Evaluating Machine Learning Performance: Development of Quantification Protocol to Evaluate AI Segmentation Algorithms

Centre de Biochimie Structurale (CBS) de Montpellier - 14 worked days

This manuscript was compiled on June 4, 2022

Corresponding authors

Nessim Louafi, Ali Kanso
M1 Quantitative BiologyUniversity of Montpellier
nessim.louafi@etu.umontpellier.fr
ali.kanso@etu.umontpellier.fr

Project Supervisors

Marcelo Nollmann, Jean-Bernard Fiche
Researchers at the CBS de Montpellier
marcelo.nollmann@cbs.cnrs.fr ,
jb.fiche@gmail.com

Abstract (989 characters):

Computer science has become a pivotal discipline when approaching biological problems. In microscopy for instance, the instruments and the growing scalability of the experiments have led the data analysis process to be computer dependent. Machine learning, as it has been proven recently to be a powerful tool for analysis, relies on the concept that a program can learn from data, and thus recognize specific patterns and make decisions to predict an output. When talking about image analysis, the fact that we must set rules for determining boundaries in data, makes image segmentation a major problem. Fortunately, such a problem is commonly encountered in computer sciences and many techniques have been developed. In this report we introduce a method that quantifies the accuracy of machine learning in performing image segmentation and we discuss different metrics that can be used to supervise such a process. We also propose a method to correct different networks for new training.

Introduction (4155 characters):

Machine Learning (ML), referring to a set of topics dealing with the creation and evaluation of learning algorithms, has shown remarkable utility in solving problems in various fields. Its methodology can be applied to large quantities of data and, therefore, can be particularly useful in biology. ML methods, and Neural Networks (NN) in particular [1], have been adapted in biology for a long time [2], but recently the field has seen a true explosion of interest [3]. For example, with the progress of microscopy techniques and the fast-growing amounts of acquired data (microscopists can easily collect terabytes of images in a few hours), it became crucial to automatize image analysis solutions in biological studies.

In this report we will develop one particular application of automatization: segmentation. We have been given data acquired in fluorescence microscopy on *Drosophila melanogaster* embryos with sequential imaging approach (Hi-M) that permits simultaneous detection of chromosome organization and transcription in single nuclei [4,5]. However, it is difficult to achieve powerful and accurate nucleus/cell segmentation. Microscopy images often express background with many artifacts, noises (e.g. blurred regions) that can be introduced during image acquisition, and causing potential poor contrast between the foreground and the background making the signal hard to detect. Also, the heterogeneity in intracellular intensity and the fact that they can be clustered and might partially overlap with one another makes the detection even harder.

In our case the question of detection of relevant objects (loci) in the images is something that is crucial to be automatized. Indeed, prior to analysis, the loci must be detected and defined. Many efforts have been made to tackle some or all of these challenges using machine learning. In this report we used a specific branch of machine learning called Deep learning (DL) that relies on the use of NNs to extract information. DL is particularly useful in image processing for its ability to extract information without depending on variations between images [6]. Convolutional NN (CNN) present a family of NN that we will use for the segmentation. CNN work by detecting specific and useful features on the input images. The network then creates (i.e “learns”) a set of filters specific to the data. Each filter is composed of a set of weights that are adjusted during the training steps (epochs). We worked with StarDist [7] as a tool for the detection and segmentation of cell nuclei. It works with 2D data and 3D volumes. An illustration on the principle and the output of a StarDist NN is shown in figure1. We will refer to the objects outputted by the network as labels. These techniques although powerful often require high computing power and memory. Hence, during our internship we have been introduce to working on Linux and through servers to lunch powerful computation on dedicated computers.

To evaluate training in segmentation a testing dataset is used consisting labeled images (ground truths). From these images one can define a loss function to track the training process. Common statistics also involve calculating the true and false positive/negative detection (contingency table) as shown in annex 2. Although quantitative these methods require prior knowledge on the image as they need to compare expected versus predicted values. Moreover, no information is given on what the network missed and why the accuracy is not 100 %. In this report we were interested in developing a way to evaluate the performance, this method will allow us to find when the network fails, and in addition it can be applied to new images that the network has never seen before.

In this report we will show how we developed a method to evaluate both qualitatively and quantitatively the performance of a network and show how this information can be used to compare different networks. Moreover, we showed a way to investigate a network tendency to "miss" bright fluorescent events. Finally, we show how this investigation can allow to retrain a network in the hope of improving its performance.

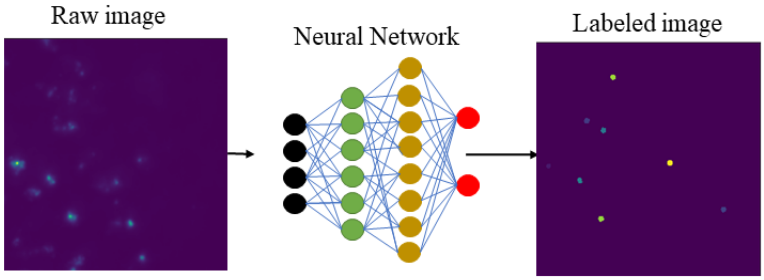


Figure 1: Principle of a NN

2.Methods (2447 characters…):

2.1. Neural network:

The networks were based on Stardist[7]. Stardist networks were trained to detect fluorescent spots on a 3D image. The network is using as input a 3D fluorescent image and returns as output a 3D stack of the same dimension were all the detected objects have been instantiated (pixel value > 0). We worked with three different networks based on different trainings; The *PSF-simulated network* trained using simulated images (PSF-shaped objects), the *data network* trained with experimental data labeled using conventional imaging-based algorithms, and the *retrained network* that we will develop later in this report.

2.2. Qualitative network performance:

To evaluate qualitatively the network performance, we used scikit image [8] to retrieve the coordinates of the different labeled objects. We then transposed those coordinates into the raw image and extracted the region of interest. This process was done iteratively on all the labels and a gallery was then created (figure 2). Furthermore, the gallery was classified by increasing intensity (from left to right) by comparing the maximum fluorescence of every snippet of the raw image.

2.3. Quantitative exploration of the results:

Using the labeled image, we gathered the statistics of every labels using the package scikit image and calculated the following properties: volume, maximum intensity, the ID of the detected object.

2.4. Intensity sensitivity analysis:

To assess the network, a comparison analysis was developed using the algorithm ASTROPY as a reference [9]. It uses intensity thresholding to detect objects. To compare we calculated an accuracy metric described as the ratio between the number of objects detected by both methods divided by the number of objects detected by the ASTROPY algorithm only.

2.5. Correction of a network:

To correct the network, we first applied ASTROPY detection on the raw images to detect the brightest loci. We compared the positions of the detections with the labelled image obtained from the network. For every ASTROPY detection, the coordinates were reported on the labeled image and verified if the pixel value was superior to 0 meaning that a locus was segmented. In case of missing objects, we modified the labelled image by adding an artificial sphere and used a subset of this image for the training. This process was done on n=4 (2048,2048,70) images and the corrected images were split between a training and a testing dataset.

3. Results (3000 characters):

3.1 Qualitative approach to network performance:

Our first approach to the question of evaluating segmentation performance was qualitative. The network outputs stacks of 2048x2048x70 which are difficult to visually inspect. To visualize the labels, we thus relied on the gallery of images described in the material section 2. This gallery represents every object the network considered as "loci". Although not quantitative, this step allowed us to have a first idea on the kind of objects outputted. An example of output is present in figure 2. We see that the network segmented a majority of bright and round shaped objects (orange arrows). Surprisingly, some objects were close to background level (black arrows). However, we see some non-PSF objects which indicates a poor network performance (indicated by white arrows).

3.2Quantitative approach to network performance:

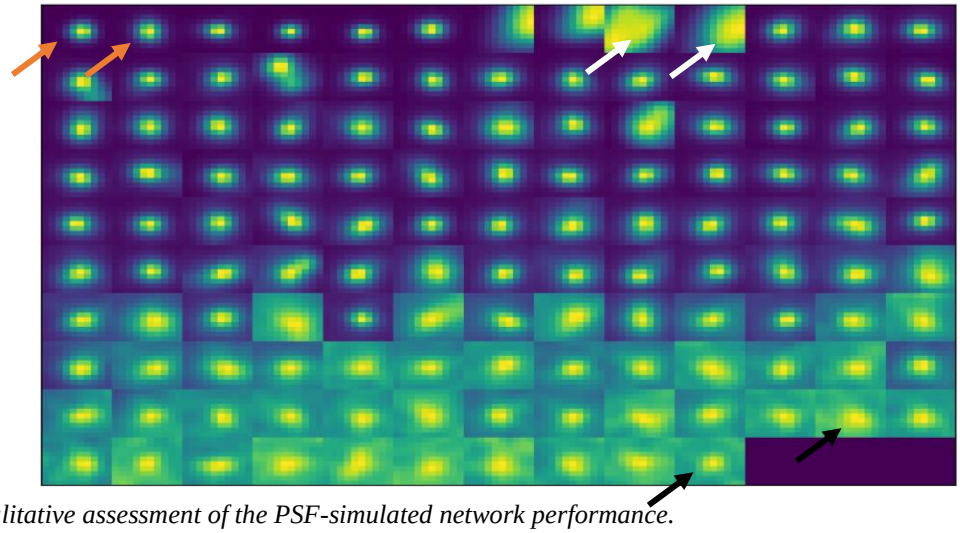


Figure 2: Qualitative assessment of the PSF-simulated network performance.
Coordinates of segmented objects reported on the raw image and ranked by increasing maximum intensity from left to right and top to bottom. Orange arrows represent bright round PSFs, white arrows show noisy objects and black arrows show objects close to background level.

Having a qualitative approach allowed us to have a first insight to a network performance. However, we wanted to have a quantitative analysis to be able to compare performance between networks and also between trainings. As described in the method section 3 we used the intensity and volume of labels metrics. Figure 2 shows the comparison

between two different networks. Looking at panel C we see that the *PSF-simulated network* detected object with lower intensity than the *data network*

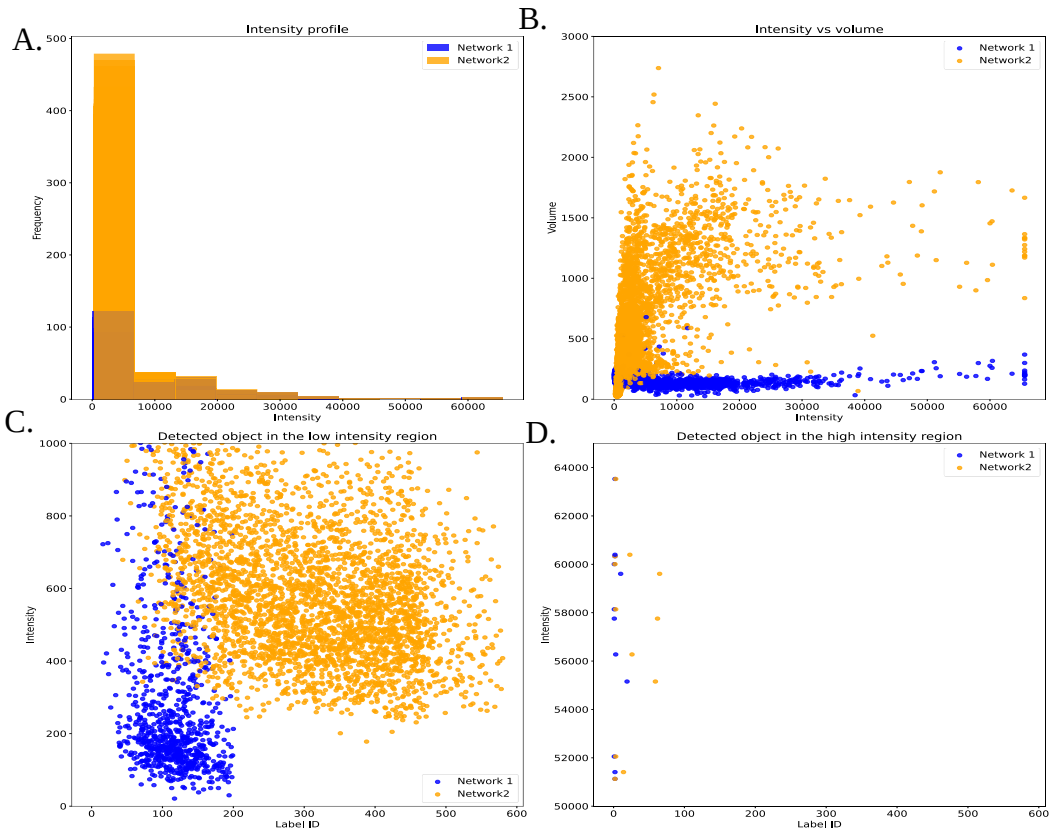


Figure 3: Quantitative comparison between two network shows clear differences in network performance
Network 1 PSF-simulated network 2: Data network A.) Intensity histogram of all the segmented objects B.) Volume versus intensity for all the objects segmented for the two networks C.) Intensity versus label ID for the two networks. Every labeled object is attributed a unique label by the network D.) Intensity profile in the high intensity region for both networks. The high intensity has been set using the camera properties and the maximum pixel value.

3.3 Further study on intensity sensitivity:

As shown in figure 3.D the high intensity region shows a small number of labels. To study this observation, we looked at intensity-based segmentation algorithm for comparison. The Astropy algorithm was originally developed for astronomers to segment stars [9]. To qualitatively evaluate such a comparison, we relied on the accuracy metric described in the method section 4. This metric will yield higher value for images where all the high intensity objects were detected by the NN. As shown in figure 4 not all the bright objects were segmented by the network (between 70 and 80% accuracy, annex 4). This conclusion confirmed prior observations that bright objects tend to be "missed". Considering the importance of bright fluorescent events, we focused on finding a way to correct for missed events. We developed an algorithm to modify images with missing event that were used to retrain a new network.

3.4 Correction of a network:

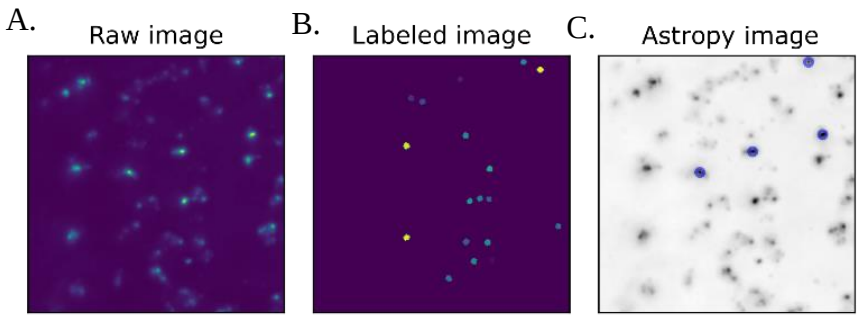


Figure 3: The network shows difficulties to segment high intensity objects;
A.) Raw image acquired by the microscope. B.) Output of the network colored by labels (one label = one color). C.) Output of Astropy algorithm with in blue the detected high -intensity objects

We ran our algorithm on n=4 images to build a new training set. The network was given 500 epochs of 100 steps (based on the parameters of the previous networks). Statistics of the training can be found in annex 2. To evaluate the effect of our correction we performed a segmentation with the retrained network on a never-seen image that yielded the results shown in figure 5. We compared the results with the *PSF-simulated network*. We observe a shift in the volume distribution in the retrained network (panel A). Surprisingly, the retrained network didn't show an increase in high intensity detection rather a global reduction of number of detected objects (panel B).

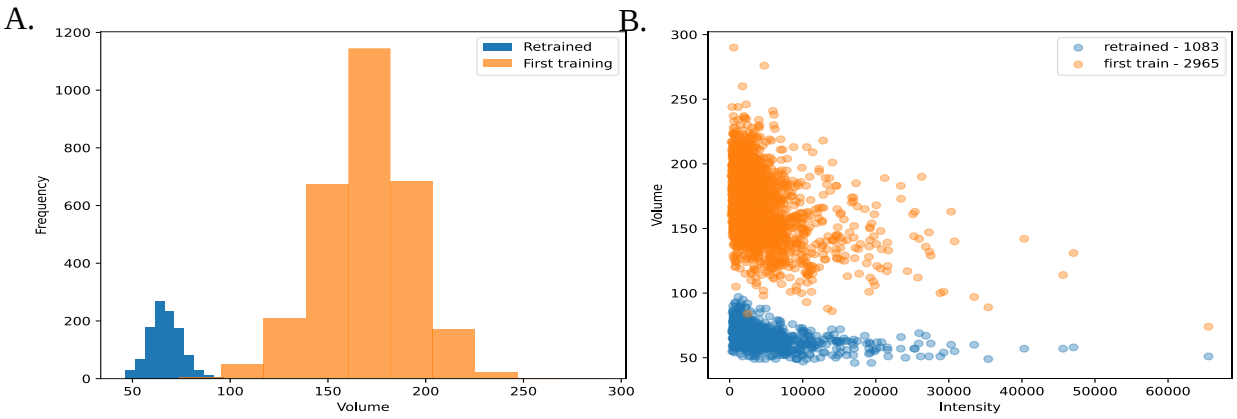


Figure 2: Quantitative assessment of the retraining process shows no improvement in the object detection
In blue: the retrained network. In orange: the simulated network. A.) Intensity distribution of all the segmented objects by the networks. B.) Volume versus intensity profile the number next to the legend is the number of segmented objects.

4. Conclusion (1453 characters):

In this project, we have developed qualitative and quantitative methods to assess the performance of NN on fluorescent microscopy imaging data. We applied these methods to characterize the performance of 2 existing differently trained networks. The '*data-trained network'* detected large objects with a clear cutoff at low intensities. The *'PSF-simulation based network'* detected mostly PSF-shaped objects, but in some instances, it failed to detect spots that have high intensity. Furthermore, we created a python pipeline to automatically correct images where high-intensity spots were not detected (when compared to ASTROPY) and used the new images as a ground truth to widen the training data set and trained a new network. The new training set was made of a mix of simulated PSF and corrected fluorescence data. This resulted in poor performance, suggesting that more data in the training network do not guarantee better segmentation results.

Indeed, the retrained network segmented less objects with smaller volumes. The reason may be that the network was unable to generalize enough and incorporate the new information carried by the corrected images. However, further work is necessary for more thorough interpretation of our results. Further analysis could include the training of a new network using only the corrected images, or lessen the proportion of existing data in training set to see how this affects the performance of the segmentation.

5. Data availability:

All the notebooks used and the plots showed throughout this report can be found on Github as well as our respective Lab-notebooks. A summary of the content of the different notebooks can be found in annex 3.

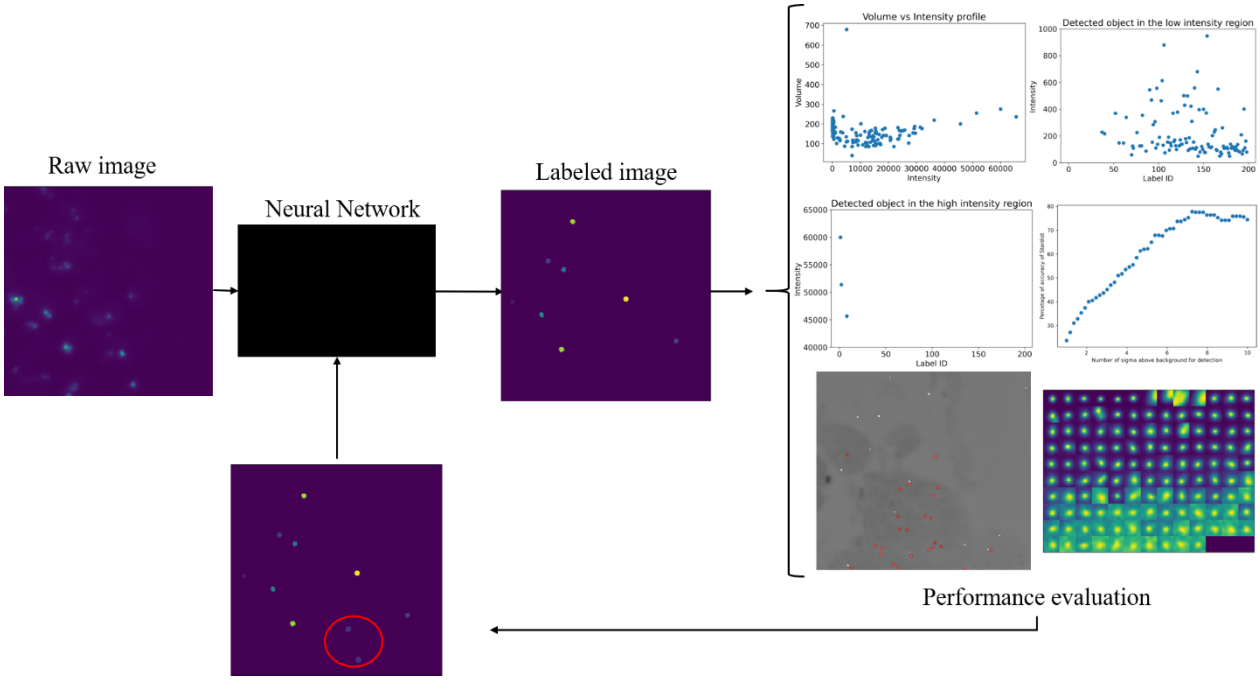
6. Acknowledgments

We would like to thank our supervisors M.Nollmann and J.B Fiche and the other lab members for their supervision, guidance and review of this report.

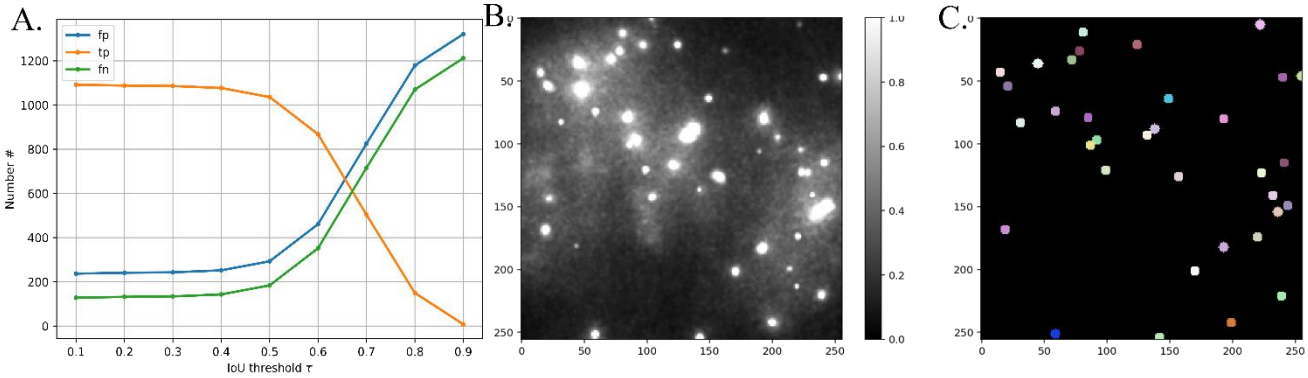
7. References:

[1] R. C. Eberhart, *Neural network PC tools: a practical guide*. Academic Press, 2014.
[2] G. D. Stormo, T. D. Schneider, L. Gold, et A. Ehrenfeucht, « Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli* », *Nucl Acids Res*, vol. 10, nº 9, p. 2997-3011, 1982, doi: 10.1093/nar/10.9.2997.
[3] T. Ching *et al.*, « Opportunities and obstacles for deep learning in biology and medicine », *Journal of The Royal Society Interface*, vol. 15, nº 141, p. 20170387, avr. 2018, doi: 10.1098/rsif.2017.0387.
[4] A. M. Cardozo Gizzi *et al.*, « Microscopy-Based Chromosome Conformation Capture Enables Simultaneous Visualization of Genome Organization and Transcription in Intact Organisms », *Molecular Cell*, vol. 74, nº 1, p. 212-222.e5, avr. 2019, doi: 10.1016/j.molcel.2019.01.011.
[5] S. M. Espinola *et al.*, « Cis-regulatory chromatin loops arise before TADs and gene activation, and are independent of cell fate during early Drosophila development », *Nat Genet*, vol. 53, nº 4, Art. nº 4, avr. 2021, doi: 10.1038/s41588-021-00816-z.
[6] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, et D. Van Valen, « Deep learning for cellular image analysis », *Nat Methods*, vol. 16, nº 12, p. 1233-1246, déc. 2019, doi: 10.1038/s41592-019-0403-1.
[7] U. Schmidt, M. Weigert, C. Broaddus, et G. Myers, « Cell Detection with Star-Convex Polygons », in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, vol. 11071, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, et G. Fichtinger, Éd. Cham: Springer International Publishing, 2018, p. 265-273. doi: 10.1007/978-3-030-00934-2_30.
[8] S. van der Walt *et al.*, « scikit-image: image processing in Python », *PeerJ*, vol. 2, p. e453, juin 2014, doi: 10.7717/peerj.453.
[9] L. Bradley *et al.*, *astropy/photutils:* Zenodo, 2022. doi: 10.5281/zenodo.6385735.

Annexes:



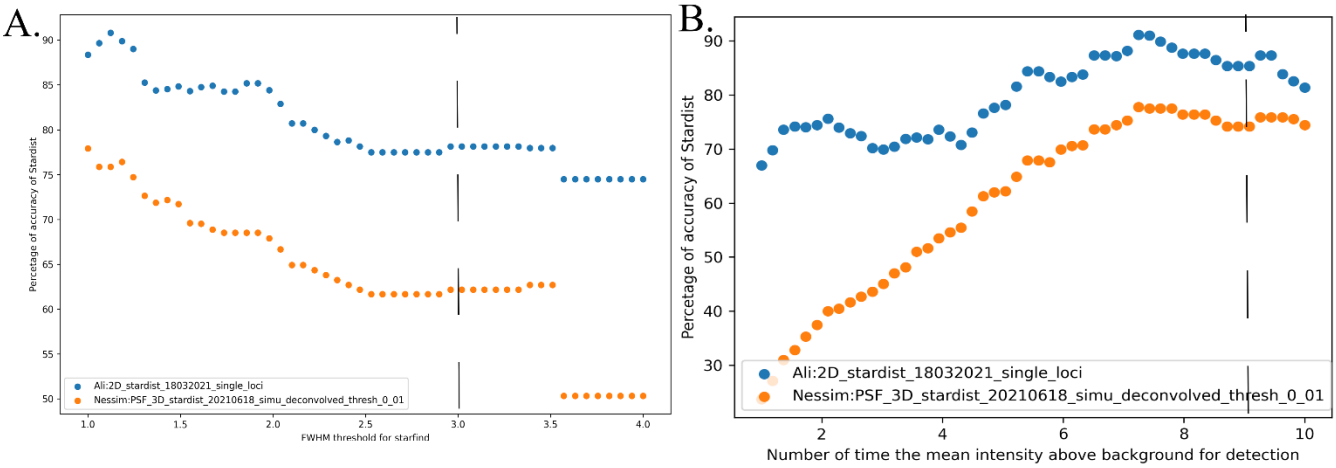
Annex 1: General workflow of the lab 1



Annex 2: Training statistics for the retraining
The IoU threshold (intersection over union) is used to allow an object to be considered true positive or false positive. It represents the ratio between object present in both the ground truth and the prediction over the total number of objects. If that value is over the set threshold it is considered as true positive. Evolution of the different metrics of the contingency table versus the IoU threshold. B.) Example of a training image given to the retrained network. C.) Labels predictions of the training

Name of the jupyter notebook	Content
Comparing_network_performance.ipynb	Final metrics computations
krakatoa_run_stardist_exploring metrics.ipynb	Exploration on different metrics to compare networks
krakatoa_run_stardist_exploring_results.ipynb	Gallery creation
Updating_stardist_with_astropy_exploratory.ipynb	Test the intensity sensitivity and the correction process
Updating_stardist_with_astropy_final_pipeline.ipynb	Correction pipeline for the network and creation of the training dataset

Annex 3: Table of the different notebooks produced



Annex 4: Accuracy measurement for varying thresholding parameters
In blue: data network. In orange: PSF-simulated network. A.) Accuracy measurement for one image using both networks (PSF-trained and data-trained) with increasing full width half maximum parameters (width of the PSF) B.) Accuracy measurement for one image using both networks (PSF-trained and data-trained) with increasing intensity cutoff. In this case the threshold was set as a x times the mean intensity of the image.