Semantic versus Instance Segmentation in Microscopic Algae Detection

Jesus Ruiz-Santaquiteria[a], Gloria Bueno[a], Oscar Deniz[a], Noelia Vallez[a], Gabriel Cristobal[b]

[a] *University of Castilla-La Mancha, ETSI Industriales, Visilab, Ciudad Real, Spain*
[b] *Institute of Optics "Daza de Valdés", Spanish National Research Council (CSIC), Madrid, Spain*

Abstract

Microscopic algae segmentation, specifically of diatoms, is an essential procedure for water quality assessment. The segmentation of these microalgae is still a challenge for computer vision. This paper addresses for the first time this problem using deep learning approaches to predict exactly those pixels that belong to each class, i.e., diatom and non diatom. A comparison between semantic segmentation and instance segmentation is carried out, and the performance of these methods is evaluated in the presence of different types of noise. The trained models are then evaluated with the same raw images used for manual diatom identification. A total of 126 images of the entire field of view at 60x magnification, with a size of 2592x1944 pixels, are analyzed. The images contain 10 different taxa plus debris and fragments. The best results were obtained with instance segmentation achieving an average precision of 0.85% with 0.86% sensitivity and 0.91% specificity (up to 0.92%precision with 0.98%, both sensitivity and specificity for some taxa). Semantic segmentation was able to improve the average sensitivity up to 0.95% but decreasing the specificity down to 0.60% and precision to 0.57%. Instance segmentation was also able to properly separate diatoms when overlap occurs, which helps estimate the number of diatoms, a key requirement for water quality grading.

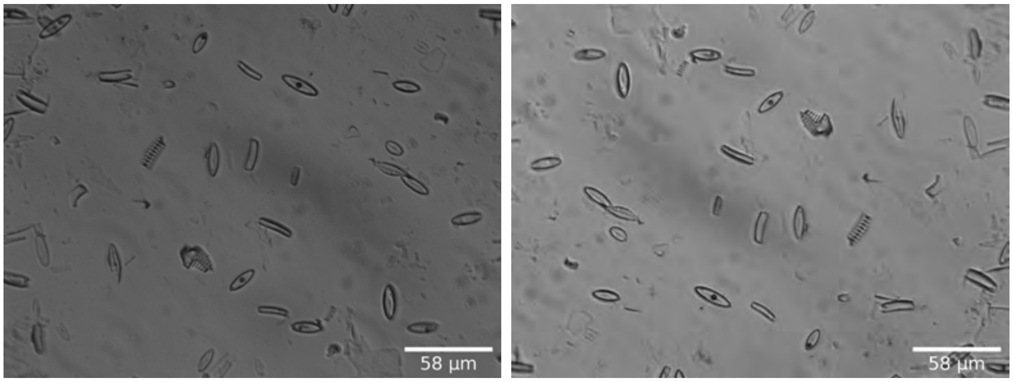
Keywords: Microscopic algae identification, deep learning diatom segmentation, semantic segmentation, SegNet, instance segmentation, Mask-RCNN

1. Introduction

The automatic identification of diatoms in water samples is a challenging problem that has a high impact on water quality assessment. Diatoms are a type of plankton called phytoplankton, a type of microscopic algae that live in water areas like oceans, rivers or lakes and which are used as a bioindicator of its quality [1]. Diatom identification and quantification in water samples are currently done manually, which is a time consuming task.

In order to assess the quality of a water sample, as per the standard workflow, diatoms on 40 field of views (FoV) must be quantified. The implementation of automatic tools based on computer vision and machine learning techniques to perform this task is needed. A number of recent works have dealt with automatic diatom classification, that is, from an image sample containing a single diatom the model tries to predict the correct taxon name. Some classifiers, based on general handcrafted features, have provided good results, around 95% [2] and 98% of accuracy [3]. However, approaches based on convolutional neural networks (CNN) obtain better results, above 99% accuracy [4].

Although automatic classification results are very promising, in practice the taxonomist will handle full size microscopic images containing several taxon shells from different taxa in the same FoV. Thus, it is common that in a single FoV, several diatoms of different species, sizes and shapes appear, along with debris, fragments and dust specks, as shown in Figure 1.a).



(a) Original image.

(b) Modified image.

Figure 1: a) Microscopic image of one FoV from a water sample; b) Modified version of an image for data augmentation with a 180° rotation and contrast enhancement.

Therefore, segmentation methods or region of interest (ROI) detection algorithms are needed to locate all the diatoms present in the image. Once the diatom is detected, by generating a bounding box and/or mask for each instance a classification may be performed for all ROI detected.

A recent review of phytoplankton image segmentation methods is presented in Tang et al. [5]. Most of the methods are based on classical methods such region based segmentation [6], [7], [8], [9] and active contours (AC) [10]. As far as the authors know, there are only two works using deep neural network based segmentation methods ([5] and [11]).

The performance of previous classical methods ranges from 88% to 95%. The main drawbacks are that they are sensitive to noise, like those based on region segmentation, or they need to manually set the initial curve, in the case of AC. Moreover, all of them have been demonstrated only on a single taxon and on images containing a single diatom shell. Only the work of Zheng et al. [12] was demonstrated on images with multiple diatom shells but for a single taxon with an average precision of 0.91% and a sensitivity of 0.81%.

Segmentation techniques based on deep learning may be divided into two approaches: i) object detection and ii) pixel-wise binary classification, i.e., into two classes (ROI or background). In (i) all the instances of the ROI can be located within the image using a bounding box and classified. In (ii) a mask with exactly the pixels that belong to each ROI is inferred.

The object detection algorithms have been tested on diatoms, in previous work by the authors [11], using a Region-based Convolutional Neural Network (R-CNN) [13], [14] and a framework called Darknet [15] with YOLO method [16]. In R-CNN the first step is to provide region proposals and based on these proposals a CNN extracts image features to be classified by a Support Vector Machine (SVM). In YOLO, a single neural network is applied to the whole image. The network divides the image into regions and predicts the class and the bounding box probabilities.

YOLO gives better results than R-CNN in the evaluation carried out with 10 taxa in full microscopic images with multiple diatom shells [11]. This is due to the fact that the model has information about the global context since the network is fed with the full image. Thus, an average F1-measure value of 0.37 with 0.29% precision and 0.68% sensitivity is obtained by the R-CNN against an average F1-measure value of 0.78 with 0.75% precision and 0.84% sensitivity obtained with YOLO. The main problem with these methods is that they do not separate properly the ROIs when overlap occurs. Therefore, the quantification of diatoms is limited.

Segmentation methods based on pixel-wise classification can be roughly divided into two families: i) semantic segmentation and ii) instance segmentation.

Semantic segmentation for diatoms is used by Tang et al. [5] but it is applied to a single taxon on images containing a single diatom shell. Although the authors claim an improvement compared to similar previous studies for the same taxon, with a balanced result between precision and recall, the F1-measure remains low with a value of 0.69.

In this work, we present for the first time the application of instance segmentation applied to diatom segmentation and quantification. Instance segmentation is compared to semantic segmentation. Furthermore, the robustness of the method in noise conditions is analyzed. An average value of 0.85 for F1-measure is obtained with instance segmentation against 0.71 obtained with semantic segmentation applied to images containing multiple diatoms of 10 taxa. All overlapped diatoms were separated and correctly quantified.

The paper is organized as follows. In Section 2, image acquisition, image labeling and dataset preparation are described. The techniques and experiments carried out are presented in Section 3 and the results obtained together with the evaluation metrics used are summarized in Section 4. Finally, conclusions and future work are given in Section 6.

2. Materials

The development of an image segmentation model needs a dataset with samples to train the network effectively. This is a very important step in order to obtain good results, so the dataset selection, image acquisition and labeling tasks have to be done carefully.

2.1. Image acquisition

For this step, it is essential to recruit diatom experts. In this case, the taxonomist Dr. Saúl Blanco Lanza and his team from the Institute of Environment (University of León, Spain) were responsible for collecting a large number of microscopic diatom images, from the same real samples used for the manual identification task.

The typical workflow is as follows. Once the diatom samples is collected from rivers or lakes, a chemical treatment is carried out in a laboratory. First, the samples are processed with hydrogen peroxide to remove organic matter, leaving only the inorganic components like diatom frustules and valves, which are necessary to perform each taxon identification. Then, a few drops are taken in a microscope cover-objects and, after evaporation of water, using a synthetic resin, diatoms are fixed to the glass slide for further classification using microscopes.

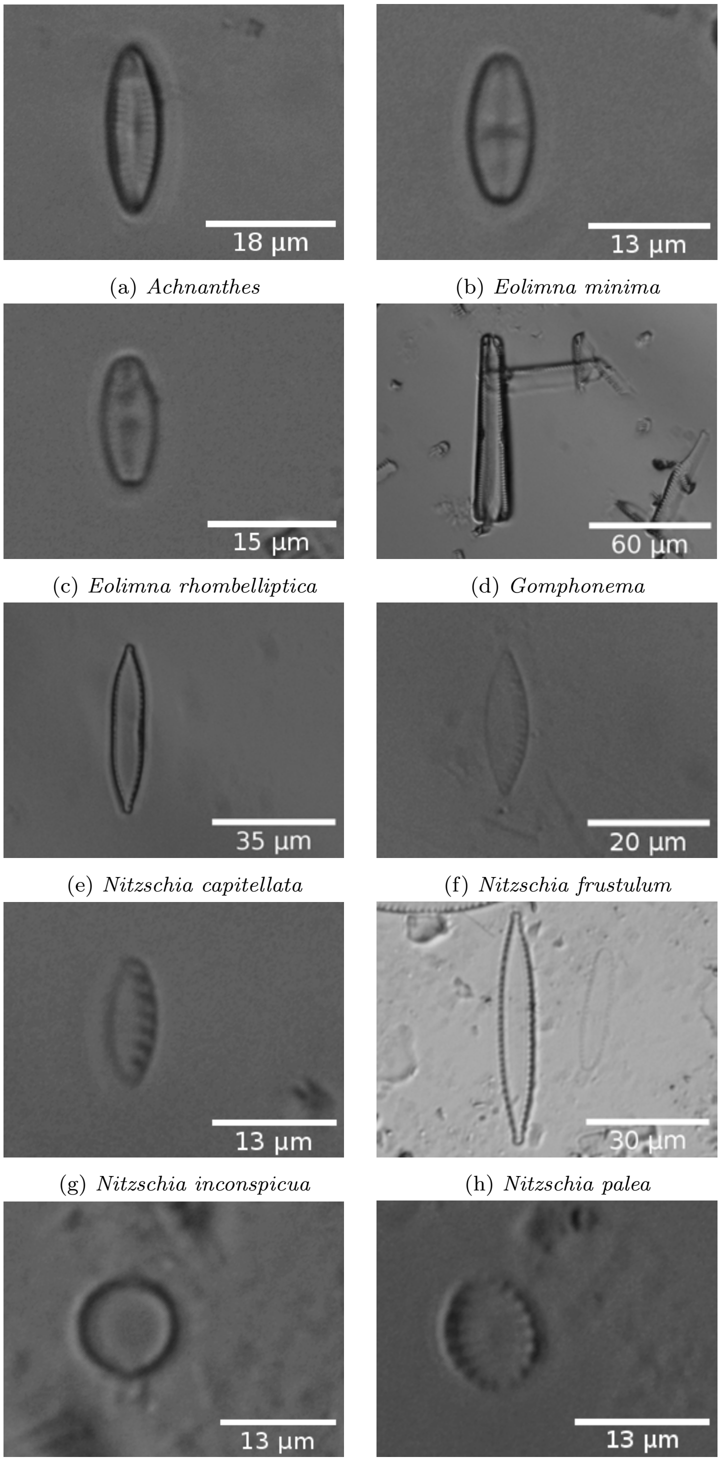
For this comparative study, 126 diatom images of 10 taxa are used, with variety in terms of diatom features (length, internal and external shape) and concentration. All the images are taken with a Brunel SP30 microscope, using a 60x objective and an image resolution of 2592x1944 pixels. In Figure 2 an individual specimen for each selected species is shown and, in Table 1, the number of images per class is presented.

Taxon or Class	**#Images**	**#Diatoms**
1. *Achnanthes subhudsonis*	21	395
2. *Eolimna minima*	12	220
3. *Eolimna rhombelliptica*	7	52
4. *Gomphonema rhombicum*	18	158
5. *Nitzschia capitellata*	10	31
6. *Nitzschia frustulum var frustulum*	14	198
7. *Nitzschia inconspicua*	11	170
8. *Nitzschia palea var palea*	10	105
9. *Skeletonema potamos*	5	55
10. *Staurosira venter*	18	62

Table 1: Diatom species chart, showing the total number of image samples and diatoms per class.

As mentioned before, deep CNNs need many images for training and 126 images may not be enough. However, a commonly used technique in deep learning to alleviate this is fine-tuning, that is based on taking CNNs pre-trained with larger labeled datasets of common objects, like COCO or ImageNet. In this way, useful image features are learned and the specific dataset is then applied to adapt the network weights to our problem.

Another common technique in deep learning to enhance the size and quality of the dataset used is data augmentation. It is based on applying different image processing algorithms to the original dataset, like image rotations, translations, crops, mirror effects, Gaussian noise, contrast enhancements,



(a) *Achnanthes*

(b) *Eolimna minima*

(c) *Eolimna rhombelliptica*

(d) *Gomphonema*

(e) *Nitzschia capitellata*

(f) *Nitzschia frustulum*

(g) *Nitzschia inconspicua*

(h) *Nitzschia palea*

(i) *Skeletonema potamos*

(j) *Staurosira venter*

Figure 2: Examples of the 10 taxa taken into account.

etc.

In the evaluated segmentation approaches, both data augmentation techniques and pre-trained networks are used. The data augmentation done is based on applying random operations such as rotation, mirror and contrast enhancement for each input image for each epoch. In this way, the total number of different images used for training is $N_e * N_{imagesTraining}$, that is the total number of epochs configured for the training by the size of the training dataset. In Figure 1 an example of a modified version of an original training image through the data augmentation procedure is shown.

2.2. Image labeling and dataset preparation

The next step is to manually label the images that will be used later to train the segmentation models. Again, this work has to be carried out by the group of taxonomists due to the difficulty of correctly identifying the ROIs (diatom specimens) present in the images. In Table 1 the number of ROIs labelled per taxon in all images is presented, i.e., the entire ground truth is composed of 1446 diatoms.

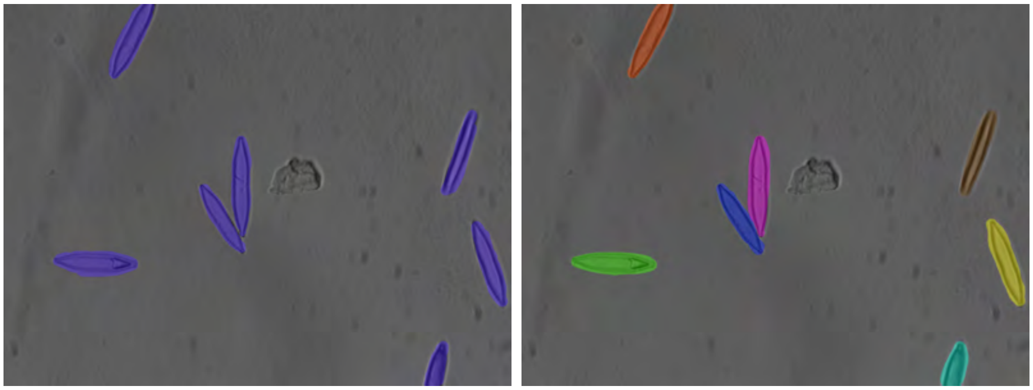
There are many free labeling tools widely used to help in this task. VGG Image Annotator (VIA) [17] was selected in our case. VIA is just a single HTML file that can be opened in any standard web browser, without installing anything else. The graphical user interface is friendly and easy to use, so once the images are imported, the user only has to select the region shape (polygon in this case) and mark the points around the diatom shape. Finally, all the information can be stored in a JSON file, which is a standard format.

To prepare the dataset, all images are divided into two different subsets, one for training (105 images), and the remaining 21 images for validation purposes. The validation subset is formed by images of all the 10 classes, different from the training subset.

3. Methods: Deep learning diatom segmentation

As mentioned in Section 1, an image segmentation algorithm aims to generate a mask indicating exactly which pixels belong to each class, that is, performing a pixel-wise classification into ROI (diatom) or background. There are several architectures or frameworks, which are generally grouped as semantic segmentation or instance segmentation. The main difference is that in semantic segmentation a pixel-level classification is performed directly, while in instance segmentation approaches an additional object detection step is needed to obtain the individual instances of all classes in an image. In Figure 3 an output mask example for each method is represented. The semantic segmentation approaches perform a pixel-level classification, so only one mask for the whole image is generated and individual instances of each class cannot be differentiated. On the other hand, instance segmentation frameworks yield an individual mask for each ROI so that individual instances can be processed separately.

In this work, a comparison of these techniques is carried out.



(a) Semantic segmentation mask.

(b) Instance segmentation mask.

Figure 3: Semantic segmentation mask compared to instance segmentation mask.

3.1. Semantic segmentation

Some of the first deep learning semantic segmentation models tried to directly apply the deep neural network architectures designed for image classification to pixel-level classification. However, the results obtained were not good enough. Convolution, pooling and sub-sampling operations performed by CNNs cause a reduction of the feature map resolution, losing spatial information which is essential for good boundary delimitation, and, therefore, for a good segmentation accuracy. To solve this, novel approaches emerged, such as Fully Convolutional Networks (FCNs) [18], DeconvNet [19], U-Net [20] or SegNet [21]. These models share a similar architecture, with slight differences. In this paper, SegNet is selected due to the good accuracy and efficiency in terms of memory and computational time.

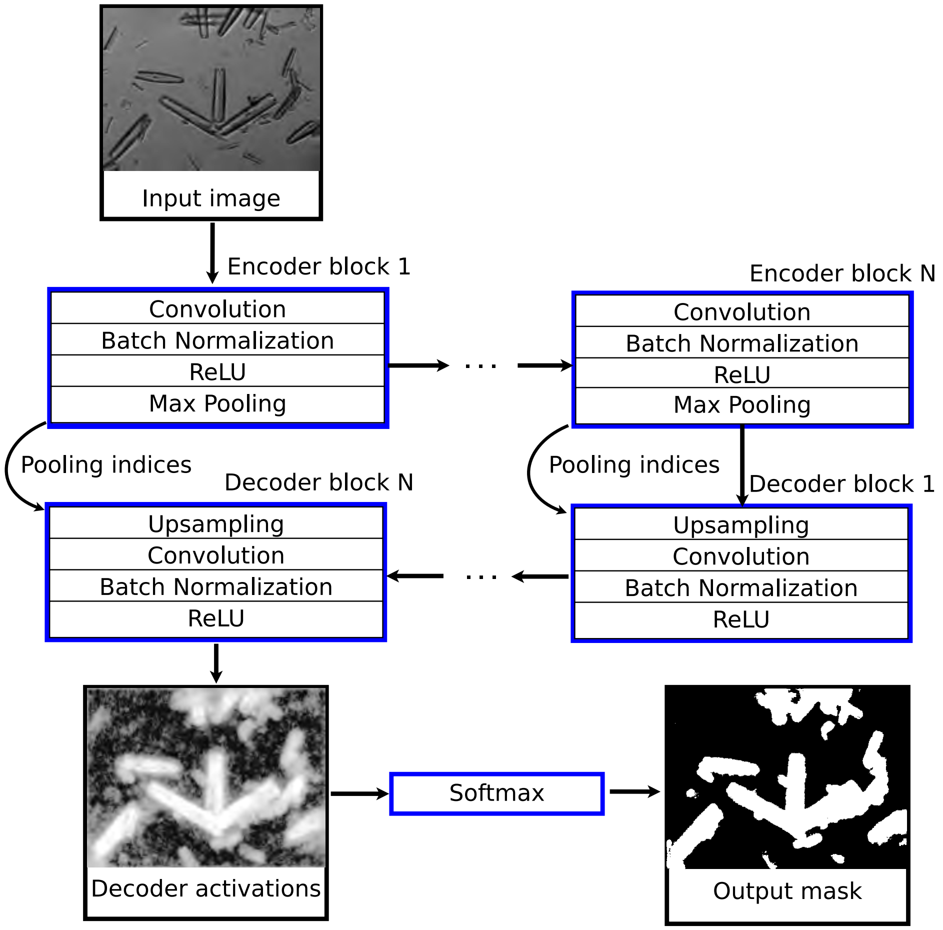


Figure 4: SegNet flowchart. First, the input image is encoded using a set of convolution, batch normalization, ReLU and pooling layers. Then, a decoder network performs the upsampling using the encoder pooling indices. Finally, the output mask is generated taking into account the class probabilities of the softmax layer.

SegNet is an architecture originally designed for scene understanding applications, such as autonomous driving. For this reason, efficiency and speed at inference time are crucial. The architecture of SegNet is formed by an encoder network, a corresponding decoder network and a final pixel-level classification layer. The encoder network is formed by the first 13 layers of the popular VGG16 network [22], pretrained on a large image classification dataset, like ImageNet or COCO. These layers are a combination of convolution, batch normalization, ReLU and max-pooling operations which generate the feature maps. As aforementioned, convolution and pooling operations performed cause a reduction of the feature map resolution, affecting the final segmentation accuracy. In SegNet, the fully connected layers of VGG16 are replaced by a decoder network (one decoder for each encoder), which is responsible for upsampling the input feature maps to a higher resolution. To achieve this, the indices of each max-pooling layer (position of the maximum feature value) at encoding stage are stored to capture the spatial information, and, at decoding stage, these indices are used to perform the upsampling. Finally, the output of this decoding stage (the high resolution feature maps) is the input of a softmax layer, which carries out a pixel-level classification. These steps are graphically summarized in Figure 4. In deep neural networks it is important to select a loss or cost function that allows a good estimate of class probability, especially in this kind of multiclass classification problems. In [23], a depth study about the necessary and sufficient conditions that a cost function must satisfy to provide estimates of the probabilities of the classes. The well-known cross entropy loss is the cost function used in the SegNet architecture, which satisfies these established conditions. There are another interesting cost functions based on the estimation of the conditional density functions of the different classes [24][25], which may be useful in several situations, but are beyond the scope of this work.

The other mentioned alternatives, like FCNs, DeconvNet or U-Net, differ mainly at the decoding stage. FCNs only have one decoder layer and uses bilinear interpolation for upsampling instead of multiple decoding layers and learnable weight filters. DeconvNet has a larger number of parameters and needs more computational resources and in U-Net the upsampling is done by taking the entire feature map at the encoding stage.

3.2. Instance segmentation

Instance segmentation models can be defined as a combination of object detection and semantic segmentation methods. Instance segmentation relies on object detection algorithms to obtain the individual instances of all classes present in an image. Then, each individual ROI is classified at pixel-level to generate the output mask. These approaches have several advantages, like segmentation accuracy and overlapping object differentiation. In the first case, as only individual ROIs are taken into account (instead of the whole image), the segmentation accuracy improves. Also, overlapping objects of the same class are easily separated, unlike in semantic segmentation techniques (which only have pixel-level classification). This is important in applications like diatom identification, in which it is essential to count the number of specimens of each class. However, instance segmentation has an important drawback. As they trust in object detection methods to find the individual instances, only the detected ones will be segmented, so its performance depends on the performance of the object detection technique used.

In the literature, several approaches have appeared recently related to instance segmentation. Some of them are based on segment proposals [26][27], which first propose segment candidates and then each candidate is classified. Another group of methods, using the output masks of semantic segmentation models, tries to separate the pixels of the same classes to create instances [28][29]. Finally, other approaches follow a different strategy, like FCIS [30] and Mask-RCNN [31], which first generate instances and then perform the segmentation and classification in parallel. In this paper, due to the good results achieved, outperforming COCO 2015 and COCO 2016 segmentation challenge winners, the Mask-RCNN method is applied to the diatom segmentation problem.

Mask-RCNN is a modified version of the Faster-RCNN object detection framework with an additional branch to perform the segmentation of the detected ROIs. The first step of the framework is to create a feature map from a given image, using a CNN. Then, a Region Proposal Network (RPN) proposes candidate object bounding boxes. The RPN takes the input feature map and, using a sliding window, several anchor boxes (of multiples scales and aspect ratios) are tested. As an output, RPN gives both the box coordinates and an object probability.

Until this point, the architecture is the same as that of the Faster-RCNN framework, although next, we describe some important differences. *RoiPool* is the Faster-RCNN layer that obtains the individual ROI feature maps using the bounding box proposals of the RPN. The way this operation is done introduces misalignments between the ROI and the feature maps. In segmentation tasks, an exact spatial location is crucial to predict pixel accurate masks, so in Mask-RCNN this layer is changed to a *RoiAlign* layer, which properly aligns the feature maps with the bounding boxes. *RoiAlign*, instead of using quantized bins in *RoiPool*, uses continuous bins and bilinear interpolation to preserve the spatial correspondence better. A fully connected branch predicts at the same time both the class (using a softmax layer) and the object bounds (bounding box regression). Also, Mask-RCNN adds a parallel mask prediction branch to perform ROI segmentation. In this stage, a FCN performs a pixel-level classification for each ROI and for each class, that is, a mask is generated for each class, so there is no competition between

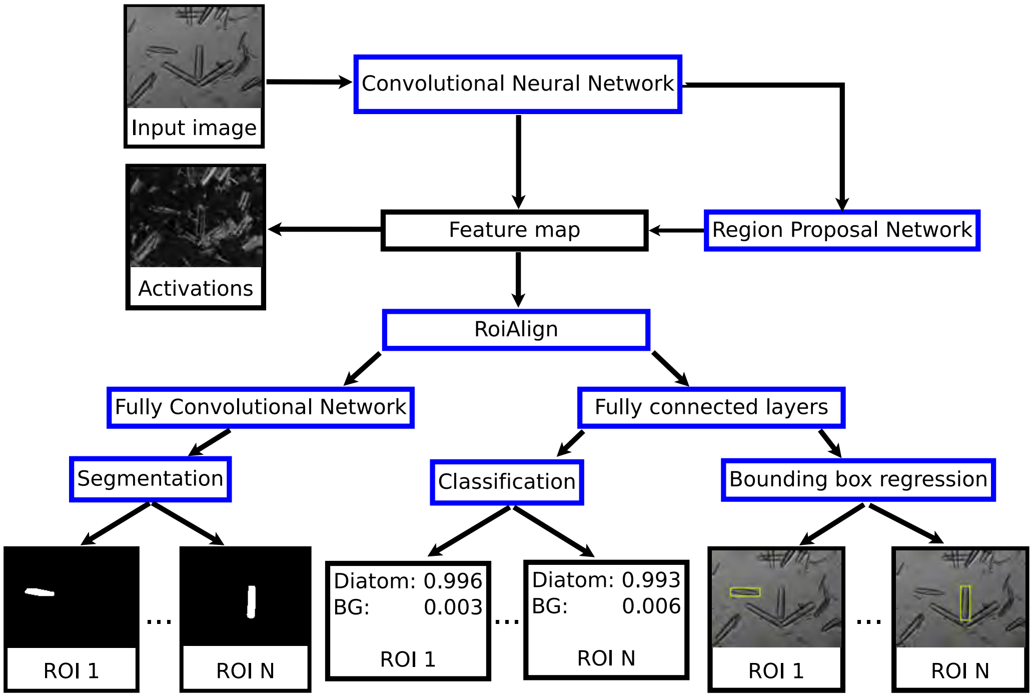


Figure 5: Mask-RCNN flowchart. The input image feature map is generated through a CNN and the RPN. *RoiAlign* layer aligns the feature maps with the bounding boxes. The class and the object bounds are inferred by a fully connected layer and, in parallel, a FCN predicts the mask for each ROI.

classes. In this way, the total loss function of the framework, L, is calculated as the sum of the individual loss functions of classification L_{cls}, bounding box regression L_{box} and segmentation L_{mask}, as defined in Equation 1.

$$L = L_{cls} + L_{box} + L_{mask} \tag{1}$$

Common semantic segmentation networks, as SegNet, use a per-pixel softmax and multiclass cross entropy loss function. However, the FCN of the Mask-RCNN framework uses a per-pixel sigmoid binary cross entropy loss, so, as stated before, there is no competition between classes.

In Figure 5 the main architectural components of the Mask-RCNN framework are presented.

4. Experiments and results

In this Section, all the experiments and their results are presented. First, the validation metrics used for this study are reviewed. Then, the implementation details of the tested frameworks and the results obtained are summarized. Finally, an image quality assessment of the images and the performance analysis of the methods with respect to different types of noise is carried out.

4.1. Validation metrics

The metrics used to measure the performance of segmentation methods are [32]:

- *Sensitivity*: The sensitivity or recall can be measured in terms of True positive (TP) and false negative (FN), at pixel-level, following Equation 2. TP pixels are those that belong to the class and are predicted as positives. On the other hand, FN, also known as type 2 error, are pixels that belong to the class although are predicted as negative. This metric gives the proportion of correctly classified positives.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

- *Precision*: Similar to the previous one although taking into account false positives (FP) instead of FN (Equation 3). FP (type 1 error) pixels are those that do not belong to the ROI although they are predicted as positive. This metric gives the probability of correct segmentation if the prediction is positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

- *Specificity*: This metric gives the proportion of correctly segmented negatives, and follows Equation 4. True negative (TN) pixels are those that do not belong to the ROI and they are predicted as negatives.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

- *IoU*: Intersection over union (IoU) is the most commonly used metric to evaluate segmentation techniques. It is also known as the Jaccard similarity coefficient, and follows Equation 5. IoU metric is an statistical sensitivity measurement that penalizes FP.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{5}$$

- *F1-Measure*: F1-Measure score is the harmonic mean between precision and recall. This metric indicates how well the predicted and the true boundary are aligned. It follows Equation 6.

$$\text{F1-Measure} = 2 * \frac{\text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{6}$$

4.2. SegNet

The SegNet implementation used for this experimentation is based on a VGG16 network pretrained with the ImageNet dataset for the feature extraction stage. Then, a decoder network upsamples the input feature maps to a higher resolution to preserve the spatial information using the max-pooling indices. As the classes are unbalanced (there are more background pixels than diatom pixels), a class weighting is performed in the classification layer.

The training procedure was configured with a learning rate of 0.05 and 100 epochs. The selected optimizer was Stochastic Gradient Descent with a 0.9 of momentum coefficient. The images were resized to 480x360, preserving the aspect ratio to allow a mini-batch size of 4 images. After the training stage, the model performance was evaluated using the validation dataset and the ground truth masks.

In Table 2 the values of the evaluation metrics are presented. These metrics were calculated both for individual species and the whole validation dataset. As the classes are unbalanced (there are more background pixels than diatom pixels), the evaluation was performed using a bounding box around each diatom of the ground truth image. That is, only the pixels inside each bounding box were taken into account, so the results are more representative (taking the whole image means a higher number of TN).

To graphically visualize the effectiveness of the trained model a plot was generated. The performance is evaluated in terms of True Positive Rate (TPR) or sensitivity and True Negative Rate (TNR), which is calculated

Class	**Precision**	**Sensitivity**	**Specificity**	**IoU**	**F1**
1. *Achnanthes subhudsonis*	0.59	0.98	0.62	0.58	0.73
2. *Eolimna minima*	0.52	1.00	0.61	0.52	0.68
3. *Eolimna rhombelliptica*	0.45	1.00	0.22	0.45	0.62
4. *Gomphonema rhombicum*	0.69	0.96	0.84	0.67	0.80
5. *Nitzschia capitellata*	0.63	1.00	0.71	0.63	0.78
6. *Nitzschia frustulum*	0.54	0.95	0.60	0.53	0.69
7. *Nitzschia inconspicua*	0.53	0.90	0.51	0.50	0.67
8. *Nitzschia palea var palea*	0.62	0.78	0.82	0.52	0.69
9. *Skeletonema potamos*	0.59	0.89	0.56	0.55	0.71
10. *Staurosira venter*	0.58	1.00	0.49	0.58	0.74
Average	**0.57**	**0.95**	**0.60**	**0.55**	**0.71**

Table 2: SegNet results for each class in the validation dataset.

as (1 - specificity). A good model should have a high TPR and low TNR. In Figure 6, the performance plot for all the validation images is presented. The SegNet model gives a high TPR, that is, the number of FN is very low compared with the TP. However, the TNR is also too high, which means that the model predicts a high number of FP.

4.3. Mask-RCNN

The Mask-RCNN implementation employed in this study is built by Matterpot [33], based on the Keras and TensorFlow frameworks. As feature extraction CNN, a ResNet101 [34] pretrained with the COCO dataset was used. Also, this implementation uses a modified version of ResNet with Feature Pyramid Network (FPN) architecture [35], which is a top-down approach that allows extracting features at different scales and gives better results in both accuracy and speed.

The training procedure was configured with a learning rate of 0.001 and 30 epochs. The selected optimizer was Stochastic Gradient Descent with a 0.9 of momentum coefficient and the mini-batch size was fixed to 2 images. After the training stage, the model performance was evaluated using the validation dataset and the ground truth masks. In Table 3 the values of the evaluation metrics are presented. These metrics were calculated both for individual species and the whole validation dataset. In the same way, as in SegNet, the evaluation was done using a bounding box around each diatom of the ground truth image.

The performance plot for the Mask-RCNN trained model is shown in Figure 7. In this case, the TPR is lower compared to the SegNet model, that is,

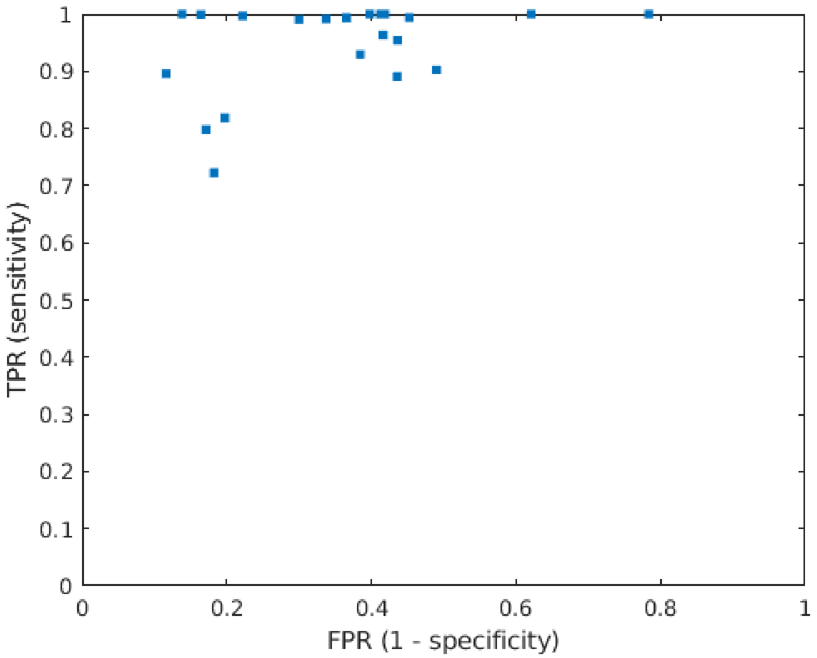


Figure 6: SegNet performance graph. Each point represents one of the validation images.

there are TPs that are not predicted correctly. However, the TNR is lower too, which means that the model predicts a low number of FPs.

In Figure 8 a pair of diatom images of the validation set with their corresponding predicted mask for the SegNet and Mask-RCNN models is illustrated. The green pixels indicate true positives, false negatives are marked in blue and false positives in red, taking into account the ground truth mask. As can be seen from the figure, the number of false negatives is smaller in SegNet mask images, although the number of false positives is higher.

In addition to global differences at pixel-level classification, the biggest difference between SegNet and Mask-RCNN is the way in which the final masks are generated. As previously mentioned, SegNet generates a single mask, which makes it impossible to distinguish directly the different instances of the same class present in the image. To approximate this, it is necessary to carry out a mask post-processing to separate and locate the different objects. The Mask-RCNN framework gives, for each located object, the class probability, a bounding box and the predicted mask, among others. In Fig-

Class	**Precision**	**Sensitivity**	**Specificity**	**IoU**	**F1**
1. *Achnanthes subhudsonis*	0.87	0.86	0.92	0.75	0.86
2. *Eolimna minima*	0.79	0.90	0.89	0.73	0.84
3. *Eolimna rhombelliptica*	0.79	0.88	0.84	0.72	0.84
4. *Gomphonema rhombicum*	0.92	0.82	0.98	0.76	0.86
5. *Nitzschia capitellata*	0.86	0.94	0.91	0.81	0.90
6. *Nitzschia frustulum*	0.86	0.90	0.92	0.78	0.88
7. *Nitzschia inconspicua*	0.82	0.67	0.91	0.59	0.74
8. *Nitzschia palea var palea*	0.86	0.79	0.95	0.70	0.82
9. *Skeletonema potamos*	0.86	0.84	0.89	0.74	0.85
10. *Staurosira venter*	0.87	0.98	0.89	0.85	0.92
Average	**0.85**	**0.86**	**0.91**	**0.74**	**0.85**

Table 3: Mask-RCNN metrics results for each class in the validation dataset.

ure 9 and Figure 10 a comparison between SegNet and Mask-RCNN in terms of individual diatom localization is performed using 10 diatom images (one for each class). The differences are more remarkable in cases of overlapping or closer diatoms, which are difficult to separate in the SegNet masks. On the other hand, in Mask-RCNN the individual bounding boxes are obtained directly.

Counting the number of diatoms present in an image is essential for water quality assessment. The final output of SegNet and Mask-RCNN aimed to quantified all diatoms per images from the predicted masks is illustrated in Figure 9, Figure 10 and Figure 11). The images represented a FoV where most of the diatoms belong to one of the taxa considered.

It is possible to see in Table 4 how the count for Mask-RCNN masks is closer to the ground truth. SegNet cannot separate properly the diatoms and counts debris as diatoms. These errors lead to a higher value of FPs when counting individual diatoms. Mask-RCNN detects properly most of the diatoms and some FN errors happen.

4.4. Image quality assessment and performance of segmentation

Nowadays, there are metrics that can objectively approximate image quality using image features like color, contrast, entropy, luminance or texture [36][37]. In this study, quality is evaluated in terms of defocusing and granular noise using anisotropy and Sum of Modified Laplace transform (SML) metrics.

Anisotropy is measured as the variance of the entropy in several directions and is based on the fact that degradation in the image damages the

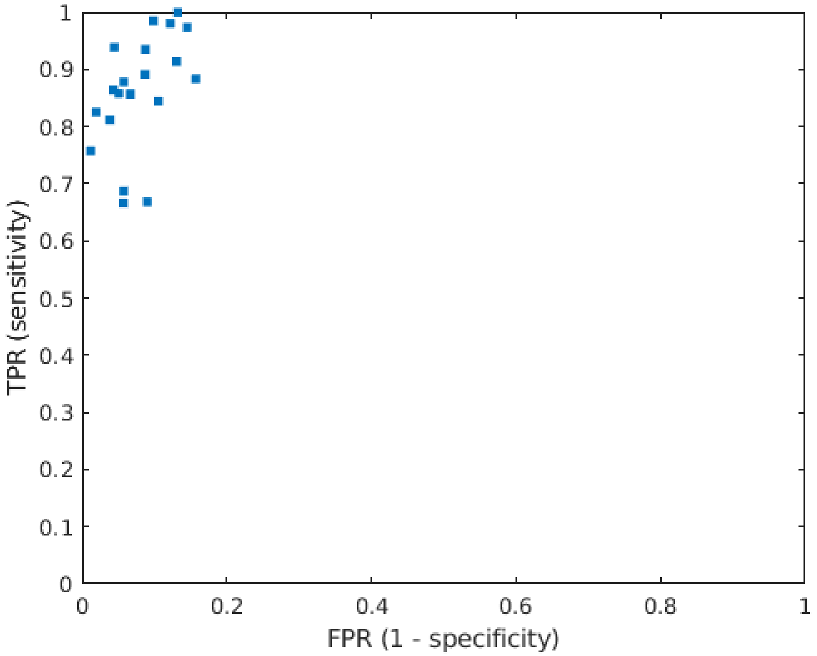


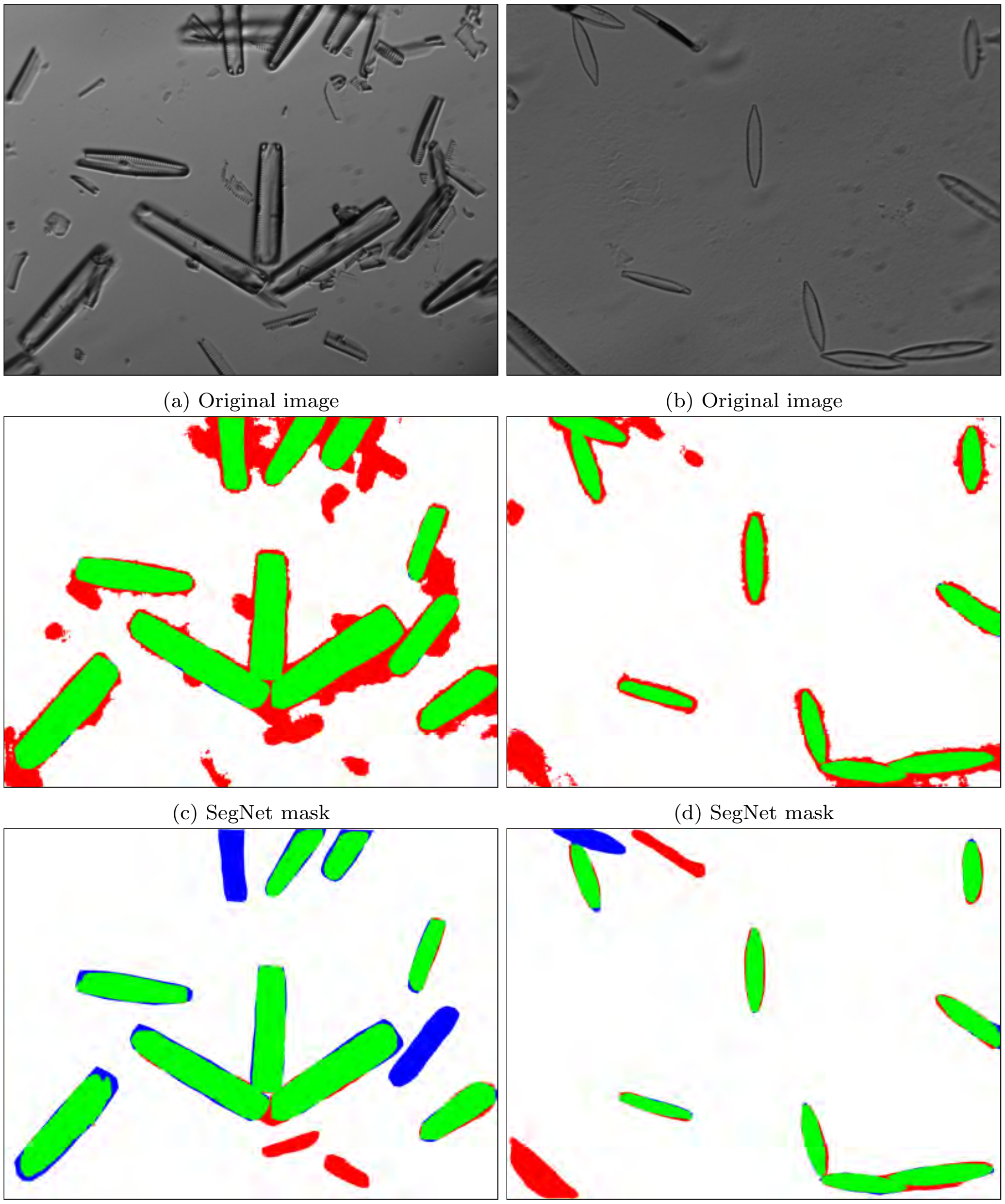
Figure 7: Mask-RCNN performance graph. Each point represents one of the validation images.

directional information. For that reason, anisotropy decreases as more distortions are added to the image and it is sensitive to blurriness. The complete description of the method is presented in [38].

SML is a derivative-based metric which uses the Laplacian operator ($\bigtriangledown^2 I(x,y)$) to evaluate the sharpness in an image ($I(x,y)$), as defined in Equation 7, where $L_x(x,y)$ and $L_y(x,y)$ are the images after convolution with the Laplacian operator.

$$F_{SML} = \sum_x \sum_y |L_x(x,y)| + |L_y(x,y)| \tag{7}$$

SML can be used to measure granular noise in an image. This metric also decrease if the image quality decrease. In Figure 12, the anisotropy and SML averaged results are presented for each class. The best average quality is provided by taxon 4, that is *Gomphonema rhombicum* and the worst by taxon 8, *Nitzschia palea var palea* and 10,*Staurosira venter*. These metrics show that, under standard conditions, there is no relationship between image



(a) Original image

(b) Original image

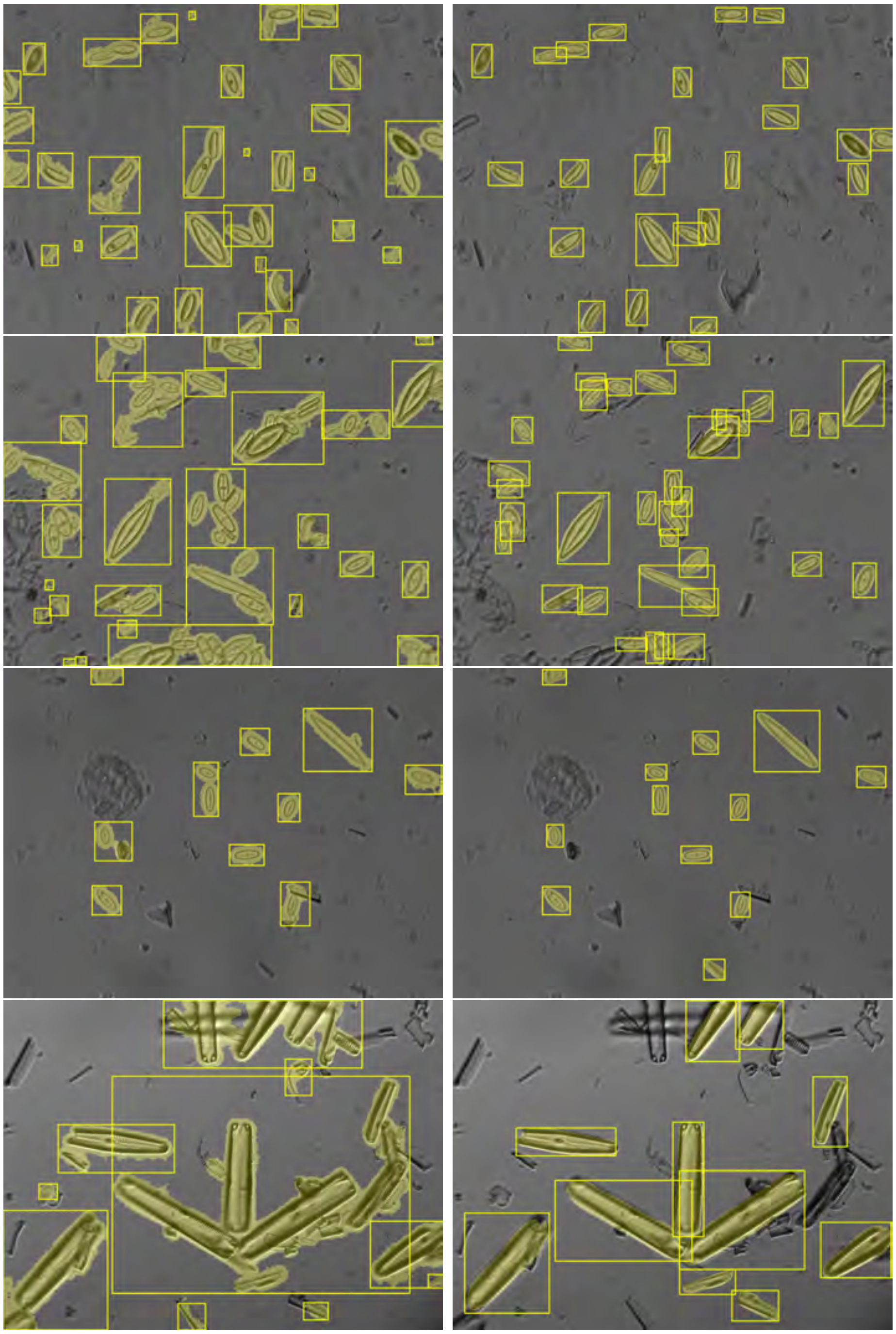
(c) SegNet mask

(d) SegNet mask

(e) Mask-RCNN mask

(f) Mask-RCNN mask

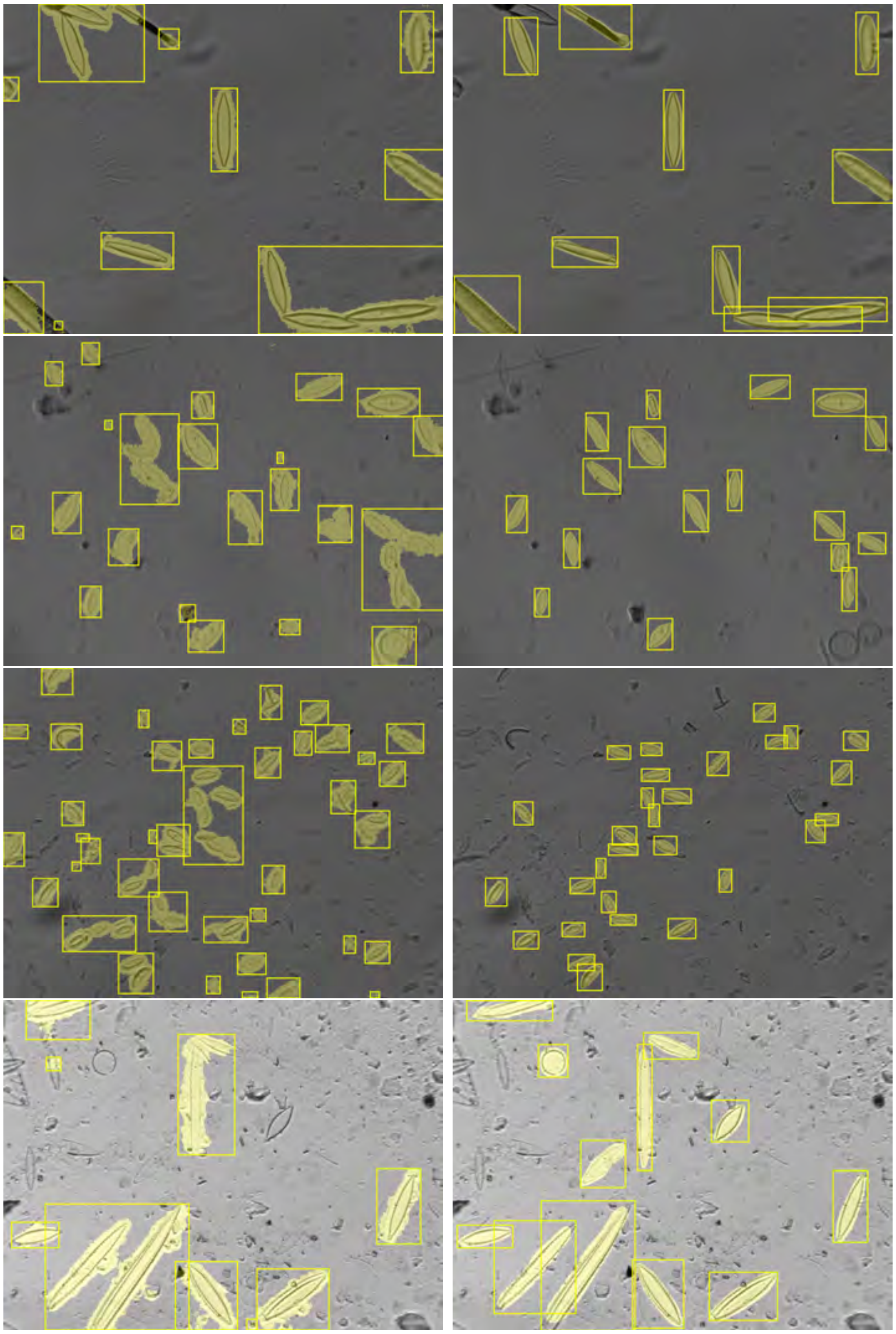
Figure 8: SegNet vs Mask-RCNN prediction masks.



(a) SegNet

(b) Mask-RCNN

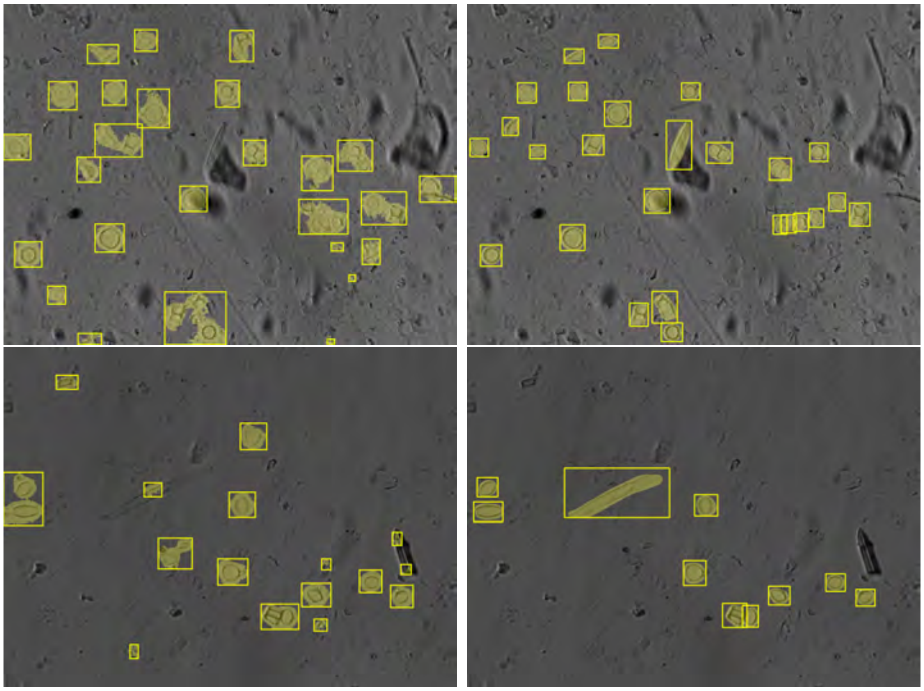
Figure 9: SegNet vs Mask-RCNN individual bounding boxes 1



(a) SegNet

(b) Mask-RCNN

Figure 10: SegNet vs Mask-RCNN individual bounding boxes 2



(a) SegNet

(b) Mask-RCNN

Figure 11: SegNet vs Mask-RCNN individual bounding boxes 3

quality and segmentation performance for each class (Table 2 and Table 3).

A deeper study is done to analyze if the presence of noise can modify the performance of the trained models. To this end, new datasets are created using modified versions of the original images, with different noise types and intensities. The first dataset is formed by several blurred images, using the Gaussian function over the original dataset. In this case, for each image, a set of 40 blurred images was generated varying the standard deviation of the Gaussian function, from 0.5 to 20 with a 0.5 step. For the second dataset, Speckle noise was selected, which is a synthetic granular noise. Similarly, as in the previous dataset, 40 noisy images were generated changing the variance of the Speckle function from 0.125 to 5 with a 0.125 step. Anisotropy and SML metrics present worse results as noise increases, so the image quality decreases.

Image	**Ground truth count**	**SegNet count**	**Mask-RCNN count**
Sample 1	22	TP: 10; FP: 20	TP: 22; FP: 2
Sample 2	35	TP: 3 ; FP: 24	TP: 34; FP: 1
Sample 3	10	TP: 7 ; FP: 3	TP: 10; FP: 2
Sample 4	13	TP: 1 ; FP: 9	TP: 11; FN: 2
Sample 5	10	TP: 4 ; FP: 6	TP: 10; FP: 0
Sample 6	20	TP: 11; FP: 11	TP: 17; FN: 3
Sample 7	29	TP: 10; FP: 30	TP: 28; FN: 1
Sample 8	14	TP: 4 ; FP: 4	TP: 12; FN: 2
Sample 9	22	TP: 6 ; FP: 20	TP: 22; FP: 4
Sample 10	9	TP: 4 ; FP: 12	TP: 9 ; FP: 1

Table 4: Ground truth diatom count compared to both SegNet and Mask-RCNN predicted masks.

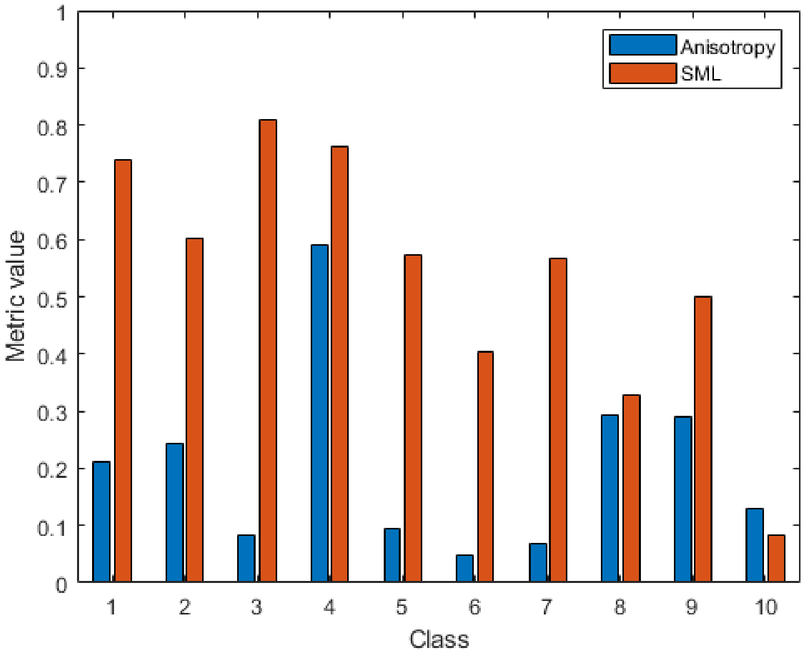
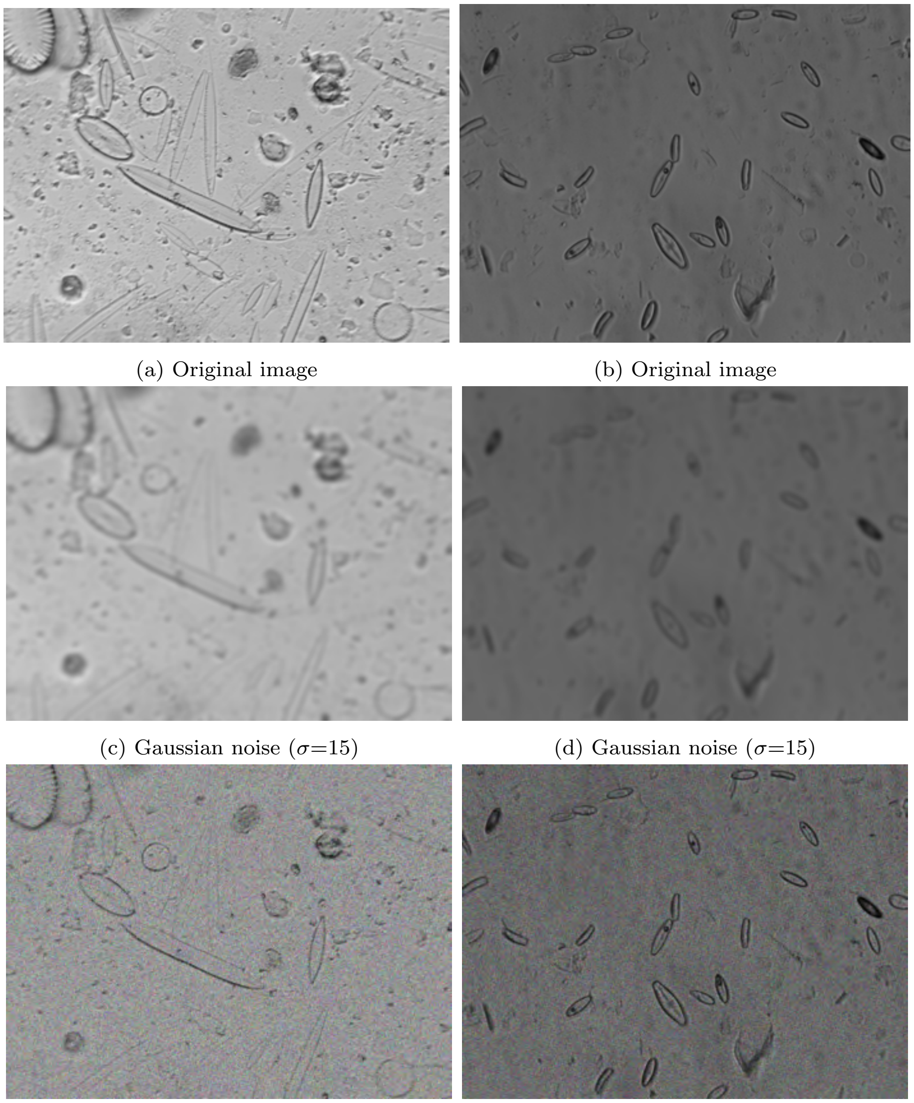


Figure 12: Anisotropy and SML averaged results for each class.



(a) Original image

(b) Original image

(c) Gaussian noise (σ=15)

(d) Gaussian noise (σ=15)

(e) Speckle noise (σ^2=1)

(f) Speckle noise (σ^2=1)

Figure 13: Noisy datasets.

In Figure 13 an example of the two types of added noise compared to the original image is presented.

The predicted masks for the two generated datasets were obtained for both Mask-RCNN and SegNet trained models, with the corresponding evaluation metrics for the segmentation, in the same way as in the original dataset. The results are graphically summarized in a plot that shows the performance in terms of TPR and TNR. For clarity purposes, the total images are divided into 5 groups, one for the original images and the rest for different noise intervals, represented in different colors. In Figure 14, the graph for the Mask-RCNN model performance with Gaussian noise images is presented. As the standard deviation of the Gaussian function increases, the TPR decreases too and the FPR remains low. The SegNet model performance for Gaussian noise images is shown in Figure 16. In this case, similarly to Mask-RCNN model, the TPR and FPR decreases as the standard deviation of the Gaussian function increases. Also, ROC representations for each class are provided in Figure 15 and Figure 17.

The same procedure was applied for Speckle noise dataset. In Figure 18 and Figure 20 the performance results are presented for Mask-RCNN and SegNet models, respectively. The Mask-RCNN model behaves similarly for both Gaussian and Speckle noise, when noise increases, TPR decreases too. However, for the SegNet model, the behaviour is different when Speckle noise increases the FPR also increases, that is, most pixels are marked as positives. Finally, ROC representations for each class are also provided in Figure 19 and Figure 21.

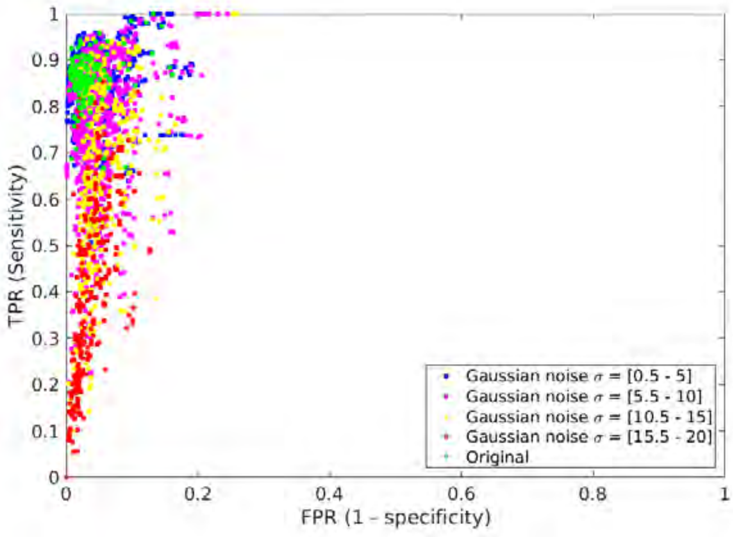


Figure 14: Mask-RCNN performance graph for Gaussian noise images.

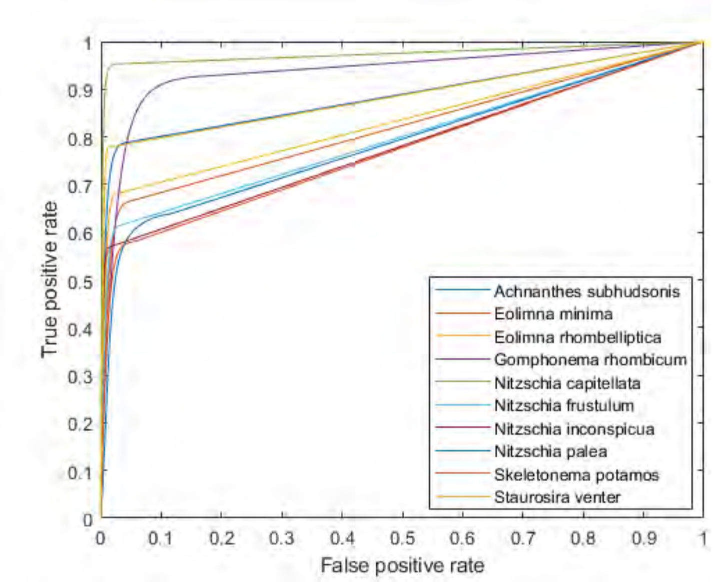


Figure 15: Mask-RCNN ROC graph for Gaussian noise images. Each line represents the ROC curve for each class

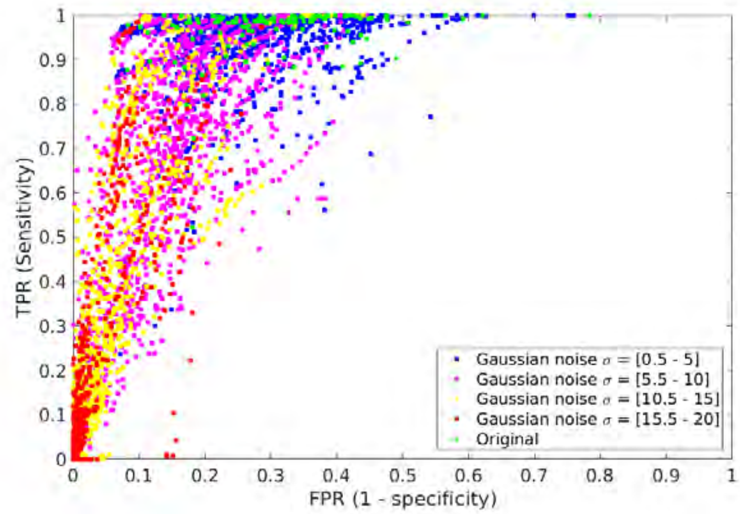


Figure 16: SegNet performance graph for Gaussian noise images.

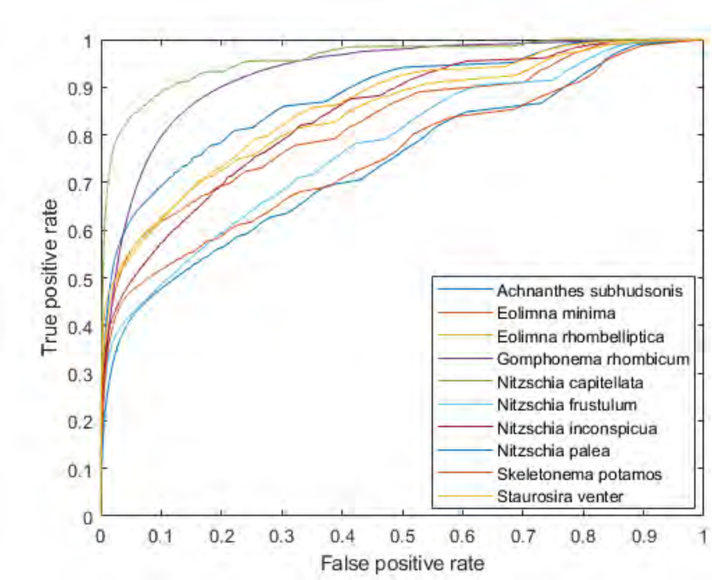


Figure 17: SegNet ROC graph for Gaussian noise images. Each line represents the ROC curve for each class

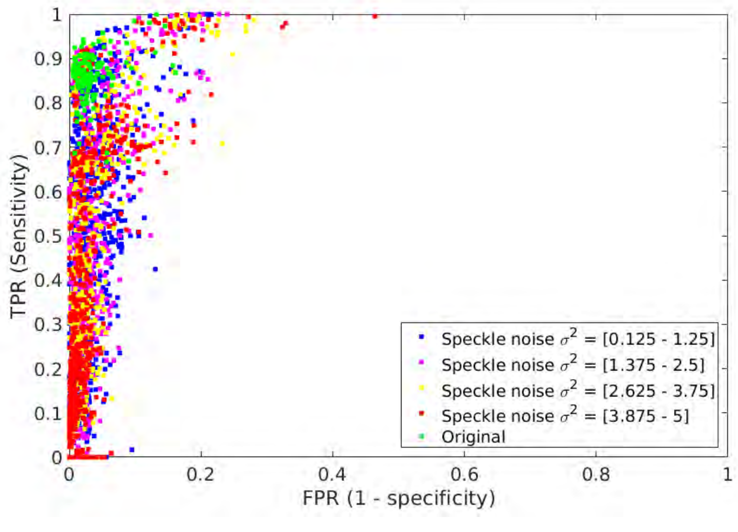


Figure 18: Mask-RCNN performance graph for Speckle noise images.

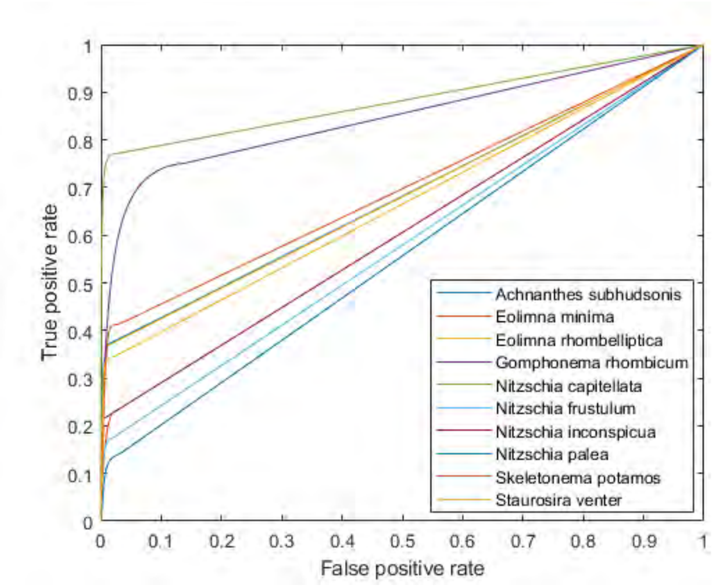


Figure 19: Mask-RCNN ROC graph for Speckle noise images. Each line represents the ROC curve for each class

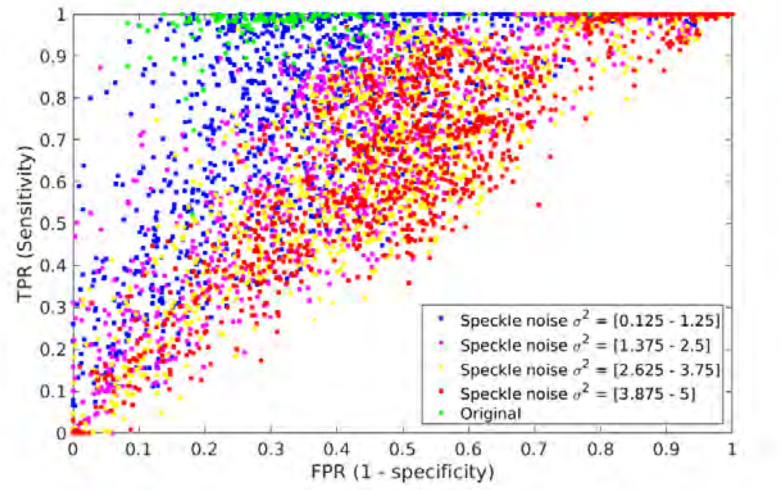


Figure 20: SegNet performance graph for Speckle noise images.

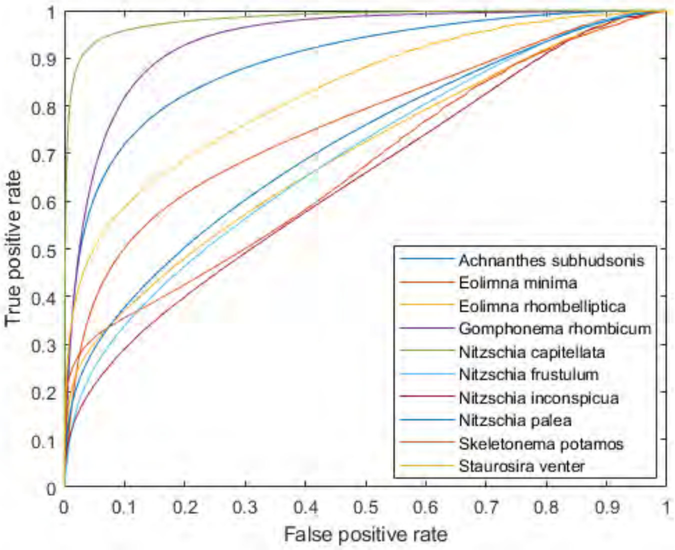


Figure 21: SegNet ROC graph for Speckle noise images. Each line represents the ROC curve for each class

5. Discussion

Mask-RCNN and SegNet models are capable of segment diatoms from the same raw images used for manual identification, without any cropping or preprocessing step. However, the Mask-RCNN model obtains better results because the model has information about the global context. Mask-RCNN first extracts the individual ROIs from the whole image and then performs the segmentation for each one. This approach has two main advantages. The first one is that an individual mask for each ROI is obtained, and so, unlike semantic segmentation approaches, all instances from the same class can be differentiated and quantified. The second one is that the segmentation quality is better in the Mask-RCNN model than in the SegNet model, that is, the border alignment between the ground truth and the predicted mask is more accurate, as the IoU and F1-measure scores show.

The robustness of the trained models was evaluated with modified datasets. These datasets were created adding Gaussian and speckle noise of different intensities to the original images, obtaining 80 new images for each FoV. For the Gaussian noise dataset, in both Mask-RCNN and SegNet models when the noise intensity increases, the TPR decreases and less diatoms are segmented correctly. For the speckle noise dataset, the Mask-RCNN behaviour is similar to the Gaussian noise dataset. However, for the SegNet model, the FPR increases as noise intensity increases, predicting as positive a large number of pixels in the image.

6. Conclusions

A comparison between semantic segmentation and instance segmentation is carried out to detect and quantify microscopic algae (diatoms) of 10 different taxa. This is the first time that the use of deep learning approaches is demonstrated for the identification and quantification of diatoms in images with multiple diatom shells and for more than one taxon.

Instance segmentation with Mask-RCNN achieved an average precision of 0.85% with 0.86% sensitivity and 0.91% specificity, and up to 0.92% precision for taxon *Gomphonema rhombicum* with 0.98%, specificity. This taxon obtained the best image quality measured with the anisotropy and sum of modified Laplace transform metrics.

Regarding future work, the promising results of the Mask-RCNN model encourage us to continue working on instance segmentation approaches, especially with object detection techniques to extract the individual ROIs to be segmented. The main drawback of Mask-RCNN is that the performance of the detection step limits the performance of the segmentation. This fact explains why some diatoms are not segmented in the Mask-RCNN model, resulting in a lower sensitivity score than the SegNet model. Therefore, there is still room to improve this step of the Mask-RCNN procedure.

Acknowledgements

The authors received financial support from the Spanish Government under the Aqualitas-retos project (Ref. CTM2014-51907-C2-2-R-MINECO). Project website: `http://aqualitas-retos.es/en/`

References

[1] S. Blanco, E. Bécares, Are biotic indices sensitive to river toxicants? a comparison of metrics based on diatoms and macro-invertebrates, Chemosphere 79 (2010) 18–25.

[2] K. Schulze, U. M. Tillich, T. Dandekar, M. Frohme, Planktovision-an automated analysis system for the identification of phytoplankton, BMC bioinformatics 14 (2013) 115.

[3] G. Bueno, O. Deniz, A. Pedraza, J. Ruiz-Santaquiteria, J. Salido, G. Cristóbal, M. Borrego-Ramos, S. Blanco, Automated diatom classification (part a): handcrafted feature approaches, Applied Sciences 7 (2017) 753.

[4] A. Pedraza, G. Bueno, O. Deniz, G. Cristóbal, S. Blanco, M. Borrego-Ramos, Automated diatom classification (part b): a deep learning approach, Applied Sciences 7 (2017) 460.

[5] N. Tang, F. Zhou, Z. Gu, H. Zheng, Z. Yu, B. Zheng, Unsupervised pixelwise classification for chaetoceros image segmentation, Neurocomputing 318 (2018) 261–270.

[6] A. C. Jalba, M. H. Wilkinson, J. B. Roerdink, Automatic segmentation of diatom images for classification, Microscopy Research and Technique 65 (2004) 72–85.

[7] A. Verikas, A. Gelzinis, M. Bacauskiene, I. Olenina, S. Olenin, E. Vaiciukynas, Phase congruency-based detection of circular objects applied to analysis of phytoplankton images, Pattern Recognition 45 (2012) 1659–1670.

[8] H. Zheng, H. Zhao, X. Sun, H. Gao, G. Ji, Automatic setae segmentation from chaetoceros microscopic images, Microscopy Research and Technique 77 (2014) 684–690.

[9] H. Zheng, R. Wang, Z. Yu, N. Wang, Z. Gu, B. Zheng, Automatic plankton image classification combining multiple view features via multiple kernel learning, BMC Bioinformatics 18 (2017) 570.

[10] A. Gelzinis, A. Verikas, E. Vaiciukynas, M. Bacauskiene, A novel technique to extract accurate cell contours applied for segmentation of phytoplankton images, Machine Vision and Applications 26 (2015) 305–315.

[11] A. Pedraza, G. Bueno, O. Déniz, J. Ruiz-Santaquiteria, C. Sanchez, S. Blanco, M. Borrego-Ramos, A. Olenici, G. Cristobal, Lights and pitfalls of convolutional neural networks for diatom identification, Optics,

Photonics, and Digital Technologies for Imaging Applications V 10679 (2018) 106790G.

[12] H. Zheng, N. Wang, Z. Yu, Z. Gu, B. Zheng, Robust and automatic cell detection and segmentation from microscopic images of non-setae phytoplankton species, IET Image Processing 11 (2017) 1077–1085.

[13] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, International Journal of Computer Vision 104 (2013) 154–171.

[14] R. Girshick, Fast R-CNN, Proceedings of the IEEE International Conference on Computer Vision (2015) 1440–1448.

[15] J. Redmon, Darknet: Open Source Neural Networks in C, `http://pjreddie.com/darknet/`, 2013–2016. Accessed: 25/01/2019.

[16] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 779–788.

[17] A. Dutta, A. Gupta, A. Zissermann, VGG image annotator (VIA), http://www.robots.ox.ac.uk/ vgg/software/via/, 2016. Version: 2.0.5, Accessed: 25/01/2019.

[18] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015) 3431–3440.

[19] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, Proceedings of the IEEE International Conference on Computer Vision (2015) 1520–1528.

[20] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-assisted Intervention (2015) 234–241.

[21] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, arXiv preprint arXiv:1511.00561 (2015).

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

[23] J. Cid-Sueiro, J. I. Arribas, S. Urbán-Munoz, A. R. Figueiras-Vidal, Cost functions to estimate a posteriori probabilities in multiclass problems, IEEE Transactions on Neural Networks 10 (1999) 645–656.

[24] J. I. Arribas, J. Cid-Sueiro, T. Adali, A. R. Figueiras-Vidal, Neural networks to estimate ml multi-class constrained conditional probability density functions, IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339) 2 (1999) 1429–1432.

[25] J. I. Arribas, J. Cid-Sueiro, T. Adali, A. R. Figueiras-Vidal, Neural architectures for parametric estimation of a posteriori probabilities by constrained conditional density functions (1999) 263–272.

[26] P. O. Pinheiro, R. Collobert, P. Dollár, Learning to segment object candidates, Advances in Neural Information Processing Systems (2015) 1990–1998.

[27] J. Dai, K. He, Y. Li, S. Ren, J. Sun, Instance-sensitive fully convolutional networks, European Conference on Computer Vision (2016) 534–549.

[28] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, C. Rother, Instancecut: from edges to instances with multicut, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017) 5008–5017.

[29] M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017) 5221–5229.

[30] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017) 2359–2367.

[31] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, Computer Vision (ICCV), 2017 IEEE International Conference (2017) 2980–2988.

[32] G. Csurka, D. Larlus, F. Perronnin, F. Meylan, What is a good evaluation measure for semantic segmentation?., BMVC 27 (2013) 2013.

[33] W. Abdulla, Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow, `https://github.com/matterport/Mask_RCNN`, 2017.

[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) 770–778.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017) 2117–2125.

[36] A. Jiménez, G. Bueno, G. Cristóbal, O. Déniz, D. Toomey, C. Conway, Image quality metrics applied to digital pathology, Optics, Photonics and Digital Technologies for Imaging Applications IV 9896 (2016) 98960S.

[37] J. Ruiz-Santaquiteria, J. L. Espinosa-Aranda, O. Deniz, C. Sanchez, M. Borrego-Ramos, S. Blanco, G. Cristobal, G. Bueno, Low-cost oblique illumination: an image quality assessment, Journal of Biomedical Optics 23 (2018) 016001.

[38] S. Gabarda, G. Cristóbal, Blind image quality assessment through anisotropy, Journal of the Optical Society of America A 24 (2007) B42–B51.