A two-staged Approach for Localization and Classification of Coral Reef Structures and Compositions

Kirill Bogomasov, Philipp Grawe, and Stefan Conrad

Heinrich Heine University, Universitätsstraße 1, 40225 Düsseldorf, Germany
http://dbs.cs.uni-duesseldorf.de
{bogomasov,grawe,stefan.conrad}@hhu.de

Abstract. In this paper we present the approaches that achieved the first place in this years ImageCLEFcoral challenge. The task of the challenge was the localization and classification of corals within images of sea ground. Therefore we had to extract bounding boxes for each coral and labeling them with the specific type of substrate.

We applied a state-of-the-art deep learning approach (YOLO) and also developed a two-staged approach, using a grid along with two classifiers. One that classifies the tiles of the grid, the other that classifies the found boxes.

We had moderate results using YOLO and discovered that locating the corals is the most challenging part. Furthermore class imbalance and intersecting boxes, made the problem even harder.

Keywords: Image Segmentation · Image Classification · Object Localization

1 Introduction

Climate change is one of the major problems of the 21st century. Its impact is growing every year and therefore researched a lot. Since corals are a significant part of the maritime environment they are affected by the climate change in many ways [6]. Corals have their own self-contained and over many decades developed ecosystem, which is why the influence of damage to coral reefs can have serious consequences for every maritime organism. Every year the danger of complete destruction of coral reefs becomes more realistic. To sophisticatedly plan protective procedures, coverage of current stocks are required. For this purpose, images of the sea ground are currently viewed and annotated manually, which is nearly impossible for the whole considered surface area. This raises the question of whether an automatic localication and annotation of the coral is feasible. We will address this question in this paper. Therefore we use this year's

ImageCLEFcoral dataset [2] as the base for our research and also participated in their challenge, which is part of the ImageCLEF 2019 [8]. The task can be divided into two logical subtasks, localization and classification of objects. This is a wide spread research field of computer science, with many fields of application. The automotive industry seems to be an obvious field of research [3] and is commercially relevant. Today, driver assistance systems are ubiquitous. Recognition of road signs is a part of it. At first images of the road are taken via vehicle camera while driving. Second road signs are searched and classified in these recordings. The greatest results in such application scenarios are achieved by artificial neural networks. YOLO [11] showed one of the best results. The application scenario can be transferred very well. The localization and labeling of corals is similar, because the images are taken automatically and contain corals in unknown areas.

2 Data

The training set, which we define as dataset (A), contains 240 images with 6670 annotated substrates. Generally there is a differentiation between 13 substrate types. Which are: "Hard Coral – Branching, Hard Coral – Submassive, Hard Coral – Boulder, Hard Coral – Encrusting, Hard Coral – Table, Hard Coral – Foliose, Hard Coral – Mushroom, Soft Coral, Soft Coral – Gorgonian, Sponge, Sponge – Barrel, Fire Coral – Millepora and Algae - Macro or Leaves" [2]. For the submitted runs a test set containing 200 raw images is used, which correct labels and boxes were not available at the time of the publication.

Table 1: **Substrate types** with their relative frequency in the training set.

Class label	Relative frequency
c_algae_macro_or_leaves	0.0046
c_fire_coral_millepora	0.0015
c_hard_coral_boulder	0.1549
c_hard_coral_branching	0.1280
c_hard_coral_encrusting	0.0528
c_hard_coral_foliose	0.0082
c_hard_coral_mushroom	0.0258
c_hard_coral_submassive	0.0031
c_hard_coral_table	0.0009
c_soft_coral	0.5223
c_soft_coral_gorgonian	0.0024
c_sponge	0.0808
c_sponge_barrel	0.0145

The substrate types have an unbalanced distribution, as shown in table 1. Furthermore does the quality of the images vary, as well as the resolution. Some of the images contain a measurement white line, which is an obstacle while image processing.

2.1 Investigating the Dataset

When investigating the dataset, the problem of overlapping boxes appeared to us. Many of these bounding boxes fully contained or intersected with other boxes. To be more specific, only 2672 of 6670 bounding boxes do neither overlap or are contained in a bigger one. For this reason, we had started to investigate whether the substrates differ from each other at all, why we searched for meaningful features. These features were extracted from extracted bounding boxes. We applied a classical approach using SIFT [9]. Furthermore we calculated structure[5], texture[7] and color histograms in another approach. We used the calculated features to train a k-Nearest Neighbors classifier. The considered neighborhood k was set to $[3, 25] = \{k \in \mathbb{N} | 3 \leq k \leq 25\}$. Setting k to a higher value would lead to a strong dominatation of the neighborhood by frequent classes. With a train-/validation split of $80 : 20$ we got our best results on a combination of texture, structure and color features. The following values show that the rare substrates are basically not found. In this way, we did not succeed in improving these values.

2.2 Augmentation

The amount of given data is remarkably low. Usually even a pre-trained neural network requires a larger data set, why we decided to use data augmentation. To generate new data, we used the following methods: noise and blur[1]. Other augmentation methods did not seem practical, since it would change bounding boxes. Therefore, we generated a second dataset (B) and could triple the data set size. Within the new dataset, which consists of substrate bounding boxes, we kept the class distribution, due to the probability of finding a frequently represented substrate type is significantly higher than that of a rare one. Also because balancing the dataset would require to cut frequent substrate types, which did not seem appropriate regarding the low number of annotated corals.

2.3 Sharpening

The images vary in quality and many of them are out of focus or blurry. To counter this and to create an improved dataset (C), we increased the contrast of entire images and highlighted the details. For this purpose, each pixel value was replaced by the weighted average of its 3×3 neighborhood. The following matrix shows the filter:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

3 Approaches

The challenge of the annotation and localization task is to find corals within images of sea ground, define bounding boxes for each coral and label them with their specific type of substrate.
We applied one state-of-the-art deep learning approach and additionally developed an own one. These two are presented in the following subsections, whereas the focus lies on explaining our own approach.

3.1 YOLO

In contrast to comparable neural networks, like "fast R-CNN" [4], which locate and classify objects multiple times for various regions of an image, the YOLO architecture passes the whole input image at once. That is achieved by dividing each image into square cells inside of which bounding boxes are predicted. In our work we scaled input images to a size of 608 x 608 pixels, because of the many corals contained in each image. This is the largest resolution we tested on our GPU and was the most promising. The classification process is basically a regression problem, which leads from image pixel values to bounding boxes with their class probabilities in one go. Part of the training is the optimization of predicted class probabilities, which defines the bounding boxes. In doing so, the calculation of each box considers features of the entire image. Therefore YOLO has the advantage of making less background errors as R-CNN, because more context information is taken into account. YOLO also outputs a confidence, which is calculated as the product of the precision of an object and its intersection over union (IoU). In a later step, this is multiplied by the conditional class probability of an object. Finally an output confidence is obtained, which describes how probable the particular class of the box is and how well the predicted bounding box fits this particular object.

Limitations However, there are some limitations. On the one hand each cell of the grid predicts only two boxes, which share the same class label. This is an algorithmic limitation on the number of objects with different labels, if the objects are close to each other. On the other hand the authors of YOLO mention that they treat errors in small bounding boxes the same way they treat large bounding box errors. Because of that, errors in small boxes have a larger impact on IoU, which leads to incorrect localization.

3.2 Own Developments

We developed a two-staged approach that first locates and then labels the substrates. Both of these steps make use of machine learning, to be more precise classification algorithms. This leaves room to improve the classification task, e.g. by evaluating different classification algorithms.

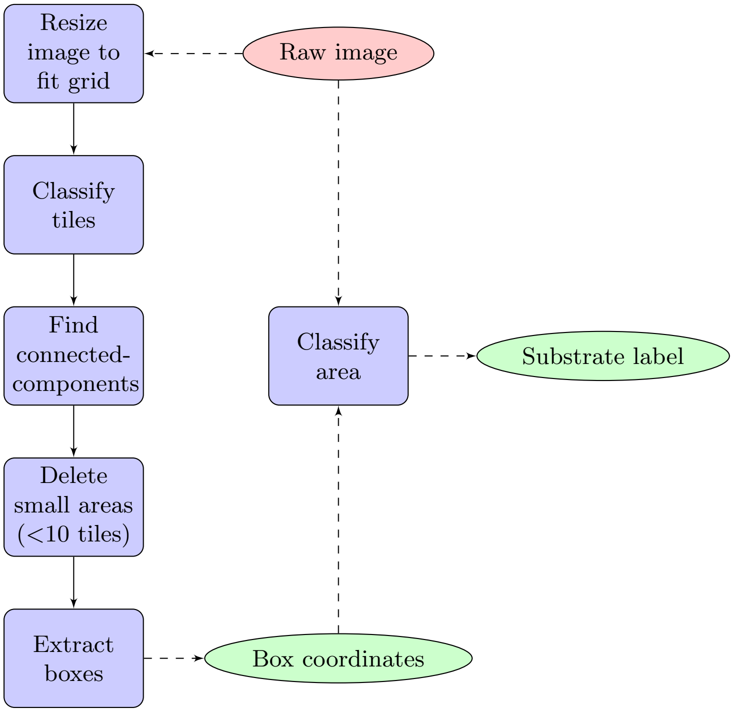


Fig. 1: **Flow diagram of our approach.** Red clouds are input data and green ones output data. The blue boxes are steps described in text.

One advantage of this two-staged approach is, that the two stages are independent from each other, which makes it possible to combine different algorithms and approaches together. The algorithm of our approach is shown in figure 1.

Locating Substrates The main idea behind locating substrates is based on the assumption, that the coral images have coral and non-coral areas. Such non-coral areas should look relatively similar for all coral types. This is quite different for images showing objects like cars or birds. Following our assumption, we segmented an image in coral and non-coral areas. First the image is divided in a grid, small enough to predict all boxes. To classify these areas we used a grid and then extract features from the tiles of this grid. We used a square of a fixed size for the tiles, which is based on the size of the smallest boxes in the training set, i.e. the integer average of the smallest width and height. Based on the training set, we recommend to use a tile size of 12×12, so that the smallest box can be located completely without background. To ensure that all tiles of an image have the same size, i.e. image size is the whole multiple of the tile size, the image is scaled to fit.

Next we extracted features for every tile of each of the training set images. For this purpose we used concatenated feature vectors consisting of features describing the color, texture and shape. For color, normalized histograms are used which describe the characteristics regarding the color [10]. The textural features of the tiles are modeled by Haralick texture features [5], which applies co-occurrence matrices on the gray scale level. Lastly the shape is represented by Hu moments [7]. Hu moments are invariant to translation, rotation and scale. All of these characteristics are useful for the domain of coral images.

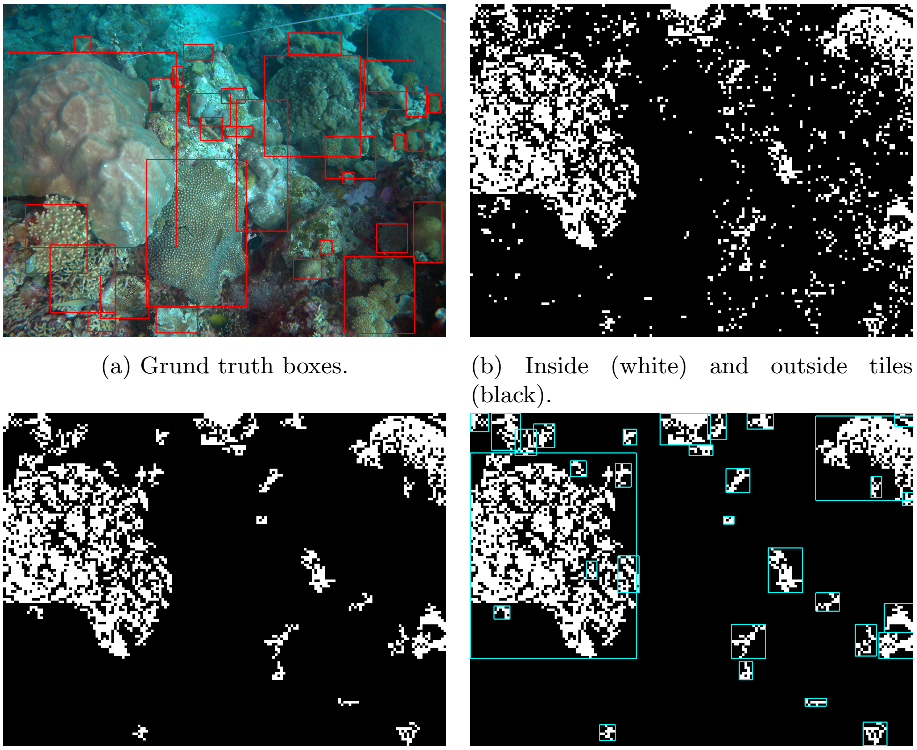
These features are used to train a binary classifier, which classifies whether a tile is a coral area or a non-coral area. An area of a training set image is considered a coral area, if more than 50% of its area intersects with a bounding box area of the ground truth. To classify areas of images that should be predicted, this image is also divided into tiles of the same previously defined size. Now the labels were obtained by feeding the features into the learned classifier. We decided to use K-Nearest-Neighbor with $k = 15$ to classify the tiles, because $k = 15$ performed best on our validation split.

After each tile of the grid was classified, we got an black and white image with 12×12 pixel large tiles. There are multiple strategies to extract boxes out of the resulting picture. We used a relatively naive approach with the application of connected-component labeling. Since we discovered a large amount of single, not connected tiles, we only kept components, that consisted of more than ten tiles. This counters a less beneficial performance of our classifier.

Each unique component is now bordered with a bounding box, that borders the outside tiles of the component. Figure 2 is showing the different stages of the location process (b - d), as well as the ground truth (a).

Labeling Found Boxes The found bounding boxes were classified on previously mentioned features 2.1 using a k-Nearest Neighbors Classifier. In addition to this already presented classification approaches, we studied whether the features can also be classified using a convolutional neural network. For the research, we subdivided the training data into a training and validation set in a ratio of 80:20 as previously. For comparability of the results we scaled the input data to the size of our grid. In consideration of the low amount of image data, we begun our work with a correspondingly small CNN, which we call baseline. The given CNN consists of one convolutional layer with maxpooling and rectified linear activation. We use dropout to prevent overfitting. The deactivation of neurons happens with a 20% probability. Subsequently, the data is handed to a flattening layer which serves as connection between convolutional and following dense layers. The result first enters a dense layer with RELU as the activation function and is then passed on to a density layer with softmax as activation function. This leads us to a confidence for each bounding box to belong to one of our 13 classes.

Considering that such a simple architecture may not be able to "remember" all relevant features of coral images, we extended our baseline architecture. Therefore we enlarged the existing architecture with two additional convolutional hid-



(a) Grund truth boxes.

(b) Inside (white) and outside tiles (black).

(c) Refined inside and outside areas.

(d) Bounding boxes of connected components.

Fig. 2: Visualized process of localization corals with our approach. The raw picture is taken from the ImageCLEFcoral dataset [2].

den layers.
Finally we looked for an extra deep architecture for comparison. All networks were trained with a batch size of 100 and with up to 1000 epochs. We decided to use VGG19 [12] and trained it on our data via transfer learning, since it has been proven to be gold standard in recent years.

4 Evaluation and Results

The following section discusses the submitted runs at ImageCLEF 2019. For a better understanding of the results of the approaches, we evaluated the localization and labeling separately. The results show that YOLO is considered state-of-the-art for a reason.
Besides presenting our results of the submissions, we also discuss the limitations and potentials of our approach as well.
In table 2 we present the results of our submissions. Our own approach is marked

with I, and YOLO based submissions with II. MAP_0.5 stands for the localised mean average precision for each submitted method with an IoU ≥ 0.5 of the ground truth and R_0.5 for the recall value, respectively MAP_0 represents the image annotation average without any localization. The results for I show, that CNNs and k-NN deliver comparable results. Sharpening has not led to better results, perhaps because it accentuates noise. YOLO combined with statistical probability distribution provides best results with an precision of 0.243 and a recall of 0.131.
Table 2 shows that we worked only on data set (A) and (C). We did not use data set (B) for our run submissions, since it did not lead to any kind of improvement.

Table 2: **Results of our submitted runs at ImageCLEF 2019.** Comparison of the results of the different approaches in our submissions. The methods used in I are our developed two-staged approach, whereas II are approaches using YOLO.

	Approach	Dataset	MAP_0.5	R_0.5	MAP_0
I	k-NN, k = 13	A	0.003	0.004	0.272
	Statistical labeling	A	0.002	0.003	0.203
	Baseline CNN	A	0.003	0.004	0.228
	Transfer Learning	A	0.003	0.004	0.291
	3-Layer CNN	A	0.003	0.004	0.205
II	YOLO + k-NN	A	0.229	0.131	0.500
	YOLO + Statistical	A	**0.243**	**0.131**	**0.488**
	YOLO + k-NN	C	0.210	0.122	0.455
	YOLO + Statistical	C	0.220	0.122	0.442

All approaches we used have some limitations and therefore leave space for improvement. Some of which we will describe in the following.

YOLO The weakness of YOLO is evident on rather smaller bounding boxes. Predictions on the validation data set showed that small coral substrates are either not found or subsequently labeled incorrectly. This results in the low recall value of 0.131. Corals that are found however, are mostly labeled as "c_soft_coral". Nevertheless, even on larger corals, YOLO shows rather moderate results. In a quarter of images it did not find boxes at all, that is why we used the found boxes from our other approach I to complete the results.

Our Approach Not only the performance (see Table 2) shows flaws in our approach, but also some obvious conclusions do. Since we got an accuracy of 0.534 on labeling boxes, which was evaluated on a 80 : 20 split of the training set, we assume that our approach fails to locate corals correctly. We also tested using SIFT features which had an accuracy of 0.4744.
One problem of the two-staged approach is the assumption, features of coral and

non-coral tiles are distinct enough. This leaves room for further evaluation and research, regarding the choice of features and labels. It might be beneficial to use more than two labels, i.e. more than just coral and non-coral. This could e.g. be water in the background, because we discovered that water in the background is often ”false positive” classified, i.e. as coral area. An additional label would also need an additional annotation.
The question arises, whether 14 (13 substrate classes + background) labels could be used. This approach would only need one, instead of two classifiers.
Regarding the tile classification, the usage of CNNs to classify the tiles sounds promising because of the high number of tiles.
Another issue with our approach is the size of the tiles. Big tiles prevent small boxes from getting found and increase the chance of two corals in one tile. From a design perspective, boxes should be as small as possible to be as precise as possible. But if tiles are chosen relatively small, not only does the computational time extend, but features contain less information. This could lead e.g. to forms not getting recognized. We encountered the problem of an enormous computational time, because of that we increased the size to 24 × 24. Also we reduced the training set of tiles by 80%, which decreases the performance not significantly as seen in table 3. An approach of using a sliding window should also be considered in future work.

Table 3: Performance of 20% of the training set tiles compared to all tiles.

Amount of dataset	Precision		Recall		F_1 Score	
	Non-coral	Coral	Non-coral	Coral	Non-coral	Coral
1.0	0.6916	0.4950	0.7664	0.4013	0.7271	0.4433
0.2	0.6907	0.4900	0.7603	0.4034	0.7238	0.4425

Lastly using connected components as the method to extract boxes from the tile images, could be not sophisticated enough. Firstly with a perfect labeling of coral and non-coral tiles, it would not be able to recognize inlying boxes. And secondly it only considers two labels as features. The use of density-based clustering, working on more than just the predicted labels could lead to better results.

5 Conclusion

Overall, our approaches show moderate results. The idea to use neural networks proved to be promising. However, afterwards we can assert that YOLO was not the best choice. It completely fails to find smaller bounding boxes.

The concept of using feature engineering and searching for features or feature constellations, which are able to describe and represent different types of benthic substrate, still seems to be useful regarding the small amount of given data. But there is a lot of room for improvement.

Beside of that there are multiple images in the data set that show the same sea ground and contain for the most part the same corals. This kind of information can be used locally to improve the bounding boxes of corals, since their position can be tracked.

With regard to our approach 2b of labeling coral and non-coral areas, we can make the conclusion that the chosen features are not working properly. Probably we need a kind of back propagation to mark wrong labeled areas and process images multiple times. Additionally we could investigate the set of our features for a more performant subset using boosting. We would also stick to the deep learning approach and try another, maybe more time consuming but also more precise neural network, like an R-CNN.

Finally, the concept of combining deep learning and classic feature engineering is where we see the most potential.

Besides that, another point of potential improvement is the correction and balancing of the data set itself. Currently, seven of 13 coral type classes have a relative ratio of less than two percent, six out of them even less than one percent. The quality of the pictures is very variable too. Some of the images do not even seem to be completely annotated.

For future approaches, we would recommend publishing a larger and more balanced data set, in which each class has almost the same number of representatives.

To address the initial question whether an automatic localization and annotation of corals is feasible, we see good chances for future research.

References

1. Bloice, M.D., Stocker, C., Holzinger, A.: Augmentor: an image augmentation library for machine learning. arXiv preprint arXiv:1708.04680 (2017)
2. Chamberlain, J., Campello, A., Wright, J.P., Clift, L.G., Clark, A., García Seco de Herrera, A.: Overview of ImageCLEFcoral 2019 task. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org (2019)
3. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark (2009)
4. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
5. Haralick, R.M., Shanmugam, K., et al.: Textural features for image classification. IEEE Transactions on systems, man, and cybernetics (6), 610–621 (1973)
6. Hoegh-Guldberg, O., Mumby, P.J., Hooten, A.J., Steneck, R.S., Greenfield, P., Gomez, E., Harvell, C.D., Sale, P.F., Edwards, A.J., Caldeira, K., et al.: Coral reefs under rapid climate change and ocean acidification. science **318**(5857), 1737–1742 (2007)
7. Hu, M.K.: Visual pattern recognition by moment invariants. IRE transactions on information theory **8**(2), 179–187 (1962)
8. Ionescu, B., Müller, H., Péteri, R., Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Lux, M., Gurrin, C., Cid, Y.D., et al.: Imageclef 2019: Multimedia retrieval in lifelogging, medical, nature, and security applications. In: European Conference on Information Retrieval. pp. 301–308. Springer (2019)

9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (Nov 2004)
10. Pass, G., Zabih, R., Miller, J.: Comparing images using color coherence vectors. In: ACM multimedia. vol. 96, pp. 65–73. Citeseer (1996)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)