

DataCamp

Mars-Craters Detection

Damien GRASSET

The way I tried to do it with a limited GPU

Table of contents

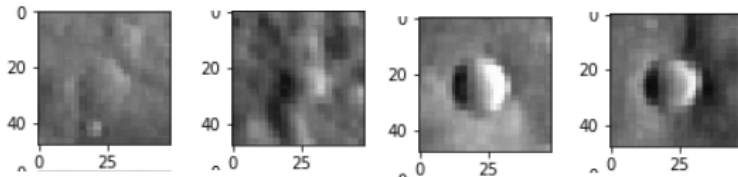
1. First Try : Sliding Windows
2. Second Try : Mask
3. Improvements

First Try : Sliding Windows

Sliding Windows

Run a classification **Is it a crater? Yes/No** on small images (depending on the crater's radius)

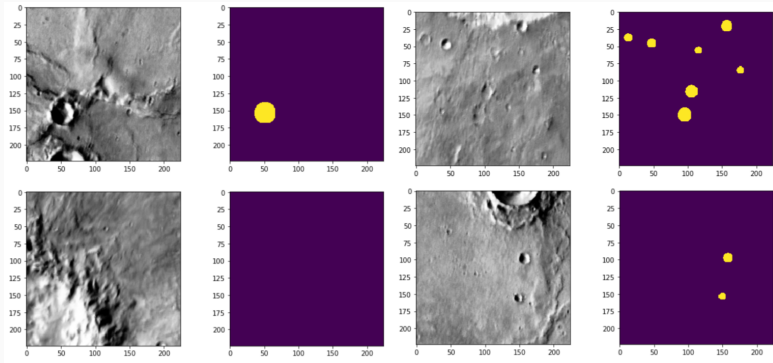
1. Extract images of craters and Generate new ones. Extract as well non-crater images to build a balance dataset.
2. Process images with pre-trained VGG-net neural networks to get vectors of size (1,512) or (1,1024).
3. Run a RandomForest Classifier the balanced dataset. (3 classifiers for 3 different sizes of the craters).



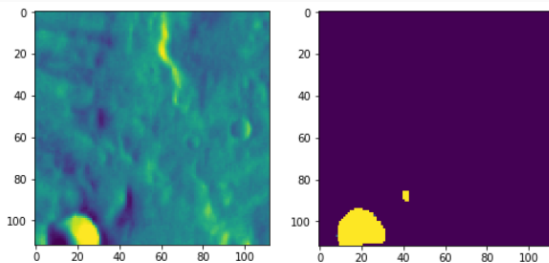
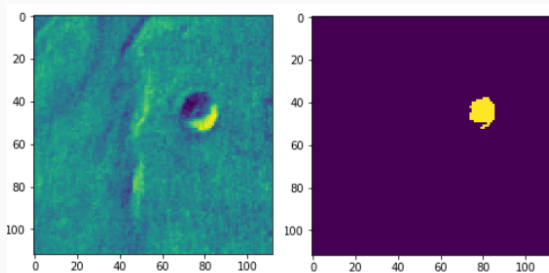
Second Try : Mask

Mask (U-Net)

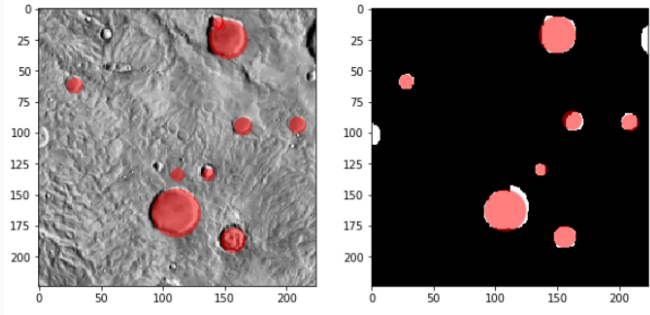
Use of U-Net algorithm to get the mask of an image : return an image of the same size using image segmentation



Mask Prediction



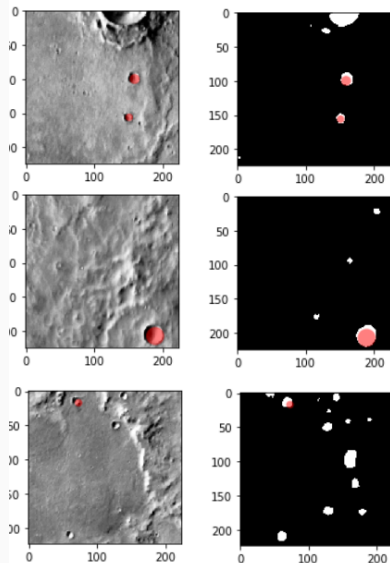
DETECTION CIRCLES WITH CIRCULAR HOUGH TRANSFORM.



Good Recall :

submission	ap	prec(0.5)	prec(0.9)	rec(0)	rec(0.5)	rec(0.9)
	⬆️⬆️	⬆️⬆️	⬆️⬆️	⬆️⬆️	⬆️⬆️	⬆️⬆️
zorro_recall_yx	0.143	0.153	0.106	0.906	0.844	0.583

DRAWBACK OF USING MASK

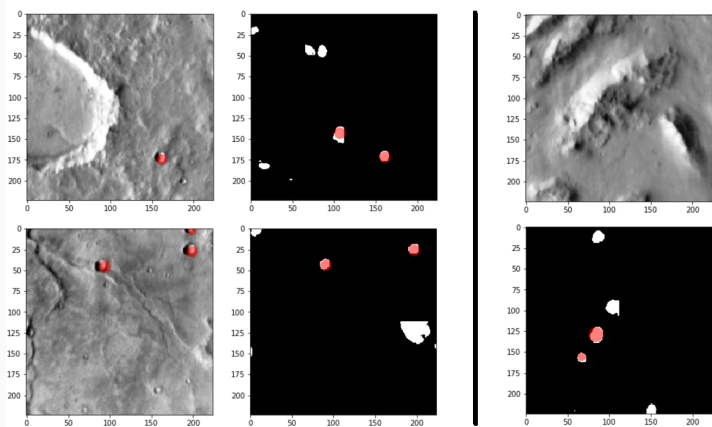


Improvements

IMPROVING BY MERGING MASK WITH CRATER CLASSIFICATION

Idea : applying the classification used in the sliding-window method so as to improve the precision of the mask method

→ only between 5 and 20 small images to test on each image (instead of 2000).



Locally on Xtrain and Xtest provided with the Starting-Kit :

1. Average Precision ~ 0.47
2. Precision(0.5) ~ 0.40
3. Recall(0.5) ~ 0.60

But inconsistencies in the mask training, sometimes the mask is not learning at all for some obscure reasons and return a grey image (0.5 instead of 0 and 1) which makes the prediction fail.

Questions
