.20	7.44	0.74
PAP	308	54	2.35	27.55	0.46
SO242	1094	257	16.70	20.04	0.92

https://doi.org/10.1371/journal.pone.0207498.t002

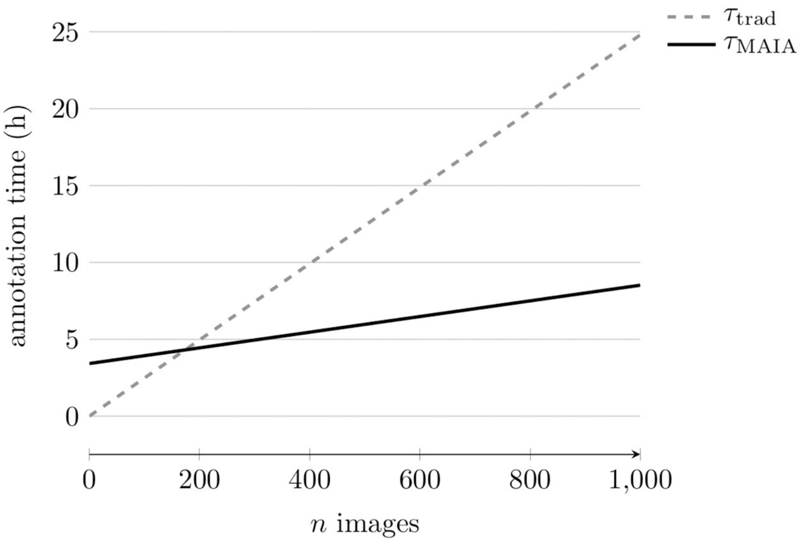

Fig 8. Time it takes to annotate images with a traditional annotation method (τ_{trad}) compared to the time it takes using MAIA (τ_{MAIA}).

https://doi.org/10.1371/journal.pone.0207498.g008

$$\tau_{\text{trad}}(n) \quad = \quad 89.27 \text{ s} \cdot \text{image}^{-1} \cdot n \tag{8}$$

A plot of the two functions shows that image annotation takes less time with MAIA than with a traditional method when the dataset contains more than 200 images (see Fig 8). The speed-up increases with increasing dataset size.

5 Discussion

The results are discussed in the same order as MAIA is described, starting with the AEN-based novelty detection. The results of the AEN parameter search indicate that clustering the images by similar background indeed improves the novelty detection performance. Novelty detection with $K = 5$ clusters yields on average considerably higher F_2-scores than with $K = 1$. However, a higher number of clusters does not further increase the scores. A larger latent layer size, which should enable the AEN to learn a higher number or more accurate image patch properties has no notable effect on the novelty detection performance. A compression factor of s_c = 0.1 seems to be sufficient to learn the visual properties of the seabed which can be briefly described by a dominant color and varying brightness (see Fig 2). The larger input patch size of 39 × 39 pixels resulted in superior F_2-scores compared to the results of the smaller patch size. While it is possible that still larger input patch sizes may further increase the scores, we judged a more exhaustive parameter search to be out of scope of this evaluation. We based this decision on our observations in other scenarios where larger window sizes decrease the performance of machine learning based classification [30]. The novelty detection performance with the presented parameter triplet already resulted in a sufficient number and quality of training proposals to be useful in the annotation method.

Manual filtering and refinement proved to be efficient with the implementation as single patch classification and the available tools of BIIGLE 2.0. Although many thousands of training proposals may sound like an overwhelming number, they can be processed much faster with an average of 24.65 s · image^{-1} than traditional image annotation with 89.27 s · image^{-1}. Contributing to this speed-up is the ordering of training proposals by novelty scores. Because of the ordering, manual processing of T_p^{SO242} was fastest, although its total number of training proposals was highest among all three training subsets T^{Γ}.

The size of the set of training samples T_s^{PAP} is small when compared to the number of training samples collected for the other datasets. While a recall of over 77% is still obtained in the final instance segmentation, a higher number of training samples would probably yield better results. This would require more than the 500 tiles of T^{PAP} to generate a higher number of training proposals and by this more training samples. Hence, the number of images in a training subset T^{Γ} should not be fixed in a future application of MAIA. It should rather be extensible on the fly with new images being added if not enough training samples have been collected.

The Mask R-CNN instance segmentation achieved an average recall of over 84% for the three evaluated datasets. Even with the PAP dataset, where only $|T_s^{\text{PAP}}| = 200$ training samples were collected, a recall of more than 77% was reached. This is a surprising result for a deep learning model like Mask R-CNN, especially considering the high variability of 57 object classes that are present in the dataset. The fact that a recall of over 77% is still achieved could be a result of the boosting applied to the training samples. The average precision of 30% may seem rather low when compared to other scenarios. In this case, however, a high number of false positive detections has a low cost regarding the review time of the annotation candidates A_c. False positives can be dismissed quickly with the Largo tool.

In contrast to image classification or information retrieval in other domains (such as medicine) there is no clear definition of thresholds for a precision or recall to be considered acceptable in marine imaging. However, we can give a heuristic recommendation based on experience. In our experiments, the results were inspected with the Largo tool in a grid overview of 50-100 image patches at the same time. There we considered it helpful if the display of such a set of image patches featured at least five true positives, just to keep and update the visual model(s) for the OOI. Thus, we would argue that 10% is the minimum value acceptable for precision, 10-50% may be considered good and more than 50% may be classified as very good. The recall values should be discussed more conservative as missed OOI can not be compensated as easily as false positive detections. However, it depends on the actual research context of the data collection how to define thresholds there.

The per-instance class labels and bounding boxes that are also produced by Mask R-CNN are ignored which reduces the output to a mere object detection of a single "interesting" class. While this is sufficient for the image annotation method here, one might argue that Mask R-CNN is not the right tool for this task. Hence, we also evaluated a minimized variant of Mask R-CNN that is "cut off" after the region proposal network, omitting the subsequent classification, bounding box refinement and segmentation mask branches. Details can be found in the supporting information (see S4 Text). Although we suspect that the low number of training samples is not sufficient for accurate classification, an evaluation of the class label output of Mask R-CNN remains a topic for future research.

The evaluation of the annotation candidate review points out different characteristics of the image datasets. The images of SO242 have the highest resolution and, with a target distance to the seabed of 7.5 m, the largest footprint of visible objects. Hence, the same number of images can contain a larger number of potential OOI than images with a smaller resolution taken closer to the ground. Combined with the rather low precision of Mask R-CNN for SO242, this contributes to the much higher number of annotation candidates in A_c^{SO242} when compared to the other two datasets. In addition to that, the review time in $\text{s} \cdot \text{candidate}^{-1}$ for SO242 is highest among the three datasets. A possible reason for this, again, might be the larger distance to the seabed since smaller objects are harder to recognize.

The review time in $\mathrm{s} \cdot \mathrm{candidate}^{-1}$ for A_c^{PAP} is much lower than for A_c^{JC77} although the images of both datasets were taken at a comparable distance to the seabed. This can be explained by differences in object classes between the datasets. Instances of the "shell" class of JC77 (see Fig 2 first column) can sometimes be hardly distinguished from rocks or shell fragments which do not belong to this class. Furthermore, some instances of the "animal" class (see Fig 2 second column) are rather well camouflaged. This is not the case for the majority of OOI in PAP.

The heterogeneity in different properties of the three datasets provides a solid basis to evaluate MAIA. Even the highest review time in $\mathrm{s} \cdot \mathrm{image}^{-1}$ stays well below the time that was determined for a traditional annotation method. The annotation time functions τ_{MAIA} and τ_{trad} show that the constant time it takes to prepare a training dataset for Mask R-CNN is quickly made up for by the much faster process of annotation candidate review when compared to manual scanning of whole images for OOI. For datasets containing 200 or more images, MAIA starts to be faster than a traditional annotation method. In case of the datasets with 550 images that were used in this evaluation MAIA is 2.19 times faster.

An interesting alternative to a visual evaluation of the Mask R-CNN results may be to classify those with another convolutional neural network. Recently, we were able to show that even the problem of strong class imbalance can be solved given the right training and data preparation methods [30].

6 Conclusion

We presented MAIA, a novel machine learning assisted method for image annotation in environmental monitoring and exploration. MAIA requires a reduced amount of manual interactions when compared to traditional annotation methods. We have used BIIGLE 2.0 in this work as the interactions required by MAIA can be performed very efficiently with this system. For datasets with more than 200 images, MAIA offers a faster annotation speed with an average recall of 84.1% when compared to traditional methods. The speed-up increases with the size of a dataset and already reaches a factor of 2.19 with datasets of 550 images that were used in our evaluation. Based on these results, we conclude that MAIA is a promising method for image annotation in all environmental monitoring and exploration scenarios with large image collections.

Supporting information

S1 Text. Image clustering. A description of the image clustering that is done prior to the training of the the AEN.
(PDF)

S2 Text. Boosting of training samples. The image transformations that are applied during boosting of the Mask R-CNN training samples.
(PDF)

S3 Text. AEN parameter search. Detailed results of the AEN parameter search.
(PDF)

S4 Text. Mask Region Proposal Network. A description of the architecture, performance evaluation and discussion of the Mask Region Proposal Network (Mask RPN).
(PDF)

S1 Fig. The label review grid overview tool Largo of BIIGLE 2.0. Image patches of training proposals or annotation candidates are displayed in a regular grid. Human observers can

quickly scroll through all image patches and select those that are of interest with a mouse click.
(TIF)

S2 Fig. The volume label review tool Volare of BIIGLE 2.0. The currently focused circle annotation is highlighted at the center of the vieport. Once the human observer finished manipulating the position and size of the circle, the viewport automatically jumps to the next circle.
(TIF)

Acknowledgments

We thank NVIDIA Corporation for donating the GPU used in this project.

Author Contributions

Conceptualization: Daniel Langenkämper, Tim W. Nattkemper.

Data curation: Martin Zurowietz, Brett Hosking, Henry A. Ruhl.

Formal analysis: Martin Zurowietz.

Funding acquisition: Henry A. Ruhl, Tim W. Nattkemper.

Investigation: Martin Zurowietz.

Methodology: Martin Zurowietz, Daniel Langenkämper.

Project administration: Martin Zurowietz.

Resources: Brett Hosking, Henry A. Ruhl, Tim W. Nattkemper.

Software: Martin Zurowietz.

Supervision: Tim W. Nattkemper.

Validation: Martin Zurowietz, Daniel Langenkämper, Brett Hosking, Henry A. Ruhl, Tim W. Nattkemper.

Visualization: Martin Zurowietz.

Writing – original draft: Martin Zurowietz, Daniel Langenkämper, Brett Hosking, Henry A. Ruhl, Tim W. Nattkemper.

References

1. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. of the IEEE ICCV; 2015. p. 1026–34.
2. Langenkämper D, Zurowietz M, Schoening T, Nattkemper TW. BIIGLE 2.0—Browsing and Annotating Large Marine Image Collections. Frontiers in Marine Science. 2017; 4:83.
3. Schoening T, Osterloff J, Nattkemper TW. RecoMIA—Recommendations for Marine Image Annotation: Lessons Learned and Future Directions. Frontiers in Marine Science. 2016; 3:59. https://doi.org/10.3389/fmars.2016.00059
4. Monk J, Barrett NS, Peel D, Lawrence E, Hill NA, Lucieer V, et al. An evaluation of the error and uncertainty in epibenthos cover estimates from AUV images collected with an efficient, spatially-balanced design. PloS one. 2018; 13(9):e0203827. https://doi.org/10.1371/journal.pone.0203827 PMID: 30226871
5. Durden JM, Bett BJ, Schoening T, Morris KJ, Nattkemper TW, Ruhl HA. Comparison of image annotation data generated by multiple investigators for benthic ecology. Marine Ecology Progress Series. 2016; 552:61–70. https://doi.org/10.3354/meps11775

6. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common objects in context. In: ECCV. Springer; 2014. p. 740–55.

7. Johnson-Roberson M, Pizarro O, Williams S. Saliency ranking for benthic survey using underwater images. In: Control Automation Robotics & Vision (ICARCV), 2010 11th Int. Conf. on. IEEE; 2010. p. 459–66.

8. Edgington DR, Salamy KA, Risi M, Sherlock R, Walther D, Koch C. Automated event detection in underwater video. In: OCEANS 2003. Proc. vol. 5. IEEE; 2003. p. 2749–53.

9. Baldi P, Hornik K. Neural networks and principal component analysis: Learning from examples without local minima. Neural networks. 1989; 2(1):53–58. https://doi.org/10.1016/0893-6080(89)90014-2

10. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proc. of the IEEE CVPR; 2015. p. 3431–40.

11. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. arXiv preprint arXiv:170306870. 2017;.

12. Zhang J, Yu J, Tao D. Local deep-feature alignment for unsupervised dimension reduction. IEEE Transactions on Image Processing. 2018; 27(5):2420–2432. https://doi.org/10.1109/TIP.2018.2804218

13. Thomas SA, Race AM, Steven RT, Gilmore IS, Bunch J. Dimensionality reduction of mass spectrometry imaging data using autoencoders. In: Computational Intelligence (SSCI), 2016 IEEE Symposium Series on. IEEE; 2016. p. 1–7.

14. Hong C, Yu J, Wan J, Tao D, Wang M. Multimodal deep autoencoder for human pose recovery. IEEE Transactions on Image Processing. 2015; 24(12):5659–5670. https://doi.org/10.1109/TIP.2015.2487860 PMID: 26452284

15. Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J, et al. Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE transactions on medical imaging. 2016; 35 (1):119–130. https://doi.org/10.1109/TMI.2015.2458702 PMID: 26208307

16. Thompson BB, Marks RJ, Choi JJ, El-Sharkawi MA, Huang MY, Bunje C. Implicit learning in autoencoder novelty assessment. In: Proc. of the 2002 IJCNN. vol. 3. IEEE; 2002. p. 2878–83.

17. Wei Q, Ren Y, Hou R, Shi B, Lo JY, Carin L. Anomaly detection for medical images based on a one-class classification. In: Medical Imaging 2018: Computer-Aided Diagnosis. vol. 10575. International Society for Optics and Photonics; 2018. p. 105751M.

18. Zurowietz M, Langenkämper D, Hosking B, Ruhl HA, Nattkemper TW. Data for the evaluation of the MAIA method for image annotation; 2018. Available from: https://doi.org/10.5281/zenodo.1453836.

19. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proc. of the 13th Int. Conf. on Artificial Intelligence and Statistics; 2010. p. 249–56.

20. Kingma D, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.

21. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:160304467. 2016;.

22. Abdulla W. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow; 2017. https://github.com/matterport/Mask_RCNN.

23. Waleed, Ferriere P, Borovec J, Pruce C, JNingWei, PaulChongPeng, et al. mzur/Mask_RCNN v2.1-doi2; 2018. Available from: https://doi.org/10.5281/zenodo.1312886.

24. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.

25. Lin TY, Dollár P, Girshick RB, He K, Hariharan B, Belongie SJ. Feature Pyramid Networks for Object Detection. In: CVPR. vol. 1; 2017. p. 4.

26. Lichtschlag A, Cevatoglu M, Connelly D, James R, Bull J. Increased fluid flow activity in shallow sediments at the 3 km Long Hugin Fracture in the central North Sea. Geochemistry, Geophysics, Geosystems. 2018; 19(1):2–20. https://doi.org/10.1002/2017GC007181

27. Morris KJ, Bett BJ, Durden JM, Huvenne VA, Milligan R, Jones DO, et al. A new method for ecological surveying of the abyss using autonomous underwater vehicle photography. Limnology and Oceanography: Methods. 2014; 12(11):795–809.

28. Greinert J, Schoening T, Köser K, Rothenbeck M. Seafloor images and raw context data along AUV track SO242/1_83-1_AUV10 (Abyss_196) during SONNE cruise SO242/1. PANGAEA; 2017.

29. Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Information Processing & Management. 2009; 45(4):427–437. https://doi.org/10.1016/j.ipm.2009.03.002

30. Langenkämper D, van Kevelaer R, Nattkemper TW. Strategies for Tackling the Class Imbalance Problem in Marine Image Classification. In: ICPR Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI). IEEE; 2018.