CraterNet

A Fully Convolutional Neural Network for Lunar Crater Detection

(qglaude@ulb.ac.be) **Quentin Glaude**
University of Liège / Université Libre de Bruxelles

1. State of the Art

We can distinguish several types of craters : simple, complex or giant. Most of sub-km craters are simple. However, the description of these craters is not straightforward. The size is highly variable, craters can overlap, the nature of the soil is geographically dependent, the illumination conditions are varying and the degradation level of their rims may be low or high. It results in objects having very different visual aspects and contrast levels.

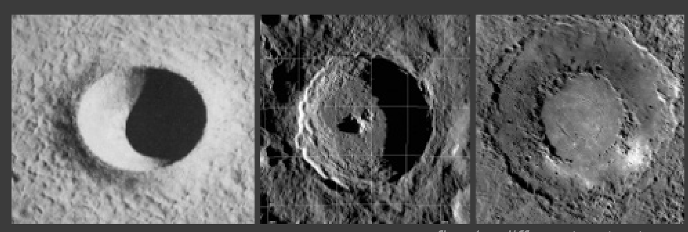
fig. 1 : differentcrater types


fig. 2 : different rims' degradation levels

There have been many attempts to develop unsupervised or supervised crater detection algorithms but there is still no unanimously accepted solution.

The recent advances in deep learning enable models to predict the label of objects based on examples. These models proved to be very efficient in various task such as semantic segmantation. All models in deep learning are based on multiple layers of neurons, forming a neural network.

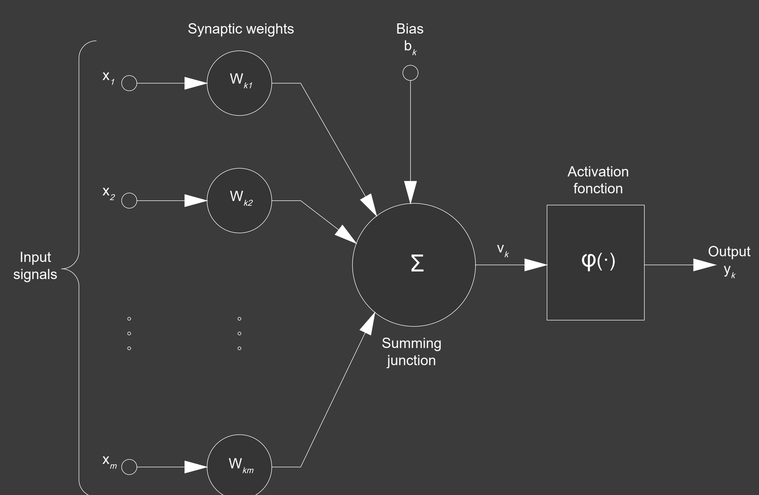

fig.3 : an artificial neuron

In image classification, convolutional neural networks and fully convolutional networks are the most used models because they are able to take the contextual information in to account.

The hypothesis developed in this master's thesis was :

A fully convolutional neural network is an innovative and convincing model for the detection of lunar craters based on remotely sensed data.

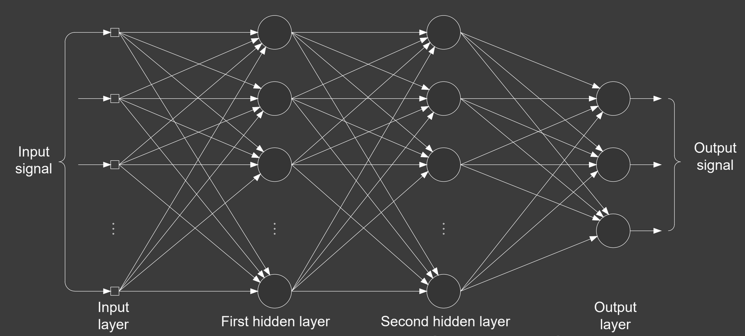

fig. 4 : an artificial neural network

2. Data

The images are provided by Narrow Angle Camera (NAC) of the Lunar Reconnaissance Orbiter. The spatial resolution of the NAC images is 0.50 m/pixel and use the panchromatic spectrum.

The data are freely available from the NASA archive. Moreover, the Arizona State University created digital elevation models using stereophotogrammetry. In the meantime they produced ortho-rectified images of several test sites using.

In total, 72 scenes were chosen by systematic random sampling on 120 degrees of latitudes, covering many illumination conditions. Note that images with a sun at very low elevation were discarded.

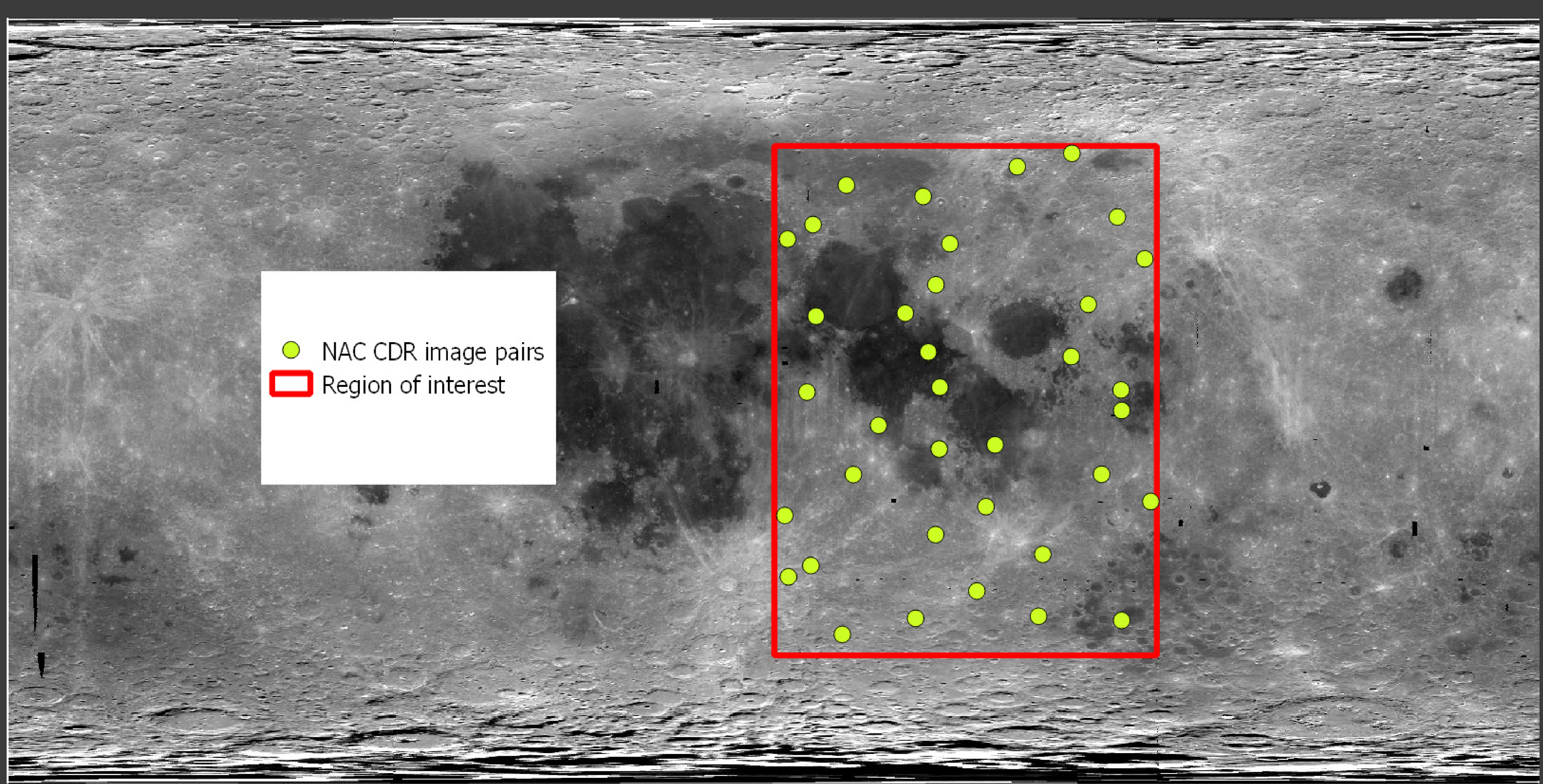

fig. 5 : locationof the 72 lunar scenes from the NAC image dataset

Deep learning needs training data to optimize its parameters. However no dataset of lunar craters existed to train our model. Hence, we created more than 11 000 annotations of craters/non-craters with the Cytomine web-designed project.

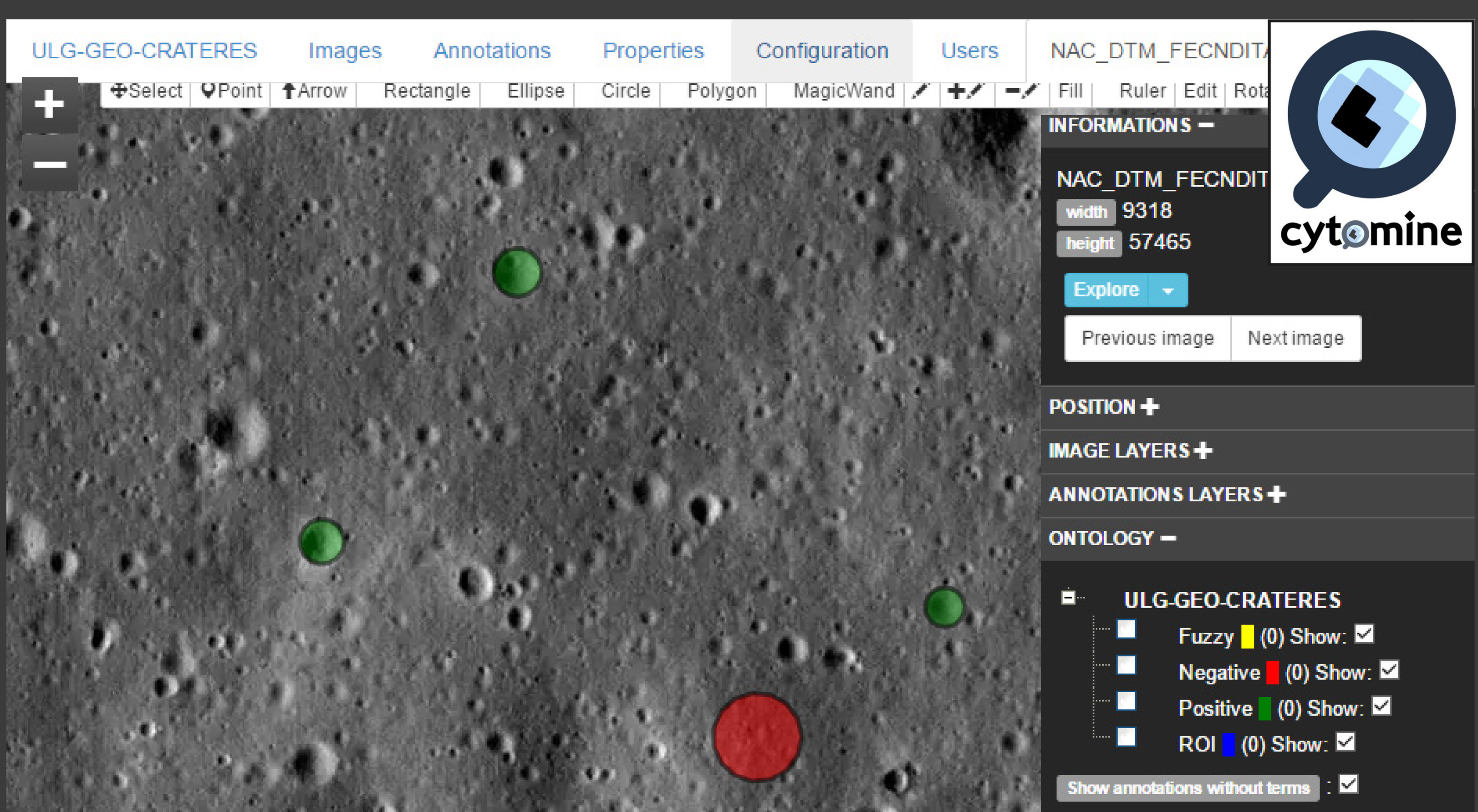

fig. 6 : creation of a lunar crater dataset in Cytomine

3. Methodology

Using the Cytomine Python API, we import the annotations from the online database. The annotations are then normalized and a set of random patches with associated ground truth is created.

These patches are separated in a train set (to optimize the parameters), a test set (to set the hyper-parameters of the model) and a validation set (to finally assess the quality of the classification).

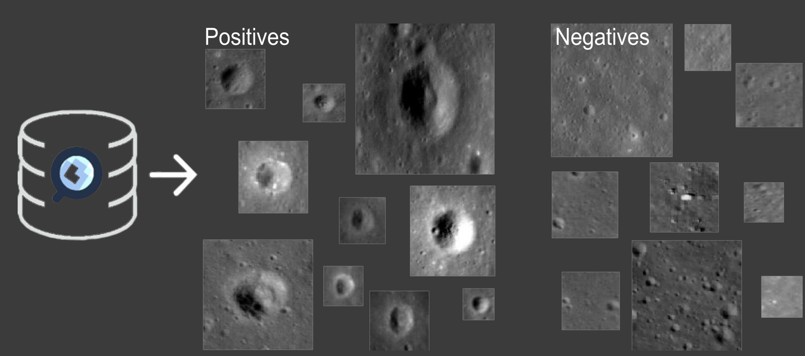

Fig. 7 : import of the annotations

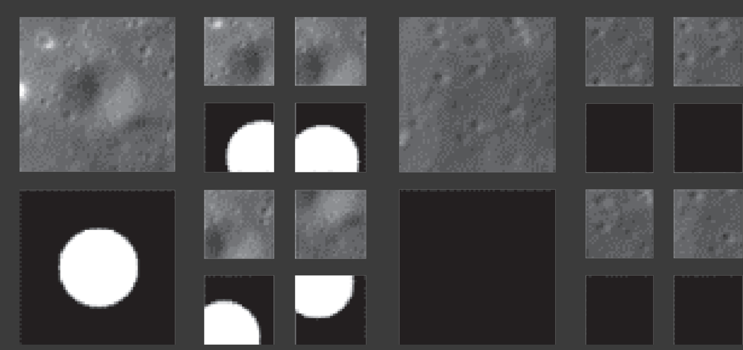
Fig 8 : creation of patches

The deep learning model employed here is a fully convolutional neural model. The reason is that we want to obtain from an input image an output of the same dimension whose pixels are the class of the input image. The model consists in a efficient series convolution, activation, pooling and deconvolion layers, ending with a sigmoid activation function to compute the probablity of the pixel to belong to a crater.

The implementation is made in Python language using the TensorFlow library developed by the GoogleBrain research project and the computation is performed on GPU using CUDNN developed by NVidia.

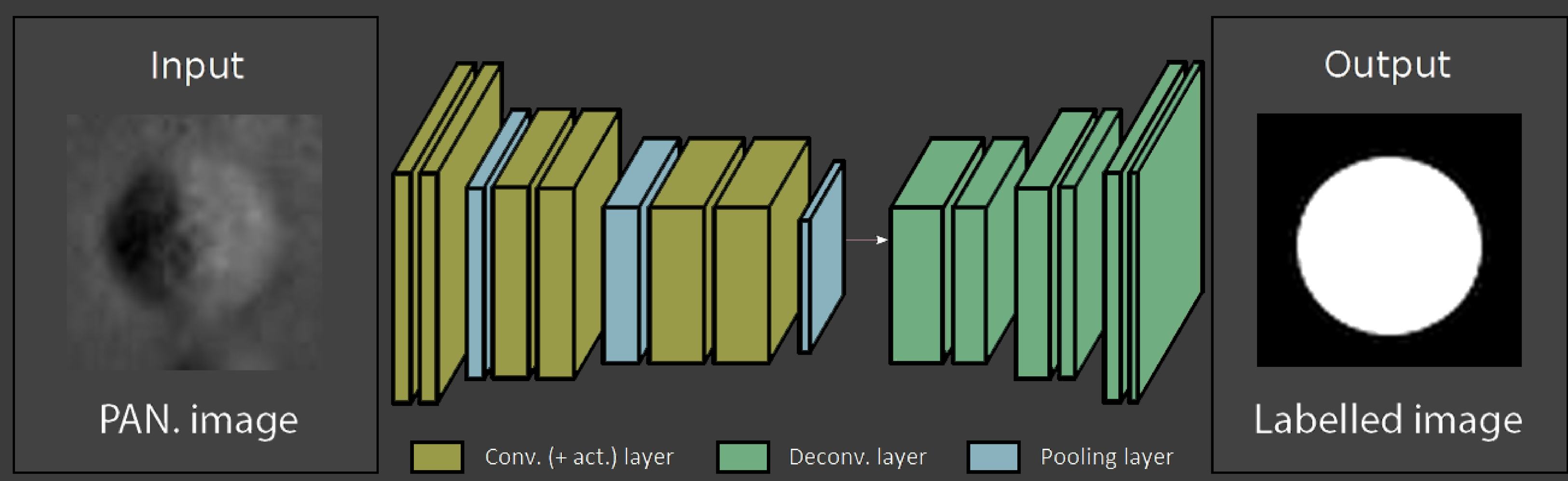

Fig. 9 : Our fully convolutional neural network

The parameters optimization is realized by gradient descent (adaptative moment estimator) with mini-batches. The goal is to minimize a specific loss function by iteratively updating the model's parameters according to a learning rate in a specific direction.

When the model is trained, we can analyze a lunar scene. Due to memory limitations, the computation was performed by creating a resolution pyramid and using overlapping tiles. Then the classified image is re-assembled and noise is reduced using opening and closing operations.

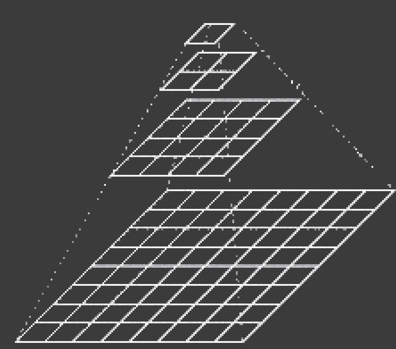
Fig. 10 : image pyramid

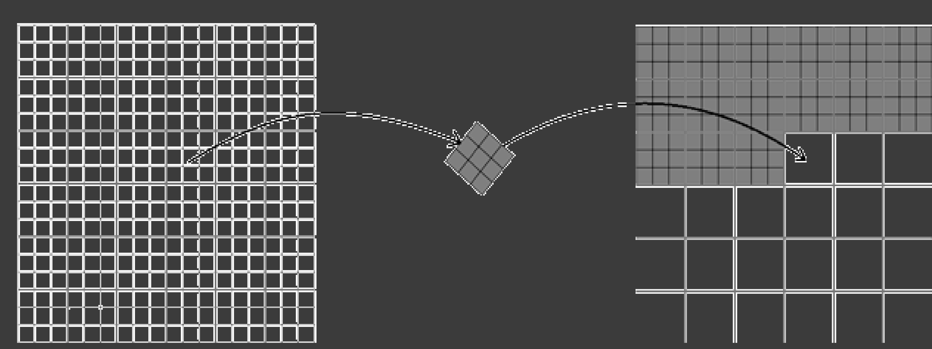
Fig. 11 : tile processing

4. Evaluation

The evaluation of the model relies on a pixel-based error matrix from which we extract classical machine learning metrics (Recall/sensitivity, specificity, precision, accuracy, F1-score, intersection-over-union IoU). The evaluation is performed using the independent validation set.

We also created ROC curves of the model. A ROC curve locate a model on a bidimensional space whose axes are two antagonist properties namely the true positive rate (sensitivity) and the false positive rate (1-specificity).

In the end, we obtain the following results :

- Accuracy ~ 90%
- F1-Score ~ 0.8
- IoU ~ 0.7

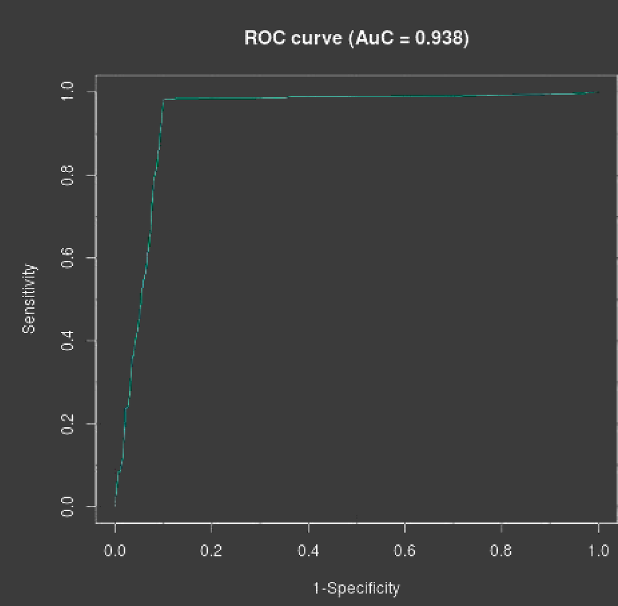

Fig. 12 : ROC curve

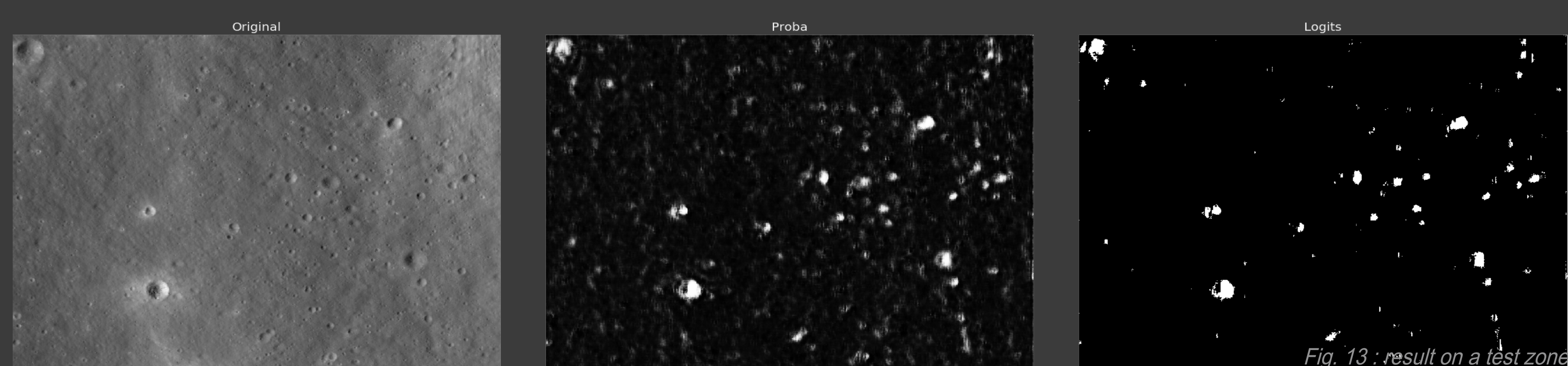

Fig. 13 : result on a test zone

5. Conclusions

In this master thesis, we propose a novel approach to detect lunar craters using new deep learning advances. The architecture of the model employed is a fully convolutional neural network that uses the freely available remotly sensed data from the «Lunar Reconnaissance Orbiter» space probe. We contributed somehow to the scientific community by providing a new dataset of craters and results using our novel methodology.

6. Ressources

NAC images : goo.gl/7j8CjM
Ortho-rectified images : goo.gl/qivZYt
My GitHub with all data and codes : github.com/QuentinGlaude/CraterNet
Cytomine Project : cytomine.be