Homework -2 Computer Vision Course

ITS-8030

Muhammed Adil YATKIN

1- Data Collection and Preprocessing

First, I chose 3 different species, Zostera Marina, Fucus and Furcellaria lumbricalis, to prepare datasets. Then I collected images of different quality from google images and resized the photos to 1024x1024. You can see the original versions of the photos I have collected in the "images/Dataset" file. My dataset contains 41 Fucus, 37 Furcelleria Lumbricas and 29 Zostera Marina.

2- Etalons

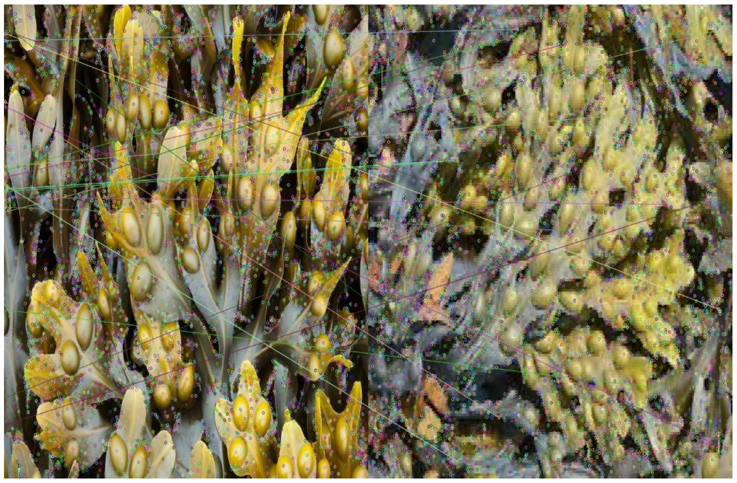
From the dataset I created, I chose small photos for each species, containing at least 50% of them and containing distinctive features. These thumbnails are representative features for at least half of the species.

3- Baseline

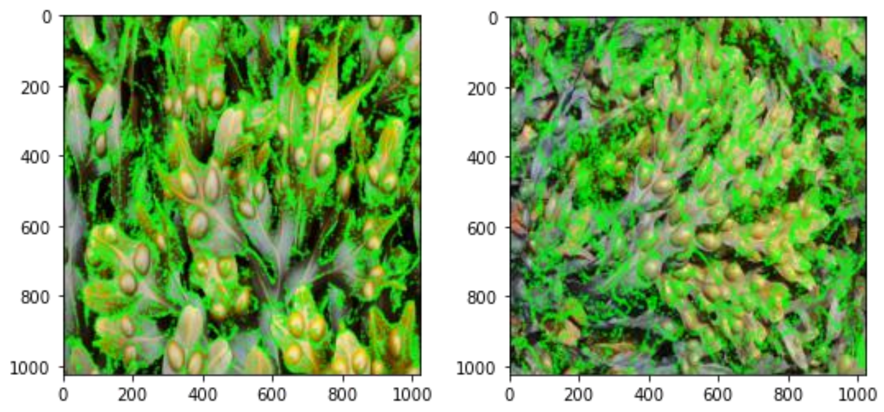
I chose three different feature extractors from Opencv: ORB, SIFT and FAST. Using these extractors, I compared two different fucus images that I selected from my dataset. Judging by Feature Matching, SIFT is clearly ahead.

ORB():

SIFT()



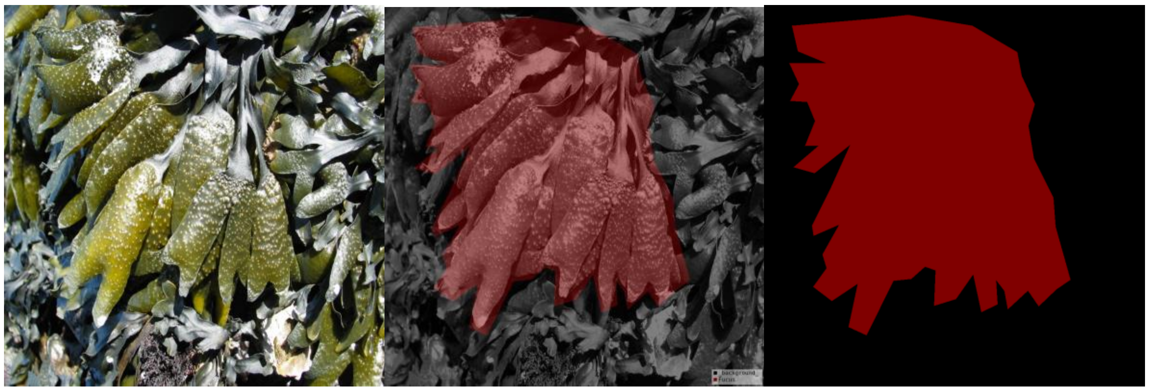
FAST()



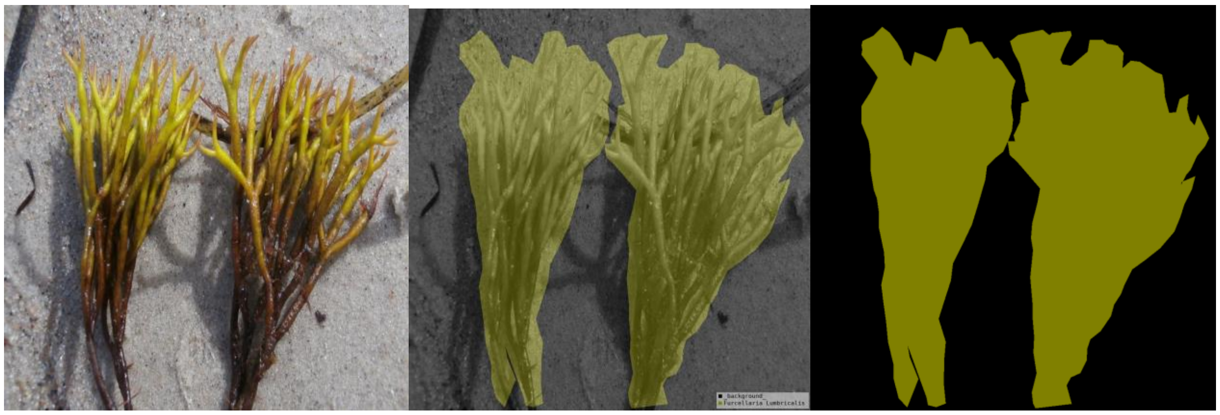
4- Semantic Segmentation

First of all, I annotated the data I had collected using the tool named labelme in accordance with semantic segmentation.

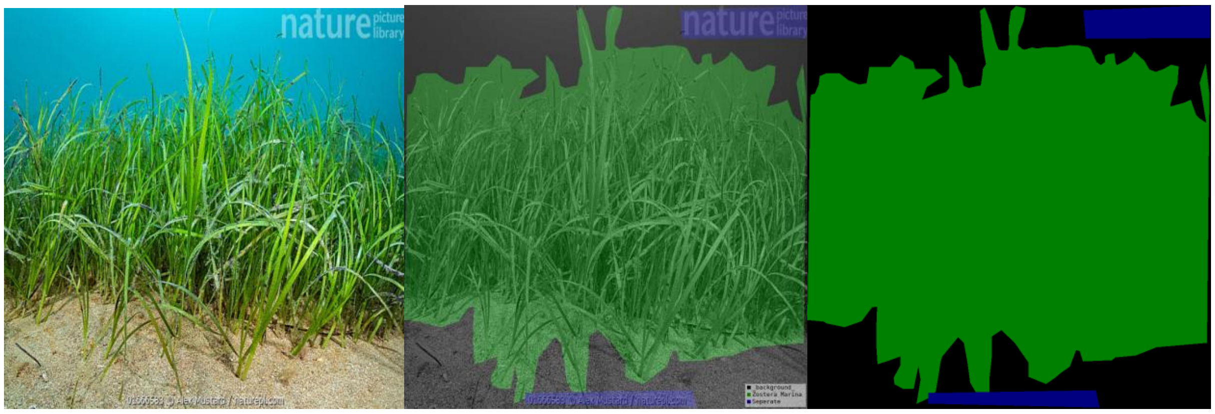
Example of annotation (Fucus):



(Furcellaria Lumbricas):

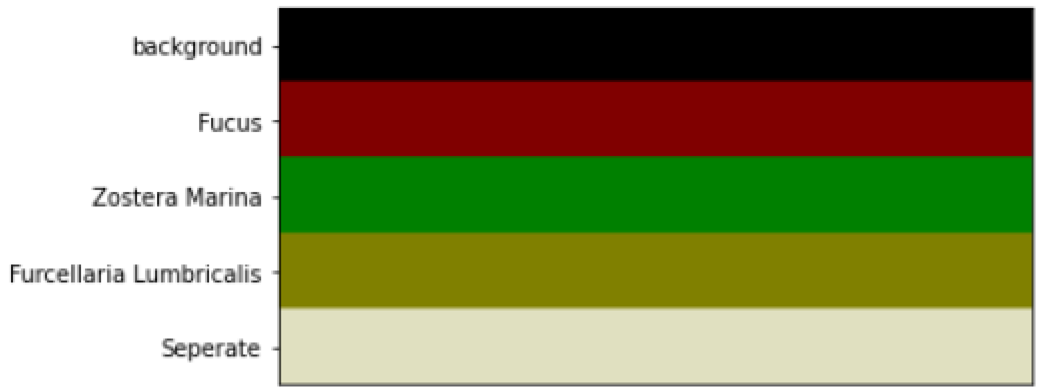


(Zostera Marina):



The annotated dataset contains 5 different classes:

```
CLASS_LABELS = ['background', 'Fucus', 'Zostera Marina', 'Furcellaria Lumbricalis', 'Seperate']
```



Seperate class represents the ignore label.

After preparing my dataset, I chose the semantic segmentation model I used. I used this paper on this subject.

I preferred the model that uses the VGG-16 backbone named "FCN-VGG16" mentioned in the paper. This model achieves a Pascal Voc Datasette meanIOU score of 56.0. I modified and trained the model in accordance with my class count.

The parameters I used during the training are as follows:

Learning_rate : 0.0001

Batch_size : 16

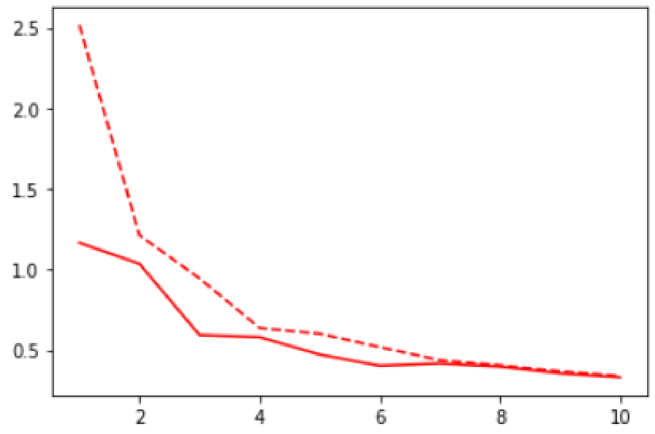
Optimizer : Adam

Loss function : Cross Entropy

Also, there were 60 images in my train dataset and 30 images in my validation dataset, and I got the results you see below:

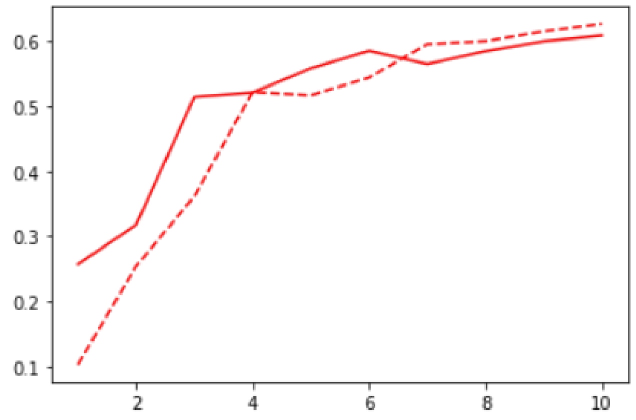
```
plt.plot(range(1,11), h2['val_loss'], '-', color='red', label='FCN32 validation loss')
plt.plot(range(1,11), h2['loss'], '--', color='red', label='FCN32 training loss')
```

```
[<matplotlib.lines.Line2D at 0x7fedd25c0910>]
```



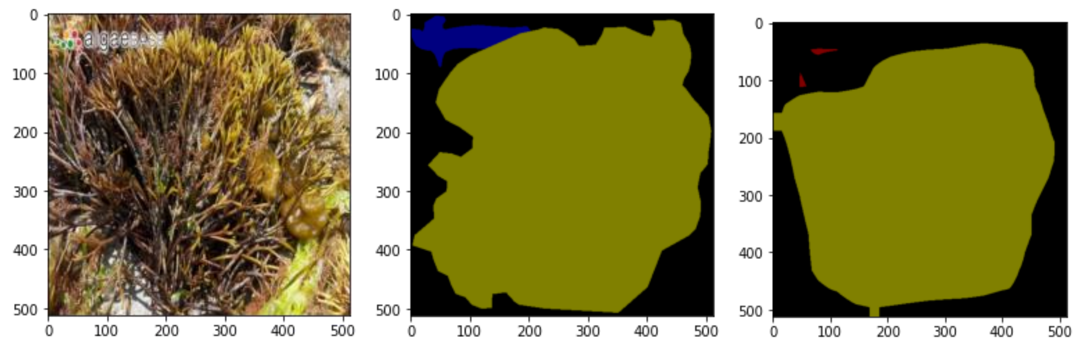
```
plt.plot(range(1,11), h2['val_meanIoU'], '-', color='red', label='FCN32 validation meanIoU')
plt.plot(range(1,11), h2['meanIoU'], '--', color='red', label='FCN32 training meanIoU')
```

```
[<matplotlib.lines.Line2D at 0x7fedd25a87d0>]
```



In the validation dataset, I reached 0.853 as pixel accuracy and 0.609 as meanIOU. Also, the pixel accuracy on the train dataset was 0.8597 and the meanIOU was 0.6266.

Examples:



```
loss: 0.4485 - crossentropy: 0.4438 - pixelacc: 0.8355 - meanIoU: 0.3481
```

I applied data augmentation techniques to improve these results I have obtained. These are Horizontal flip, translation, zoom, rotation and saturation, brightness and contrast. In this way, I increased the number of training datasets to 510. You can see the functions I used and the techniques that I applied, as documented in "Random_Augmentation.ipynb". However, due to some mistakes I made in the code, I could not train with augmented data because there was a change in the number of classes and I could not retrain the same model and compare it with the old results. I will update the results again when this problem is solved. I used google colab GPU to train the model. You can see the model in architecture that I used in the "Train_Without_Augmentation_.ipynb" code.

Method	Mean IOU Score
SIFT feature extractor	0.42492646
FAST feature extractor	0.42541698
ORB feature extractor	0.42444086
FCN-VGG16	0.6087