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

A.Kanso, N.Louafi, M.Nollman**
*Corresponding authors: kanso.ali@outlook.fr, nessim.louafi@protonmail.com
** Project Supervisor: marcelo.nollmann@cbs.cnrs.fr

Abstract (986 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 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 process. We also propose a method to correct different network for new training.

Introduction (4806 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 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 minutes), it became crucial to automatize image analysis solutions in biological studies. For example, the question of detection of relevant objects in an image is something that is crucial to be automatized. Indeed, prior to analysis, structures of interest must be detected and defined for further quantification. This is achieved through the process of segmentation, consisting on partitioning an image into homogeneous regions and replacing intensity values by region labels.

In a classical fluorescence experiment, segmentation is the process of delimiting the marked region from the background before entering in fluorescence quantification *per se*. When working with prokaryotic organisms or in a cellular scale this process means having to define thousands of objects one by one. Such process done manually would require an unnecessary amount of time. For that reason, trainable machine learning methods turned up to be powerful tools to include that knowledge in the segmentation process and improve the accuracy of the labeled regions.

Most ML techniques require high computing power to perform operations. During this internship we have been introduced to working through a server and via the Linux operating system to perform all the computations. We have been given data acquired in fluorescence microscopy on drosophila embryo 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.

Many efforts have been made to tackle some or all of these challenges. Convolutional neural networks (CNN) can present a solution if the networks are well trained, and the nuclei are segmented providing ground truth image volumes[6,7]. We got introduced to StarDist [8] as a tool for the detection and segmentation of cell nuclei. It works with 2D data and 3D volumes. Convolutional neural network work by assigning weight or coefficient to parameters used for the classification. Theses weights however need to be adjusted to fit the purpose wanted: in our case segment nuclei. The process of obtaining theses weights is called training and consist of *n* steps (epoch) i.e times in which the weights are adjusted [7]. An illustration on the principle and the output of a neural network is showed in figure1.

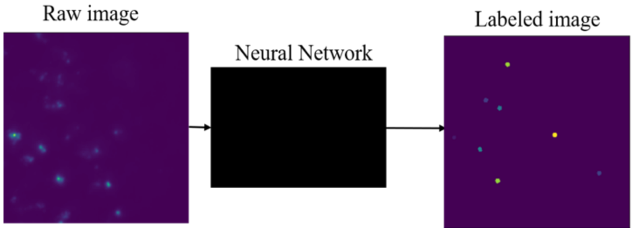


Figure 1: Principle of a neural network

To evaluate such process a testing dataset is used consisting of images with known object location (ground truths). From these images one can define a loss function to track the training process. Common statistics also involve calculating the contingency table of a known image as shown in annex 1. Although accurate these methods of evaluation of a network performance are only relevant for the network part. Moreover, such metrics all requires a prior knowledge of the image i.e. a ground truth. In this report we were interested in developing a way to evaluate the performance without this prior knowledge. Thus, this method would apply for images not used in the training which are of biological relevance. The idea was to verify the efficiency and accuracy of two neural networks in detecting objects. Intuitively, we first looked qualitatively at the output of the network and created a way to form a gallery of detected objects. Then we developed metrics to evaluate the segmented objects and showed that we could compare and find differences between networks. Furthermore, after previous observation we investigated the sensibility of the various networks to high intensity objects and showed qualitatively this sensitivity. Finally, we devised a method to verify if bright objects were missed and created corrected ground truth images used for training of a new neural network.

Methods (2451 characters):

Neural network:
The networks were based on Stardist [8]. This algorithm is based on the prediction of star-convex polygons to segment images. More specifically we worked with three different networks based on different trainings; The *simulated network* trained using simulated images (gaussian-shape objects), the *data network* trained using experimental data labeled by hand and the *retrained network* is based on the simulated network but with an addition of images that will be develop in the following section.

Qualitative network performance:
To evaluate qualitatively the network performance, we used the package scikit image [9] to retrieve the coordinates of the different labeled objects. We then transposed those coordinates into the raw image and extracted a 10x10 region of the image. This process was done iteratively on all the segmented object and a gallery was then created. Furthermore, the gallery was classified by increasing intensity (from left to right) by comparing the maximum fluorescence of every snippet of the raw image. To make sure this measure was accurate for all the objects and correcting for out of focus objects the extraction was done at the z-center position of the labeled object.

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

Intensity sensitivity analysis:
To assess the network tendency to fail the detection of bright objects a comparison analysis was developed using the algorithm Astropy as a standard [10]. This algorithm is based on 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.

Correction of a network:

Segmented images were submitted to Astropy analysis. For every Astropy detection the coordinate were reported on the segmented image if at this location (x,y,z) we verify if the pixel value was superior to 0 in which case meant a segmented object. In the other situation we classified the objects as missed added a (5,5,7) sphere and cropped (256,256,70) images in the labeled image and the raw 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.

Results (3066 characters):

Qualitative approach to network performance:

Our first approach to the question of evaluating segmentation performance was to qualitatively assess the segmentation. The network outputs stacks of 2048x2048x70 which are complicated to visually inspect. To visualize the objects, we thus relied on the gallery of images described in the material section. This gallery represents every object the network considered as "nuclei". 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 1. We see that the network segmented a majority of bright and round shaped objects which was expected. Moreover, some objects were close to background level. However, we see that some objects are noise which indicates a poor network performance.

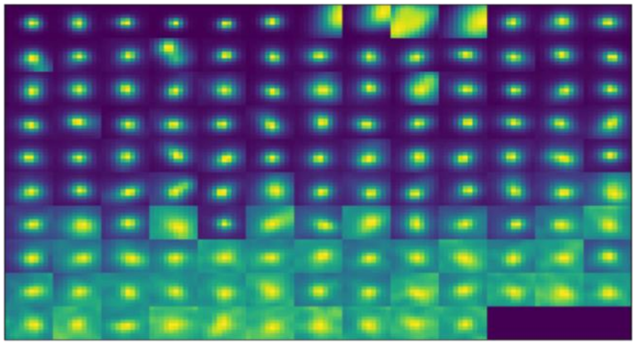


Figure 3: Qualitative assessment of a network performance. Coordinates of segmented objects reported on the raw image and ranked by increasing maximum intensity from left to right and

Quantitative approach to network performance:

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 network and also between trainings. As described in the method we used various properties of the objects to compare networks. Figure 3 shows the comparison between two different networks. We see that there is a clear difference in network performances as one network tends to segment bigger objects and with higher intensity.

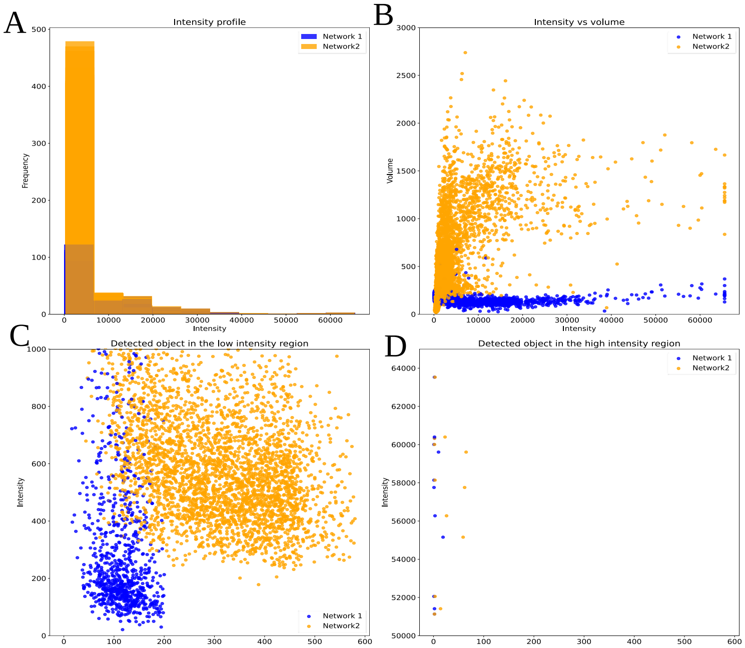


Figure 2:Quantitative comparison between two network shows clear differences in network performance

Network used: Simulated and image-fed 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.

Further study on intensity sensitivity:

To study the dependency to intensity of the segmentation we decided to compare our neural network approach to an intensity-based segmentation. The Astropy algorithm [10] was originally developed to segment images of the sky to find bright objects: stars. To qualitatively evaluate such comparison, we relied on the accuracy metric described in the method section. This metric will yield higher value for images where all the high intensity objects were detected by the neural network. As shown in figure 4 not all the bright objects were segmented by the network. This conclusion confirmed earlier visual observation 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. To do so, we decided to play on the training of the network. The neural network is trained using a set of images and tries to generalize to be able to be used on a variety of images.

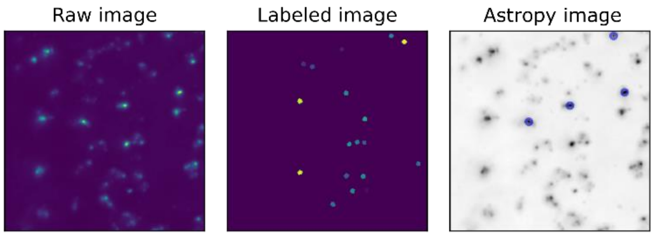


Figure 4: The network shows difficulties to segment high intensity objects;
From left to right: the raw image, the output of the network and the output of the Astropy algorithm

The network was given 500 epochs of 100 steps. Statistics of the training can be found in annex 2. Classically one can look at the contingency matrix but that is assuming a ground truth is known. To evaluate the quality of the training and thus the performance of the network we performed a segmentation with the new network on a never-seen image that yielded the results shown in figure 5.

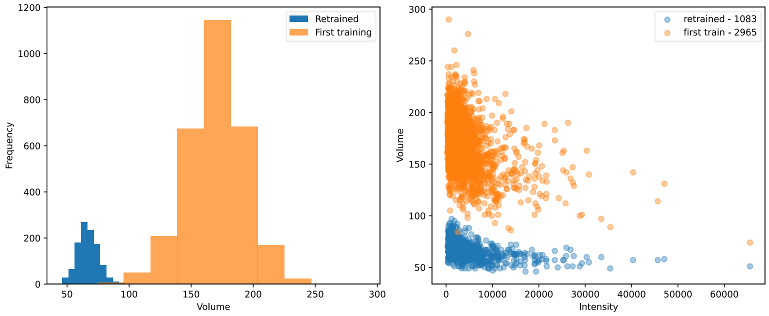


Figure 5: Quantitative assessment of the retraining process shows no improvement in the object detection
In blue: the retained network. In orange: the simulated network. A.) Intensity distribution of all the segmented objects by the networks. B.) Volume versus intensity profile the number next to the legend is the number of segmented objects.

We compared the results with the first network we worked with, described above. We observe that as the object detected seem coherent (panel A), there is a loss of the low intensity objects. Moreover, the amount of detection has decreased as well as the volume distribution. These results were not what we expected.

Conclusion (1285 characters):

Throughout this report, we saw that two differently trained neural networks performed differently in detecting and segmenting nuclei on fluorescent microscopy images. We have been able to determine the accuracy and quantify the differences between the two. The *data network* had tendency to segment more objects with bigger volumes, higher intensity, and inhomogeneous shapes, when on the other hand, the *simulated network* showed better segmentation of nuclei with a round homogeneous shape. We noticed that spots with high intensity seemed to be missed in some cases. For that and other corrections, we created a python pipeline to automatically correct images where intensity spots were not detected (when compared to Astropy) and use them as a ground truth to widen the previously prepared training data set and train a new network. The segmentation results of the new training did not match our expectations, suggesting that more data in the training network do not guarantee better segmentation results.

However, further work is necessary for more thorough interpretation of our results. We can try training new network using only the corrected images, or lessen the proportion of existing data in training set to see how this affects the classification and therefore the segmentation.

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, n° 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.

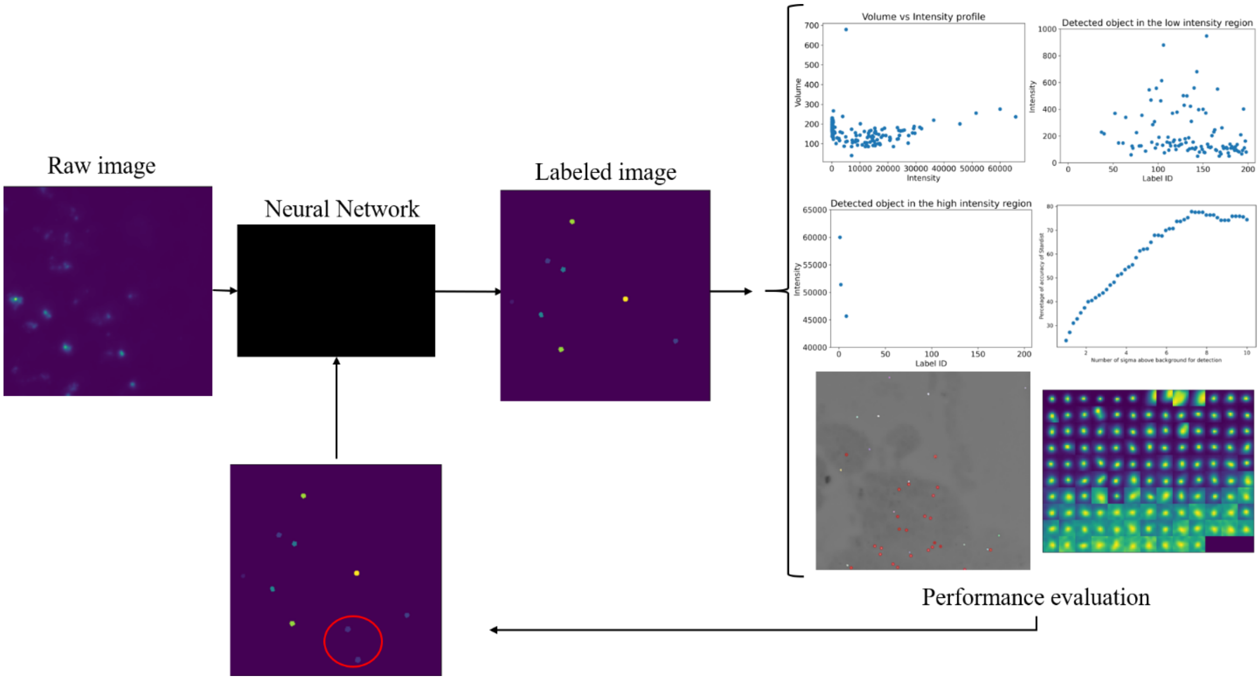
[7] J. G. Greener, S. M. Kandathil, L. Moffat, et D. T. Jones, « A guide to machine learning for biologists », *Nat Rev Mol Cell Biol*, vol. 23, nº 1, p. 40-55, janv. 2022, doi: 10.1038/s41580-021-00407-0.

[8] 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.

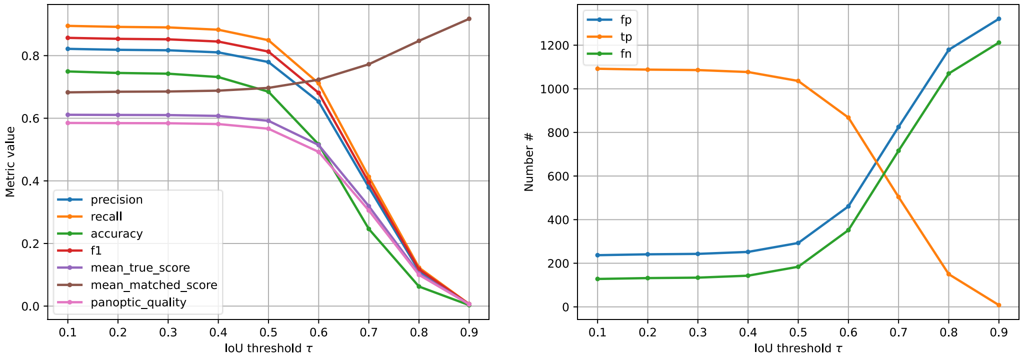
[9] S. van der Walt *et al.*, « scikit-image: image processing in Python », *PeerJ*, vol. 2, p. e453, juin 2014, doi: 10.7717/peerj.453.

[10] L. Bradley *et al.*, *astropy/photutils:* Zenodo, 2022. doi: 10.5281/zenodo.6385735.

Annexes:


Raw image
Neural Network
Labeled image
Volume vs Intensity profile
Volume
Intensity
Detected object in the low intensity region
Intensity
Label ID
Detected object in the high intensity region
Intensity
Label ID
Percetage of accuracy of Stardist
Number of sigma above background for detection
Performance evaluation

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. A.) General metrics of training versus the IoU threshold (explain) B.) Evolution of the different metrics of the contengency table versus the IoU threshold.

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