



INSTITUT  
POLYTECHNIQUE  
DE PARIS



---

# Documentation

## Fourier Ptychography Microscopy for malaria detection

---

Realized by :

**Slimane Baamara and Weiqiang Liu**

Under the supervision of:

**Prof. Patrick Horain** Telecom SudParis

October 1, 2021

## Contents

<b>1</b>	<b>Image preprocessing plugin</b>	<b>2</b>
<b>2</b>	<b>Image segmentation</b>	<b>3</b>
2.1	Image segmentation plugin . . . . .	3
2.2	Image segmentation results . . . . .	5
<b>3</b>	<b>Cells extraction</b>	<b>6</b>
<b>4</b>	<b>Image reconstruction using (FPM)</b>	<b>7</b>
<b>5</b>	<b>Image classification plugin</b>	<b>8</b>
<b>6</b>	<b>Image saving and loading plugin</b>	<b>10</b>
<b>7</b>	<b>Contour assist plugin</b>	<b>11</b>

## Abstract

CellAnnotator is a plugin for Napari viewer. It will help users to annotate cells in biomedical images, taking advantage of napari's capability to view large n-dim images interactively and python's availability to use powerful machine learning algorithms from open-science library, such as sklearn, opencv, scikit-image, or even deep learning libraries such as TensorFlow and PyTorch and keras.

The main feature of CellAnnotator is the interactive learning based on ML/DL algorithms.

ML and DL show remarkable accuracy for segmentation task in papers [1, 2, 3, 4]. However, these models would not work very well on your own data, simply because they are not optimized for your data.

As a result, their predictions are not going to be perfect and almost always need human efforts in the end.

CellAnnotator will provide ML algorithms to help users to correct annotation. Then, it will make use of the corrected annotations to fine-tune the model in order to be optimized for your own data.

We came up with a few scenarios for different use cases, inspired from existing tools such as Ilastik and AnnotatorJ, and each time we imagine what features the user might need, then we try to outperform the existing tools.

This user interface consists of several plugins which are described below: Image preprocessing 1, Image segmentation 2, Image reconstruction 4, cell image extraction 3, cell classification ??, Image saving & loading 6 and Contour assist 7.

## 1 Image preprocessing plugin

The image preprocessing allows the user to realise simple operations on the input image and to prepare the data for the segmentation part.

This plugin allows for the following operations:

- **Choose a channel:** The user can choose a channel in case of multi channel images, or choose "gray" to get the grayscale of the input image (RGB images), or "None" to pick the raw image.
- **Image smoothing:** The user can perform a gaussian filtering of the image by selecting first the image layer from the list of layers, then selecting a standard deviation value using the SpinBox, then press the button "OK".
- **Difference of Gaussians:** The user can perform a pass band filtering of the image by selecting first the image layer from the list of layers, then selecting a minimum and maximum values of the standard deviation using the 1st and 2nd SpinBoxes, then press the button "OK".
- **Invert the image:** The user can invert the image in case of white background and black foreground by selecting first the image layer from the list of layers, then pressing the "Invert" button.
- **Crop the image:** The user can draw a/many shape(s) using the shape layer then extract the pixels from those shapes(patches) as a new image layer (this could be useful for training a model or making fast processing of the images when dealing with big images), by adding first a shape layer and drawing the shapes, then selecting the source image layer from which the patches will be extracted, then pressing the "Crop" button, this results in rectangular cropped images.

Figure 1, illustrates the different buttons and options that Image processing plugin contains. While, the Figure 2 illustrates the results obtained when applying the crop and invert functions, where, a shape layer is used to draw a patch on the image region to be cropped. Also, since the image has a black foreground, it should therefore be inverted. Finally, we produce an image which will be used as input for the segmentation.

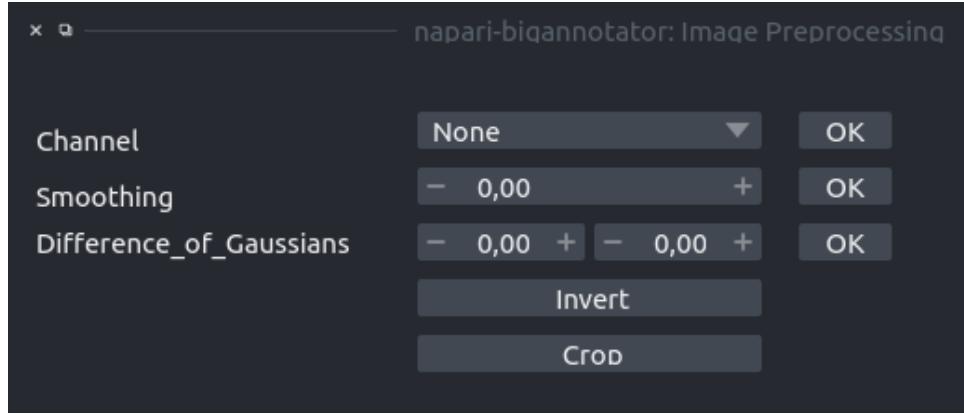


Figure 1: Image preprocessing plugin

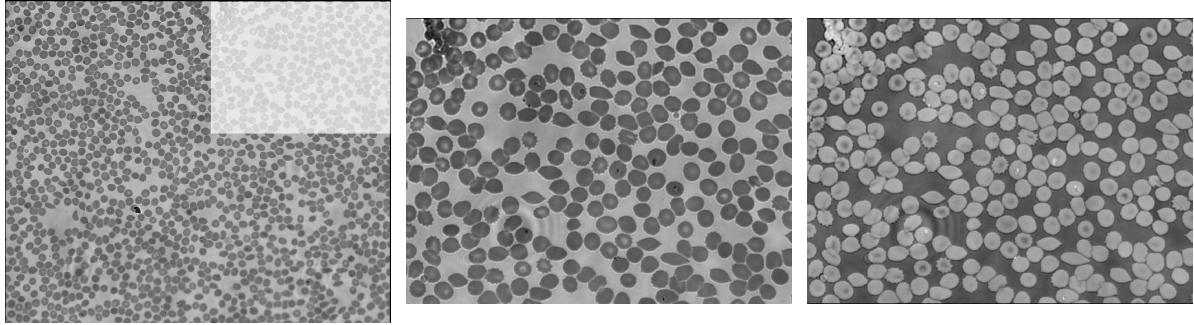


Figure 2: **Image preprocessing results:**

**Left:** original image+drawn shape, **Center:** cropped image, **Right:** inverted image

## 2 Image segmentation

### 2.1 Image segmentation plugin

The image segmentation plugin allows the user to realise classic and advanced image segmentation on the preprocessed image and to output a mask and label images.

This plugin allows for the following operations:

- **Choice of an image and shape layers:** By clicking on the reload button the list of image/shape layers will be updated. The shape layer is chosen only if the user wanted to perform segmentation only on specific regions of the image, an alternative of this is to crop the regions of interest on the preprocessing step, then perform segmentation on the cropped images.
- **Choice of the segmentation method:** After selecting the image and shape layers (if needed) the user will have to choose between using classic or advanced segmentation algorithms.
- **Classic methods:** If the classic method is chosen, then one of the 3 thresholding based algorithms are suggested.
  - **Manual thresholding:** By selecting this option the user will have to choose a threshold then to press "Run" button to perform segmentation, as illustrated in figure3 top left.
  - **Otsu thresholding:** By selecting this option the user will have to press "Run" button to perform segmentation, then the Otsu threshold is displayed in LineEdit box, as illustrated in figure3 top center.
  - **Local thresholding:** By selecting this option the user will have to choose a bloc size (the characteristic size surrounding each pixel) then to press "Run" button to perform segmentation, as illustrated in figure3 top right.
  - **Hough transform:** By selecting this option the user will have to set the parameters by pressing the "set parameters" button, then a new window will appear. The parameter's window contain a list of parameters:

- \* **Verbose:** Whether or not to plot all figures while processing.
- \* **Cell mean:** To choose a cells diameter, by default its set to 60.
- \* **Output directory:** To choose a directory where the produced mask and labels will be saved as 'tiff' images.

This is illustrated in figure3 bottom left and right.

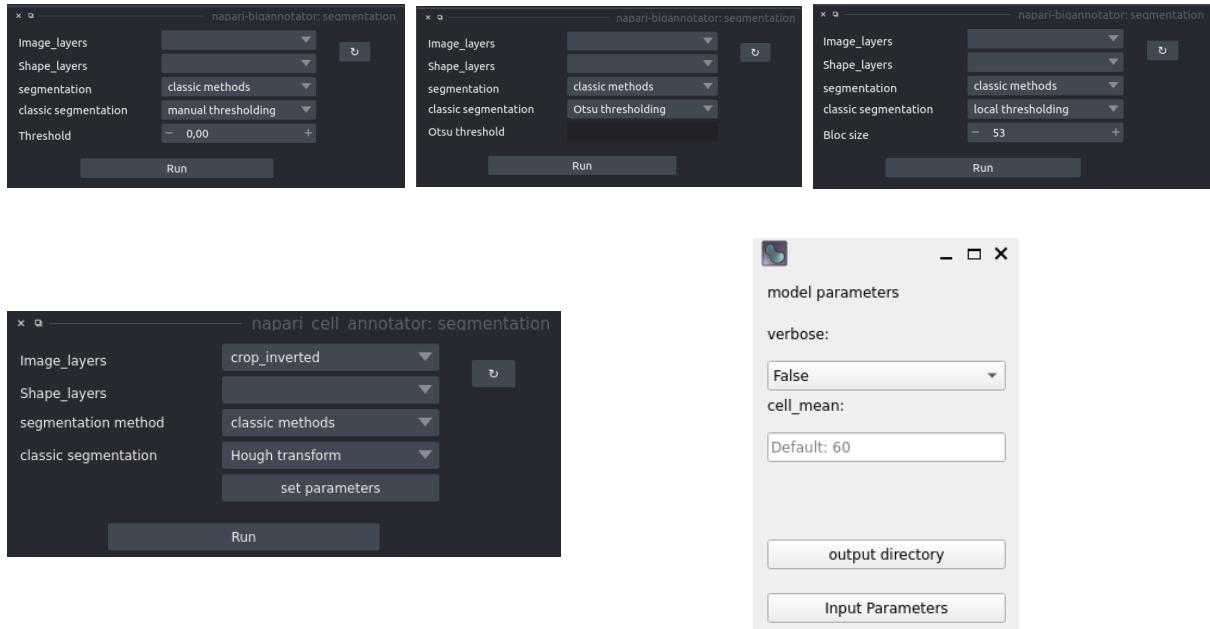


Figure 3: **Image segmentation plugin (classic methods):**

**Top left:** manual thresholding, **Top center:** Otsu thresholding, **Top right:** local thresholding,  
**Bottom left:** Hough transform, **Bottom right:** Hough parameters

- **Advanced methods:** If the user chooses to use advanced methods, then one of the 3 advanced algorithms are suggested.
  - **StarDist:** By selecting this option the user will have to choose one of the image types from the list of image types, then to press "Run" button to perform segmentation using a pretrained StarDist, as illustrated in figure4 left.
  - **CellPose:** By selecting this option the user will have to set the parameters by pressing the "set parameters" button, then a new window will appear. The parameter's window contain two lists of parameters:
    - **Model parameters:** This parameters are used to import the model.  
 The user will have:
      - Whether or not to save model to GPU, will check if GPU available (default: False).
      - To choose the model type: 'cyto'=cytoplasm model; 'nuclei'=nucleus model; if None, pretrained\_model used.
      - To set a mean 'diameter' (default: 27).
      - To set the path to pretrained cellpose model(s).
    - **Model evaluation parameters:** This parameters are used for the model evaluation.  
 The user will have:
      - To choose a list of channels, either of length 2 or of length number of images by 2. First element of list is the channel to segment (0=grayscale, 1=red, 2=green, 3=blue). Second element of list is the optional nuclear channel (0=none, 1=red, 2=green, 3=blue), (default is [0,0]).
      - To choose a flow error threshold (all cells with errors below threshold are kept), (default: 0.4).
      - To choose cell probability threshold (all pixels with prob above threshold kept for masks), (default: 0.0).

- To choose minimum number of pixels per mask, can turn off with -1, (default: 15).

Finally, the user will have to press "Input parameters" then close the parameters window, then press "Run" button to perform segmentation, as illustrated in center and right figures in figure 4.

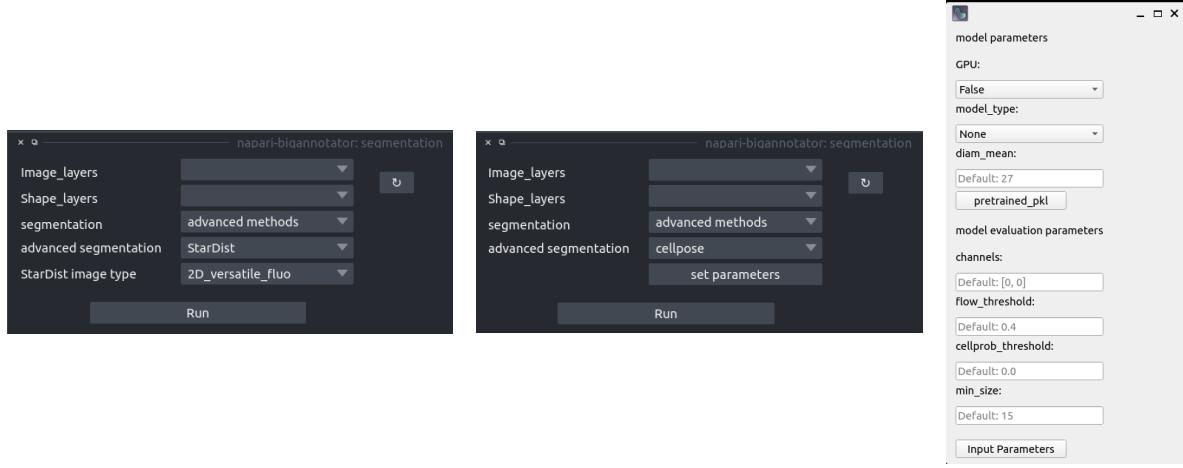


Figure 4: **Image segmentation Plugin (advanced method):**  
**Left:** StarDist, **Center:** CellPose, **Right:** CellPose parameters

## 2.2 Image segmentation results

The figures 5 and 6 illustrate a comparison of the results obtained using classic and advanced segmentation algorithms, respectively.

For the classic algorithms, we observe that, Otsu thresholding poorly detects the blood cells, as illustrated in figure 5 left, a green box is drawn to show how is Otsu algorithm segmenting a cluster of cells as when cell (false positive), in the other hand, Hough transform perform very well but have some drawbacks like: false positives (box in green) where two cells are labeled as one, or some cells get an extra mask that contain parts from neighboring cells (in red).

For the deep learning algorithms, we observe that, Cellpose outperform StarDist and results on the best segmentation results. In figure 6 left, we observe that StarDist detect well the cells, but also detect many false negatives (box in blue), whereas, cellpose detects all cells and have no False pos/neg, and both algorithms detect well the overlapping cells.

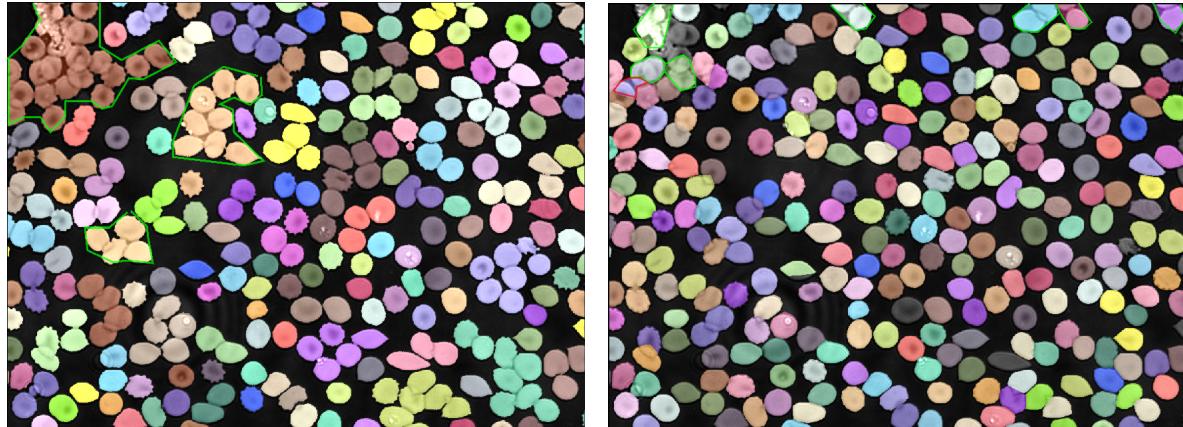


Figure 5: **Image segmentation results (classic methods):**  
**Left:** Otsu thresholding, **Right:** Hough transform

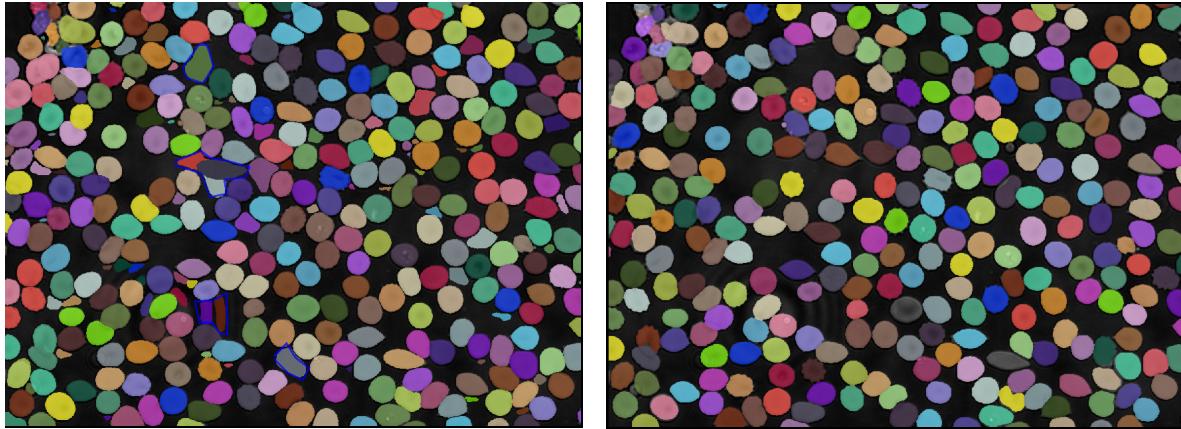


Figure 6: Image segmentation results (Advanced methods):**Left:** StarDist, **Right:** Cellpose

### 3 Cells extraction

After image segmentation, a mask and label images are generated, which indicates cells positions in the image, those images are used for cell extraction and to obtain individual cells images, which will be used later for classification to train/test the classification algorithm. This plugin allows for the following operations:

- **Choice of an image and label layers:** By clicking on the reload button the list of image/label layers will be updated. The label layer image, is the best one obtained from segmentation, in our case its Cellpose, and the image layer is the input preprocessed image.
- **Setting the parameters:** "set parameters" button allows for parameters setting. The parameter's window contain a list of parameters:
  - **Mask:** If 'True', the mask will be used to extract only cells, and outputs a rectangle of fixed size. If 'False', the outputed rectangle will contain the cell with some neighboring cells.
  - **Cell mean:** The cell diameter used for cell segmentation.
  - **Box size:** The size of the rectangle that will contain the extracted cells.
  - **Output directory:** The directory where the extracted cells will be saved as 'png' images.

This is illustrated in left and right figures of figure7.

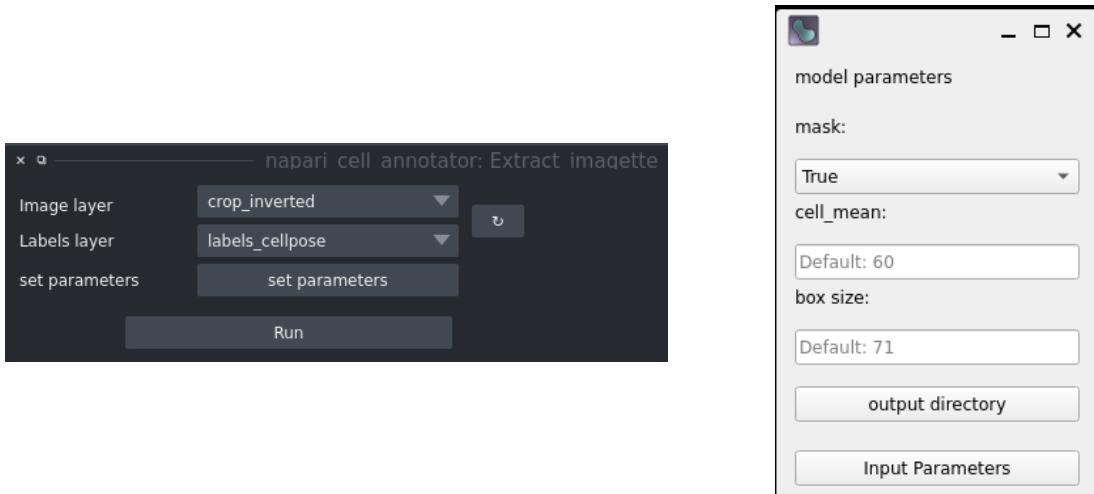


Figure 7: Cells extraction:**Left:** Cells extraction plugin, **Right:** Cells extraction parameters

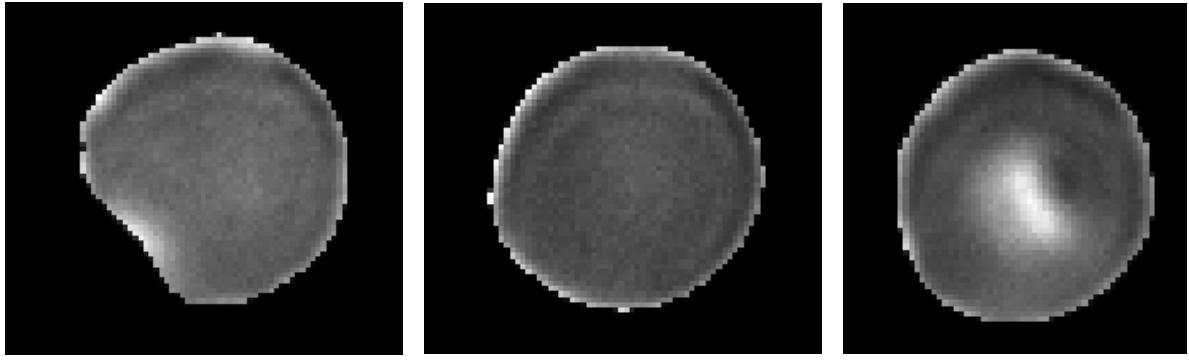


Figure 8: Cells extraction results: Left and Center: healthy cells, Right: infected cell

## 4 Image reconstruction using (FPM)

This plugin could be used even before or after segmentation to obtain a high resolution spatial image and a phase image using Fourier Ptychography algorithm, the recovered spatial image can be used as input to the segmentation algorithm in order to enhance the cells detection by the segmentation algorithms. Also, will be used for cells extraction, which produces the cells that will be used for cell classification. Whereas, phase images will be used after a cell extraction, for cells classification as an additional information about the the cells.

This plugin allows for the following operations:

- **Testing or Training:** By choosing testing, this will allow for loading a pretrained model, and to visualize the 3 images (spatial, phase, Fourier transform), that are the reconstructed images of a low resolution image that was used for training, as well as, producing the high resolution image for the input image.

To do so, the user have to:

- Choose an image
- Set the path to the pretrained algorithm's weights
- Choose a directory where the output images will be saved.

This is illustrated in the left figure of figure9. By choosing training, user have to set the parameters for the training:

- **Index down-sample:** It presents the size ratio between the high and low resolution images.
- **Image size:** The size of the input low resolution image.
- **Array size:** Whose square represents the number of LEDs used to illuminate the sample and produce the input images.
- **Input image directory:** The directory where the input images, that will be used for the training of the model are situated.
- **Output image directory:** The directory where the output high resolution images will be saved.
- **Save weights directory:** The directory where the best model parameters will be saved.

This is illustrated in center and right figures of figure9.

The produced (phase, spatial and Fourier transform) images are illustrated respectively in figure 10.  
[more info .....](#)

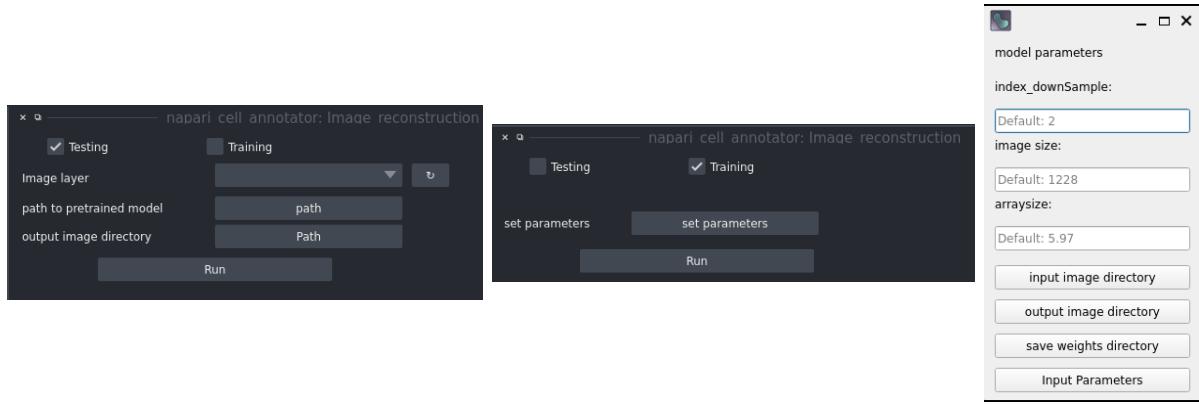


Figure 9: **Image reconstruction (FPM):**  
**Left:** Testing, **Center:** Training, **Right:** Training parameters

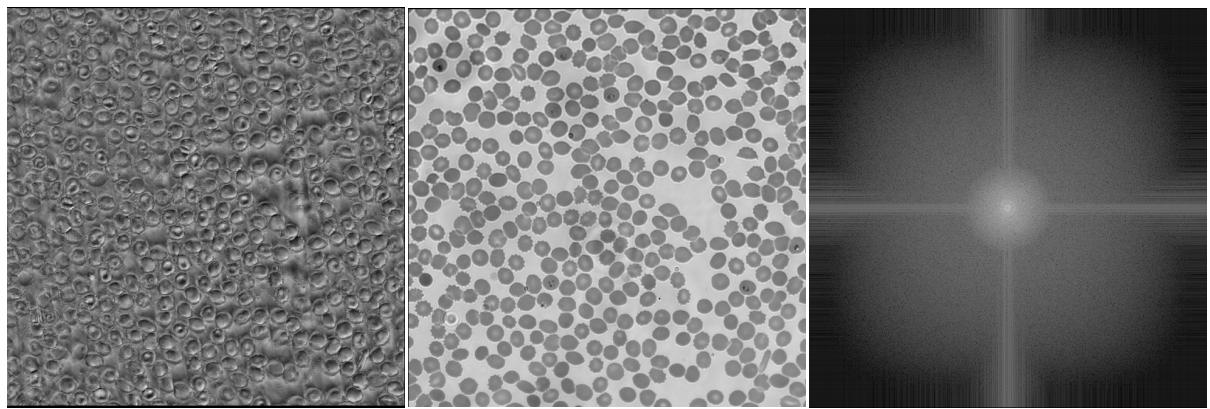


Figure 10: **Cells extraction results:** **Left and Center:** healthy cells, **Right:** infected cell

## 5 Image classification plugin

This plugin is used to allow for either the training or the testing of the algorithm. In case of training, a ResNet152 (supervised deep learning model) is trained on annotated (intensity and phase) cell images composed of 2 classes (infected and healthy), which are annotated by biologist, and represent the ground truth for classification.

In case of testing, a pretrained ResNet152 weights, are loaded and used for predicting the class if unseen cell images.

A good deep learning algorithm, is the one that can generalise to unseen datasets, which means, can predict correctly for test datasets.

This plugin allows for the following operations:

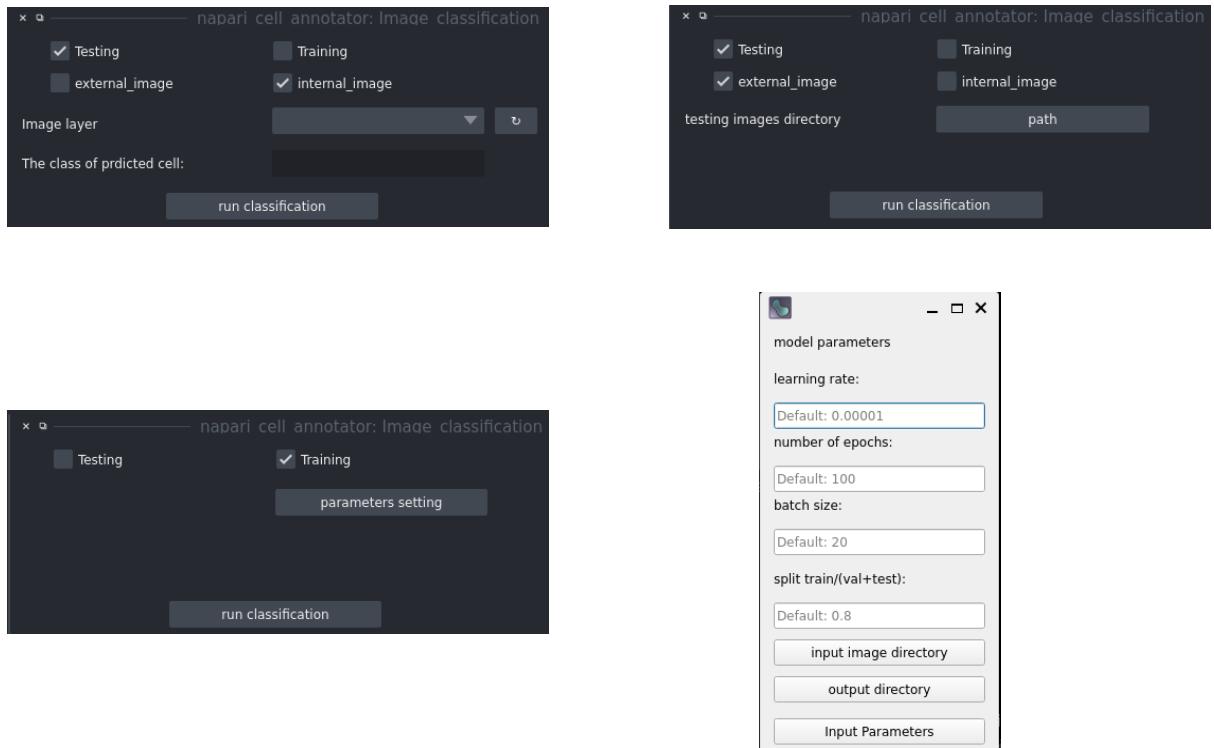
- **Testing or Training:** By choosing testing, a pretrained model's weights are loaded and used to predict the class of unseen images.  
 There are two types of images:
  - **Internal images:** Which means, images from the list of images present on the interface layers list. Thus, the user will have to set the cell image name, then click 'run classification', this will output the class of that cell on the black empty box. As shown in figure 11 top left.
  - **External images:** Which means, to load a list of cell images from an external directory, by indicating the path to that directory. This will output the class of that list of images on terminal. As illustrated in figure 11 top right.

By choosing training, the user will have to set parameters for the training of ResNet152 model:

- **Learning rate:** Which is a tuning parameter in an optimization algorithm, that determines the step size at each iteration while moving toward a minimum of a loss function.

- **Number of epochs:** How many epochs are needed to train the algorithm.
- **Batch size:** Defines the number of training examples in one forward/backward pass. The higher the batch size, the more memory space you'll need.
- **Split training/(validation + test):** If set to 0.8, this mean 80% of dataset is chosen as training set, and 10% for validation and 10% for testing.
- **Input image directory:** The directory where the cell images (phase + intensity) are situated.
- **Output directory:** The directory where the best model parameters will be saved, as well as, the training history (accuracy + loss) at each epoch.

This is illustrated in bottom two figures of figure11.



**Figure 11: Image classification plugin:**

**Top Left:** model testing (internal images), **Top Right:** model testing (external images),  
**Bottom Left:** model training, **Bottom Right:** model training parameters

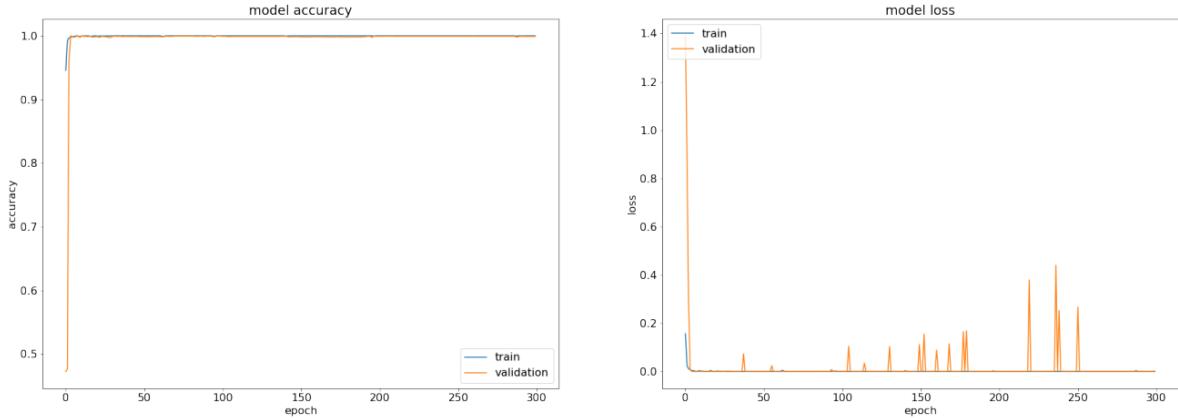


Figure 12: Accuracy and loss classification curves, using FPM reconstructed intensity and phase images.

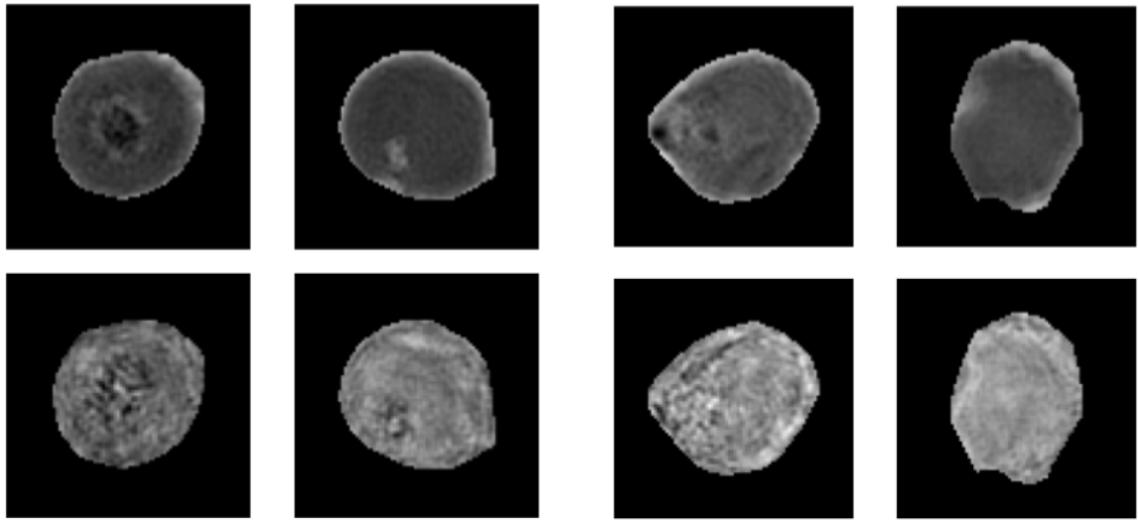


Figure 13: Example of false positives (Left figure) and false negatives (right figure) intensity and phase images, identified by trained ResNet152

## 6 Image saving and loading plugin

This plugin allows the user to save an Image or Labels layer as '.tiff' image and to load an image as Image or Labels layer.

This plugin allows for the following operations:

- **Choose an option:** The user have to first choose either to save or to load an image.
  - **Save an image:** To save images the user have to first select the Image/Labels layers from the layers' list, then to choose a directory where to save those images by clicking the button "dir", then to click "save\_load" button, as illustrated in figure 14 left.
  - **Load an image:** To load images/labels the user have to first select the list of images/labels to load by clicking the button "dir", then select the layer's type (image or label), s.t to load an image as Labels it should be of type "int", then click "save\_load" button, as illustrated in figure 14 right.

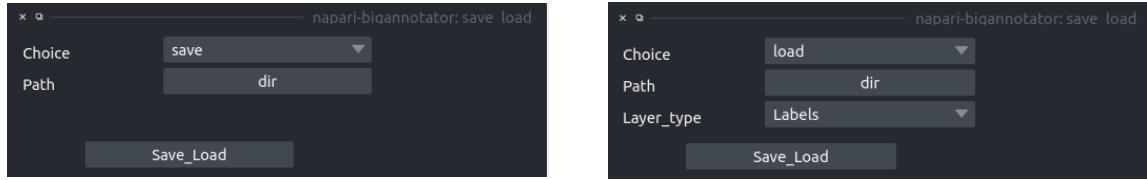


Figure 14: **save\_load image plugin:** saving an image (left), loading an image (right)

## 7 Contour assist plugin

This plugin is an initiation to contour assist and consists of two options: contour suggestion, and add modifications.

This plugin allows for the following operations:

- **Option:** The user have to first choose one of the two options: contour suggestion, and add modifications.
- **Contour suggestion:** This option suggests contours and masks based on Region Growing algorithm to segment the selected regions of the image. To select regions on the image the user have to use the Points layer and to put point on the regions that needs a contour/mask suggestion, then to click the refresh button to update the list of images then select the appropriate image layer, then the user have to set a threshold that defines the gray difference tolerated between a seed and the neighboring pixels, then click "Run". Two Labels layers (contours and masks) will be added to the layers' list, the user can use the brush tool to adjust the suggested mask/contour, as illustrated in figure 15 left.
- **Add modifications:** This option will allow the user to merge two Labels images the raw label (before modification using contour suggestion) and modified label (after modifications). To do so, the user have to click the refresh button, then to select the appropriate raw label and modified label images, then click "Run" button, this will output a final labels layer containing all the labels, as illustrated in figure 15 right.

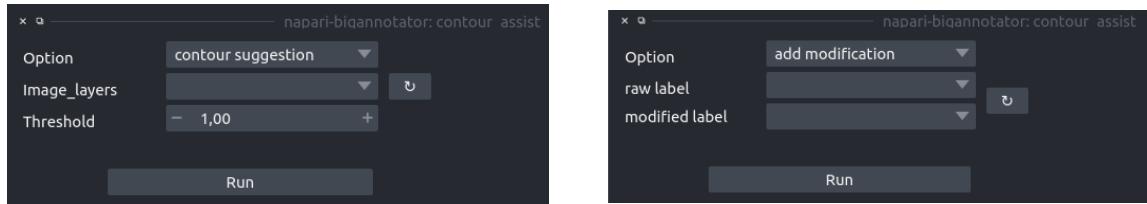


Figure 15: **contour assist plugin:** contour suggestion (left), adding modification (right)

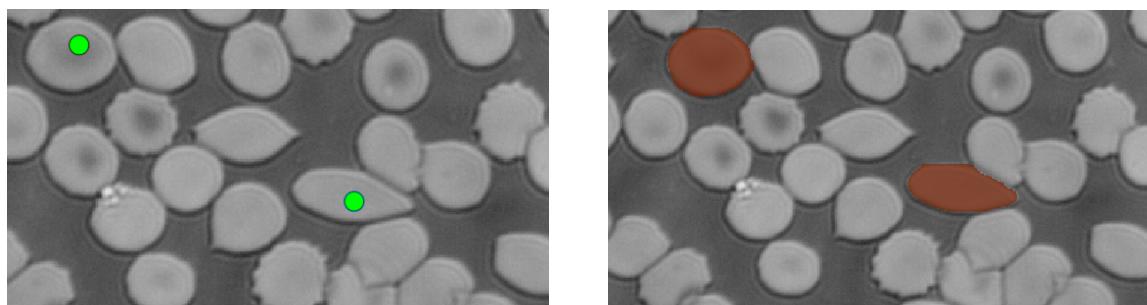


Figure 16: **Contour suggestion example:** selecting seed where non segmented cells are situated (left), masks obtained after applying region Growing algorithm (right)

## References

- [1] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, Maria Kovacs, Ede Migh, Allen Goodman, Tamas Balassa, Krisztian Koos, Wenyu Wang, Juan Carlos Caicedo , Norbert Bara, Ferenc Kovacs, Lassi Paavolainen, Tivadar Danka, Andras Kriston, Anne Elizabeth Carpenter, Kevin Smith and Peter Horvath. "nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer". Cell Systems, 2020.
- [2] Uwe Schmidt, Martin Weigert, Coleman Broaddus and Gene Myers. "Cell Detection with Star-Convex Polygons". Medical Image Computing and Computer Assisted Intervention, 2018.
- [3] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara and Gene Myers. "Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy". The IEEE Winter Conference on Applications of Computer Vision (WACV), 2020.
- [4] Feixiao Long. "Microscopy cell nuclei segmentation with enhanced U-Net", BMC Bioinformatics, 2020