Group work 1

(Adam Harnúšek & Samuel Dubovec)

Task n.1 Preprocessing

We have manually searched for images online. Images of 2 benthic species in Baltic Sea, specifically **Furcellaria lubricalis** and **Zostera marina**. Unfortunately, we were only able to find around 13 images per kind, but as we were told the results weren't that important so we decided that this is enough, and we could spend more time on other tasks. Then we wrote simple bash script (`./converteach.sh`) to convert all images to same size (1024 x 1024) as was recommended in task description.

Task n.2 Etalons

In this task we have chosen etalons by hand, we considered more options, but Rene advised us this approach. That means that we cropped small parts from each image in our dataset. We tried different sizes of etalons, but we came to conclusion that smaller is better. Then we used `cv2.matchTemplate()` function to evaluate each etalon. For our final set of etalons, we took only the best ones. Our selected set had to satisfy condition from task description of being representative on at least 50% of the images.

One of Zostera etalon

One of Furcellaria etalon

Task n.3 Baseline

In this task we used SIFT and ORB algorithms to detect features on our dataset and match them with set of etalons. In case of performance SIFT gave us better result. SIFT found more reasonable keypoints than ORB.

In the file `./sift_matches` we can see number of good matches on every furcellaria image we have. We matched all etallons on each furcellaria image.

For filtering only good keypoint matches we used Lowe's ratio test. This test works in following way, each keypoint of the image is matched with a number of keypoints from the etalon. We keep the 2 best matches for each keypoint (best matches = the ones with the smallest distance measurement). Lowe's ratio test checks that the two distances are

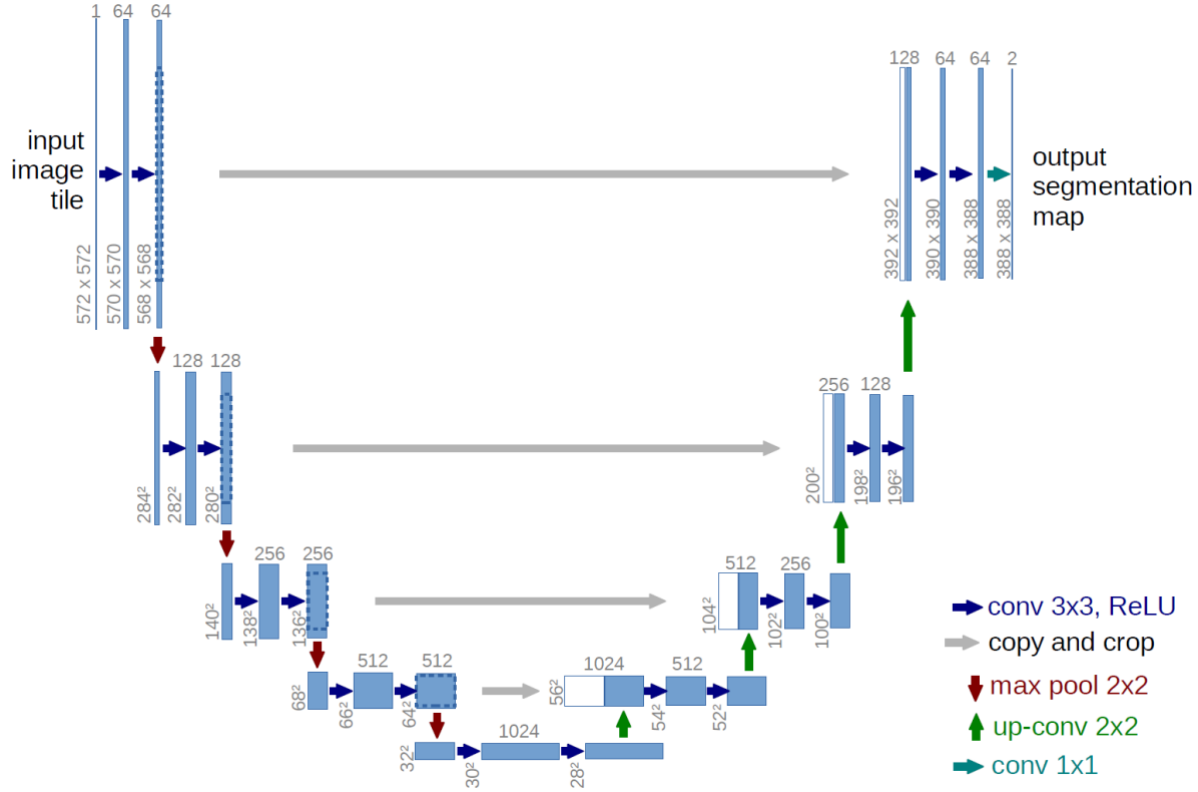
sufficiently different. If they are not, then the keypoint is eliminated and will not be used for further calculations.

At the end we created 2 images `./task3a.tiff` represents result from SIFT algorithm and `task3b.tiff` represents result from ORB algorithm. In these pictures we can see that SIFT outperformed ORB on same picture.

Task n.4 Image segmentation

Firstly, we created annotations with CVAT tools, which was recommended to use. We used 3 classes – background, zostera, furcellaria. After finishing with annotating our dataset we exported annotated images in COCO format (.json).

For this task we have chosen UNET as our neural network model. We found this implementation having pretty good results and we also followed paper about UNET. Our implementation uses 9 layered UNET architecture, which we can see in the picture bellow.



UNET architecture

We split our train and validation data in 3:1 ratio (18 train images, 6 validation images).
In our implementation we can easily change some network params, for example: batch size, epoch, output images size, etc.

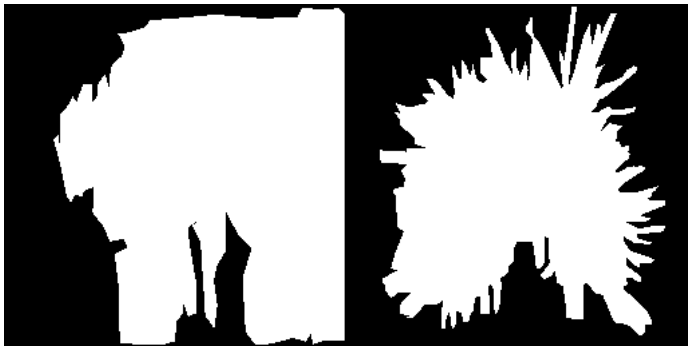
In our project we used PyTorch as library which helped us achieve our goal. We decided to use this library because of no previous experience with computer vison tools. After presentation by Rene, PyTorch looked as the right choice for us.

Our validation results are saved in (`./saved_images`) directory.

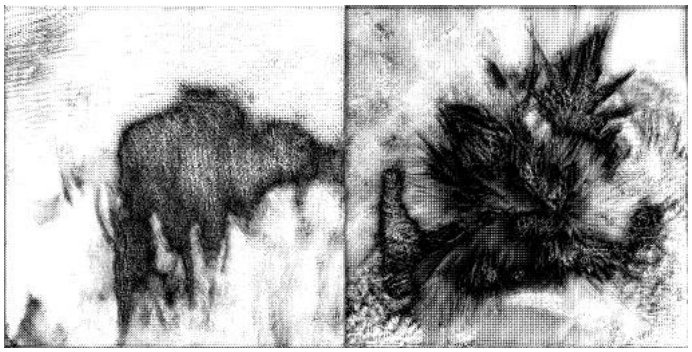
As we used feature detection in the third task and image segmentation in the fourth task, we are not able to accurately compare results of these tasks. Obviously, results are much better with CNN approach.

In this task we managed to get the highest accuracy of model around 38%.

Example of validation images



Example of validation mask



Example of our predictions – black color represents how good is our prediction. Darker color we have on image, the more accurate prediction is.