Kelp Wanted: Segmenting Kelp Forests
Ioannis Nasios, a.k.a. Ouranos

For this competition, from processed Landsat satellite image chips we had to develop an algorithm capable of creating a kelp forest mask. To confront this task, a training dataset was provided which included annotations for every training image and semantic segmentation AI models were trained. Unfortunately as these annotations came directly from a community platform where citizen scientists labeled each image by hand, this dataset contains a lot of noise and some missannotations. Furthermore some images seem to have a shifted mask related to true mask, probably due to some error during the creation of small image/mask chips out of a large EO product. As this was the case for the test set too, modelling didn't get influenced by this but if someone wanted to use this in a real case scenario, would be best to work with a curated dataset.

All train and test data contained 7 channels, SWIR, NIR, Red, Green, Blue, Clouds and Altitude. Out of these only 3 were used in training SWIR, NIR, and Green. Also, the Altitude band was used in post processing, for removing kelp predictions on land.

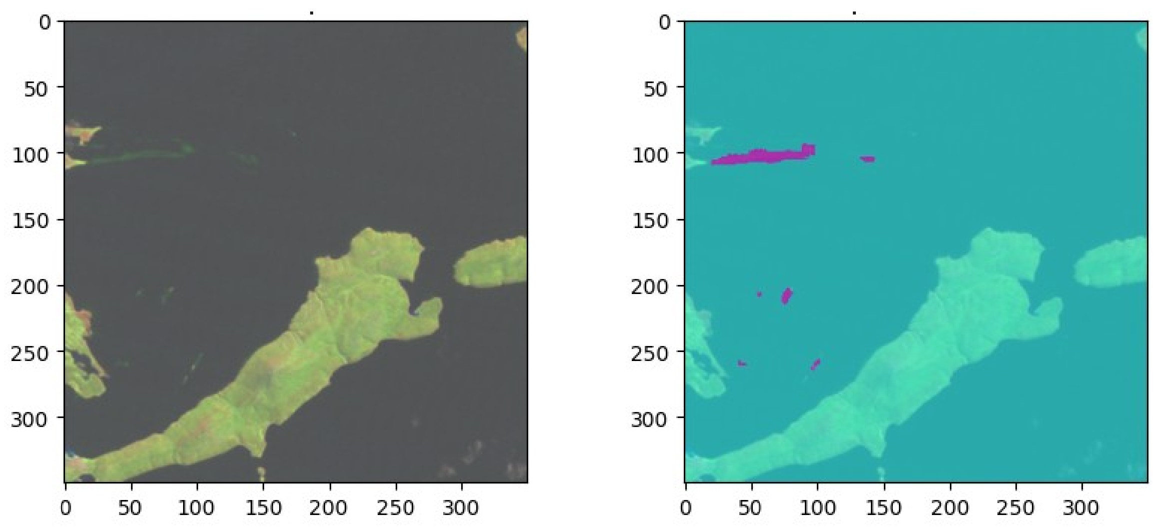


Figure 1: Landsat image chip on the left, the same image with mask overlay on the right

System

All experimentation was run on the main server as well as most of the models that was used in the final solution. Google Colab pro was only used for scaling up the solution by running the 2 of the final models in a higher image dimension.

Characteristic	**Main Server**	**Google Colab pro**
Operating System	Ubuntu 18.04	Ubuntu 22.04
Cuda	11.2	12.2
Python	3.8	3.10
Pytorch	1.8	2.1
GPU	GeForce GTX 1080	Tesla V100-SXM2-16GB
RAM	70GB	50GB

Data processing

As data values were integers ranging from 0 to 65536 with the value -32768 representing missing data, for feeding the model all data were clipped to range (6000, 24000). Missing values replaced with zero and then all divided to with 24000 to be in (0-1) range. Finally, from all images, the mean imagenet value was subtracted and then divided by the imagenet standard deviation(reduced). (img01 = (img01 - np.array([0.485, 0.456, 0.406]))/(0.9*np.array([0.229, 0.224, 0.225])))

Augmentation

For data augmentation the albumentation library was used. Vertical and horizontal flips as well as 90 degrees rotation was used (all with 50% chance). A few models also included a custom augmentation for holes in both the image and the mask (chance 25%). For all models test time augmentation (TTA) was used with all 4 possible flips.

Modelling

The training of segmentation models was done using the pytorch framework and specifically the segmentation_models_pytorch (SMP) and the transformers libraries.

Library	**Architecture**	**Backbone**	**Image size**	**weight**
SMP	Unet	mit_b1	768	1
SMP	Unet	mit_b4	512	1
SMP	Unet	mit_b3	640	1
SMP	Unet	mit_b2	640	1
SMP	Unet	mit_b1	1024	1.5
Transformers	Uppernet	convnext-tiny	512	1
Transformers	Uppernet	convnext-tiny	768	1
Transformers	Uppernet	convnext-base	512	1
Transformers	Uppernet	convnext-tiny	896	1.5

Final predictions

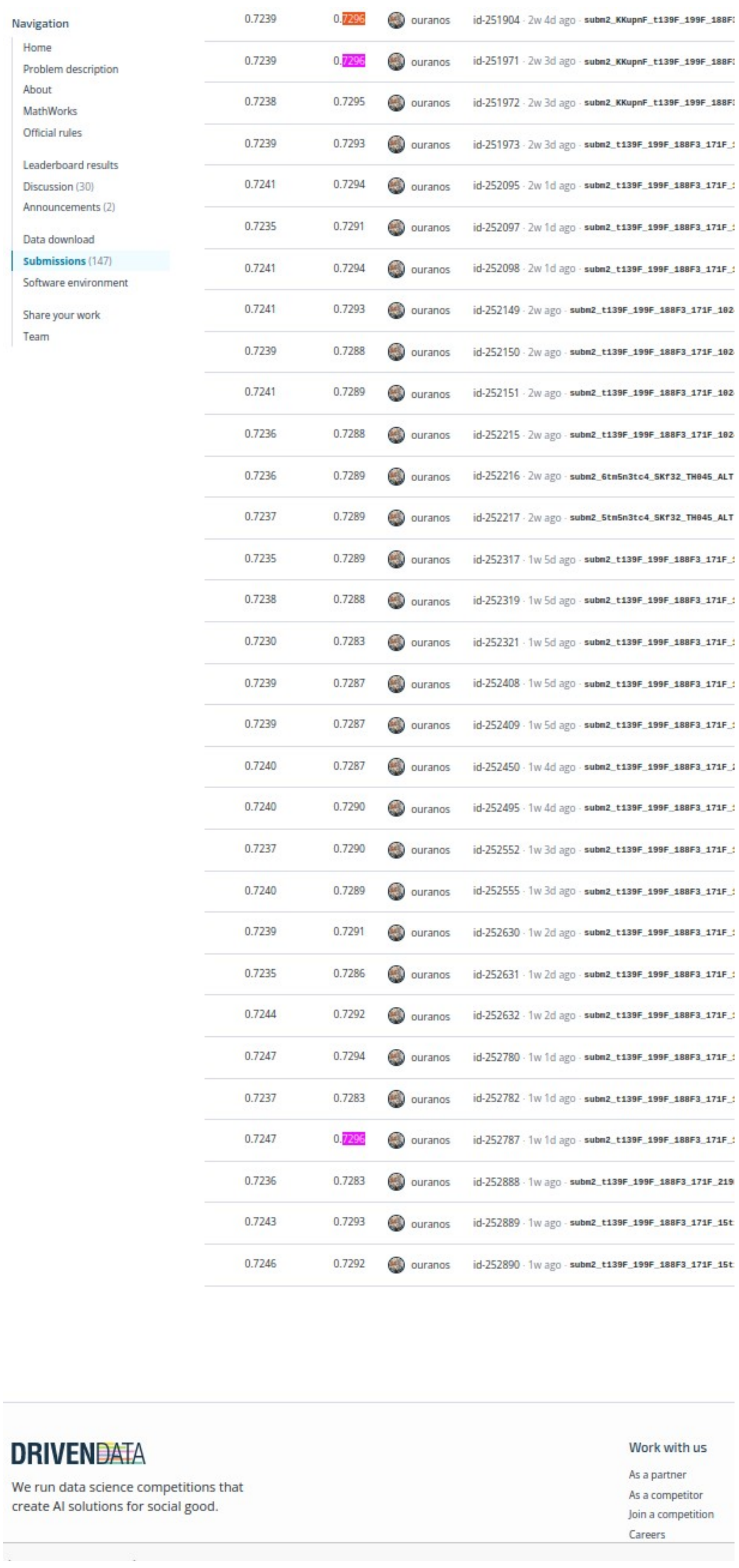
Final prediction is a weighted average of models in table above. Then, a sea-land mask was applied, derived from last image channel, to remove predictions located on land. Finally, a threshold of 0.43 was used to turn probabilities to mask as this was optimal in the validation set.

Also tried

Thing that tried but didn’t make it to final ensemble.
◦ Mixup and cutmix augmentation,
◦ Chromatic and other augmentation
◦ Other segmentation architectures
◦ Different loss functions
◦ Using indexes instead of channels
◦ Other encoders

Appendix

More than 40 submissions was scored within the same final rank. I am proud that my solution proved to be robust and scored best in all validation, public and private test sets. Furthermore, I believe that including many models in the final ensemble improves the stability of the outcome and eliminates the lucky seed effect, making the solution more robust.



0.7239	0.7296	ouranos	id-251904 · 2w 4d ago · subm2_KKupnF_t139F_199F_188F
0.7239	0.7296	ouranos	id-251971 · 2w 3d ago · subm2_KKupnF_t139F_199F_188F
0.7238	0.7295	ouranos	id-251972 · 2w 3d ago · subm2_KKupnF_t139F_199F_188F
0.7239	0.7293	ouranos	id-251973 · 2w 3d ago · subm2_t139F_199F_188F3_171F_
0.7241	0.7294	ouranos	id-252095 · 2w 1d ago · subm2_t139F_199F_188F3_171F_
0.7235	0.7291	ouranos	id-252097 · 2w 1d ago · subm2_t139F_199F_188F3_171F_
0.7241	0.7294	ouranos	id-252098 · 2w 1d ago · subm2_t139F_199F_188F3_171F_
0.7241	0.7293	ouranos	id-252149 · 2w ago · subm2_t139F_199F_188F3_171F_102
0.7239	0.7288	ouranos	id-252150 · 2w ago · subm2_t139F_199F_188F3_171F_102
0.7241	0.7289	ouranos	id-252151 · 2w ago · subm2_t139F_199F_188F3_171F_102
0.7236	0.7288	ouranos	id-252215 · 2w ago · subm2_t139F_199F_188F3_171F_102
0.7236	0.7289	ouranos	id-252216 · 2w ago · subm2_6tm5n3tc4_SKf32_TH045_ALT
0.7237	0.7289	ouranos	id-252217 · 2w ago · subm2_5tm5n3tc4_SKf32_TH045_ALT
0.7235	0.7289	ouranos	id-252317 · 1w 5d ago · subm2_t139F_199F_188F3_171F_
0.7238	0.7288	ouranos	id-252319 · 1w 5d ago · subm2_t139F_199F_188F3_171F_
0.7230	0.7283	ouranos	id-252321 · 1w 5d ago · subm2_t139F_199F_188F3_171F_
0.7239	0.7287	ouranos	id-252408 · 1w 5d ago · subm2_t139F_199F_188F3_171F_
0.7239	0.7287	ouranos	id-252409 · 1w 5d ago · subm2_t139F_199F_188F3_171F_
0.7240	0.7287	ouranos	id-252450 · 1w 4d ago · subm2_t139F_199F_188F3_171F_
0.7240	0.7290	ouranos	id-252495 · 1w 4d ago · subm2_t139F_199F_188F3_171F_
0.7237	0.7290	ouranos	id-252552 · 1w 3d ago · subm2_t139F_199F_188F3_171F_
0.7240	0.7289	ouranos	id-252555 · 1w 3d ago · subm2_t139F_199F_188F3_171F_
0.7239	0.7291	ouranos	id-252630 · 1w 2d ago · subm2_t139F_199F_188F3_171F_
0.7235	0.7286	ouranos	id-252631 · 1w 2d ago · subm2_t139F_199F_188F3_171F_
0.7244	0.7292	ouranos	id-252632 · 1w 2d ago · subm2_t139F_199F_188F3_171F_
0.7247	0.7294	ouranos	id-252780 · 1w 1d ago · subm2_t139F_199F_188F3_171F_
0.7237	0.7283	ouranos	id-252782 · 1w 1d ago · subm2_t139F_199F_188F3_171F_
0.7247	0.7296	ouranos	id-252787 · 1w 1d ago · subm2_t139F_199F_188F3_171F_
0.7236	0.7283	ouranos	id-252888 · 1w ago · subm2_t139F_199F_188F3_171F_219
0.7243	0.7293	ouranos	id-252889 · 1w ago · subm2_t139F_199F_188F3_171F_15t
0.7246	0.7292	ouranos	id-252890 · 1w ago · subm2_t139F_199F_188F3_171F_15t

DRIVENDATA

We run data science competitions that create AI solutions for social good.