Guide to Mistos

# Scope

Mistos is a **M**icroscopy **I**mage **St**orage and **O**perating **S**ystem with the purpose to standardize image storage and evaluation. Its’ features include:

* Import and store images in the integrated database
* Evaluate images individually using
  + Manual image segmentation
  + Semi-automated segmentation
  + Fully automated segmentation utilizing Deepflash (Cuda ready GPU’s only)
* Estimate a ground truth segmentation by processing multiple experts’ segmentations
* Create and store experiments
* Export
  + Raw Images
  + Experiments, you may export
    - Images
    - Binary label masks
    - Images and binary label masks cropped or padded to match a specified image size

Please refer to the readme in the main Mistos folder for details about the setup, troubleshooting, or technical details. If questions arise or issues occur, please submit them on our github page (<https://github.com/Maddonix/mistos_2>).

# Getting started

After installation, start Mistos by running the “start\_mistos.bat” script. Three command lines will be opened:

1. Mistos Frontend
2. Mistos Backend
3. Mistos Fileserver

After the system is initialized, you can access it from your Internet browser by calling the address “<http://localhost:4200>” (if you receive a 404 error try typing just “localhost:4200”).

The following Demo Experiments will show you how to use Mistos. Since later demo experiments use files created in previous examples, pleas perform experiments in the right order.

**To-Do: Add Images to guide**

# Principles of Mistos

Mistos works with a standardized format and evaluation method. Therefore, there is a list of guidelines. If followed you will achieve good results with Mistos, and also be able to standardize your own workflow.

1. If you want to measure the intensity of your images, make sure to add a background layer. Only use the label 1 in this mask. Make sure, that all channels are just background signal in the marked area.
2. If you work on a Z-Stack, keep in mind that you may label all slices individually or label them at once by checking “n-dim” in the left toolbar (only visible if a label layer is selected)
3. Functions in the Napari viewer will always use the selected layer on the left side if there is no possibility to choose.
4. Don’t choose multiple layers in the Napari viewer at once. This will produce unwanted behaviour.
5. We always measure one image modality (e.g. nuclei) at a time. For this, we create a label mask and use a different label for each entity (this means: first nucleus has label 1, the second 2, …). If we want to measure two different entities in an image use different experiments or at least different experiment groups.
6. While Napari Viewer is opened, you can not work with the browser app.

# Demo Experiment 1: Count nuclei

During this demo you will:

* Import images
* Create an experiment
* Create 4 experiment groups
* Add images to these groups
* Segment images manually
* Train, apply and reapply a random-forest classifier for segmentation
* Segment images using StarDist (<https://github.com/mpicbg-csbd/stardist>)
* Segment images using DeepFlash
* Export your experiments results

## Image import

* In the navigation bar, click on “Import”
* Now go to the folder “Tutorial/Demo Experiment 1”
* Select the images 0001.png, 0002.png, 0003.png and 0004.png
* drag and drop them in the drop zone and press the “Upload” button
* The images will now be imported, you can track the importing progress in the “Mistos Backend console”
* Click on “Images” in the navbar and verify all four images have been imported

## Create an experiment

* Click on “Experiments” in the navbar
* Click on New Experiment
* Enter the following
  + Name: “Demo Experiment 1”
  + Tags: “Nuclei”
* Create the Experiment
* Click on the new experiments “Detail” Button

## Plan your experiment

We want to segment our experiments in three different ways and compare the results. Therefore, we create three groups.

* In the “Group Summary” area, click on “Add Group” three times
* Name the groups “Manual”, “Computer assisted”, and “Neural Network StarDist” by clicking on rename
* Scroll down to the first group, click on “Add Image” and select all four images
* Repeat this for the other groups

## Getting started with the Napari Viewer

In every experiment group, we find one row for each image. By clicking on the image, we can go to the image’s detail page or delete it from this group. On the right side of the row, we find three fields:

1. “None”: This indicates the segmentation layer we use for the result calculation later. By default, no layer is selected.
2. “BG”: This field indicates if our image has a background layer saved. If we have one the text is green, otherwise it is red
3. “View”: By clicking on this field, you open the image in the Napari image viewer.

Choose the first image and click on “View”. Now we can start labelling our image:

* Wait a few seconds for the image viewer to start. If nothing happens, look at the task bar if the viewer has been started minimized.
* For an overview of all functions, refer to the Napari viewer cheat sheet. On the left side, we see the menu bar in which we can change settings like opacity and contrast. Below we see a list of all channels, we only have one channel. Between those, we find four buttons, click on the one with the label tag (not the dots, not the white polygon, not the bin; you may hover your mouse over them to see tooltips. The correct button has the tooltip “New labels layer”)
* A new layer was created, make sure it is active (blue). If not, click on it to activate.
* **Background:**
  + On the left side, activate the paint mode (Shortcut: press “p”)
  + Label representative background areas, now click on “Save BG Layer”
  + Note: The layer now “disappeared”. While it is not visible after saving, its still there. You may want to make it visible again by clicking on the crossed eye left to the layers name.
  + Note: If “BG” is still red in the experiment overview, press the “Refresh” button next to “Groups & Results”

## Manual Segmentation

* Create another layer
* Name it “Manual Segmentation” by clicking on its name
* Now label all nuclei in the image (since its just a tutorial, maybe not all of them, just so you get to know the drill)
  + Activate paint mode
  + Zoom into the image until you can comfortably label the nuclei
    - Note: To navigate the image while another mode is active, press and hold Space
  + After you labelled the first nucleus, go to the left navigation bar and choose the next label by clicking on the plus next to label (Shortcut: “m”, automatically selects the next label number which is not already used in the image)
    - Lifehack: Use the same label for all nuclei and press the “To individuals” button as soon as you are finished to save you some of that sweet, sweet time!
  + You may change the cursor size and shape in the left-hand menu under “brush size” and “brush shape”
* If you are finished, click on the “Save Label” button. After the label layer has turned invisible, the image is saved. Now close the viewer window.
  + Note:
    - After a layer is saved it cannot be altered. Anyhow, you may make changes and save it again as a new layer. To delete layers, you currently must go to the images detail page and delete it there.
    - Changing a layer’s name after you already saved it will have no effect. You must change it on the images detail page!
* Repeat this for all 4 images

## Computer assisted segmentation

* Open the first image again
* Create a new label layer and name it: “Random Forest Train”.
* Label some background areas with the label **1** 
  + This is important as the label 1 is always used to train the background class
* Find a bright nucleus, and label it with the label 2. Be precise!
  + Note: While training our classifier, we do not use different layers for each nucleus. Here every layer represents a different entity (e.g., nuclei, cells, fibres)
* Select label 1 again check the “preserve label” option in the left-hand menu.
* Now label the background around the cell. While “preserve label” is checked, old labels can not be overwritten by new ones.
* Repeat this procedure for very weakly visible nucleus (which you still would identify as one!)
* You may repeat and nuclei as you wish
* Now, save this label and go to the image segmentation menu in the bottom toolbar
  + (second from left)
* Fill out the options
  + Layer image: “Image”
  + Layer labels: “Random forest train”
  + Tags: “nuclei” (To do: delete)
  + Multichannel: “false”
* Click on segmentation (Depending on your computer this may take some time in which the app might appear to freeze. Please be patient.)
* Now you have a binary mask called, rename it to “Computer assisted segmentation”
* We will do a little post processing now:
  + Remove small objects
    - Choose a minimum pixel amount (eg. px=75) and a maximum distance at which pixels will be regarded as group (eg conn= 2)
    - Click on “del small obj”
  + Fill holes:
    - Choose a maximum pixel amount until which a group of pixels surrounded by a label will be labelled the same
    - Choose the distance
    - Click on “del small holes”
  + Separate labels:
    - If you find two nuclei whose labels are touching, delete the connection with the erase mode (Shortcut: “Alt”)
  + Delete completely wrong labels and add missing ones
  + Now click “To individuals” followed by “Save label”
* Open the next image
  + Choose the classifier we trained right now (bottom toolbar, first tool from the left)
  + Make sure your “Image” layer is selected
  + Click “LoadSegment”
  + Note
    - Currently we can only train a random forest classifier on one image, therefore it will possibly perform bad on other ones. If your images are very, very similar it can work quite well.
    - If you have a powerful computer, you can also train a new classifier for every image
    - If you want to remove trained classifier, you may do so in the classifier menu in the browser application
  + Rename the label to “Computer assisted segmentation”
  + Repeat the post processing like explained for the first image
* Repeat for the last two images

## Neural Network Segmentation: StarDist

StarDist is a Tensorflow based neural network specialized on oval shape recognition. While a 3D version of the network is available, Mistos currently supports the 2D version only. All images are processed to maximum z-projections before predicting the cells. In case of larger z-stacks this will not work! Now, let’s get started:

* Open the first image again
* Make Sure the “Image” layer is selected
* Click on “Nuclei Segmentation”
* Check your results and make small adjustments if necessary
* Click on “Save Label”
* Repeat for the other images

## Evaluation

To evaluate the segmentation, choose the corresponding labels for each group and export the results.

* Click on the field for layer selection (initially it shows “None”) and select correct layer for each group
* Click on “Generate result report”
  + Note: After you changed anything in the experimental setup (e.g., you changed a layer, added or removed an image, or anything else, you have to click this button again.
* Now click on “Export Results”
* Leave the dialog as is and click on “Export”

After the dialog closes, an excel file will have been exported to the export folder. The default export folder is in your user’s home profile (e.g. “C:/Users/{your\_username}/export”). You may change this by opening the “config.json” file in the mistos\_2 folder (you can open it with a text editor). Here you may change the filepath to anything you like, just make sure the folder exists and use “/” or “\\” but not a single “\”.

In the Export folder you now should find a folder “1\_Demo Experiment 1”. This directory contains a folder for each experiment group and a summary excel file. The experiment group folders are currently empty, since we did not export anything else during the export dialog.

**Note: If you export the same experiment again, this folder will be overwritten. Make sure that you move your files away as precaution.**

Structure of the measurement file

Mistos will always return your measurements in the structure we will now look at. While this format may seem cumbersome for some use-cases, it is possible to address almost all questions like this. Furthermore, you may develop your own routines to work with this data format and automate some steps.

Each row represents a single label. The **image** and **group** **columns** tell us, which image and group the measurement originates from. The **n\_pixel column** tells us how many pixels in the image are flagged with the label (Note: The number of pixels is calculated over all z-slices. If you want to calculate the area of a label, you might want to divide by the number of slices). The **n\_z\_slices** column tells us how many z\_slices or image has. Additionally, each channel has the following columns:

* **{channel\_name}\_sum\_intensity:** sum of all pixel values for this label
* **{channel\_name}\_mean\_background\_per\_pixel:** the mean pixel value of the background labels in this area (if background was provided)

Common Tasks:

* Background subtraction (in each row for each channel):
  + Back\_subtracted\_signal = sum\_intensity – (n\_pixel\*mean\_background\_per\_pixel)
* Object count:
  + Count the rows for each image in each group
* Sum intensity in image:
  + Calculate background subtracted signal intensity and sum it up for each image

Troubleshooting:

* You have labelled many nuclei, but see only one measurement
  + Make sure you have a layer with a different label for each object
  + If you forgot:
    - open the image
    - select the layer
    - click to individuals and verify the result
    - click save label
    - go to the image’s detail page
    - delete the wrongly labelled layer
    - assign the new layer in the experiment
    - Generate & Export your results again

# Demo Experiment 2:

In this experiment we estimate a ground truth of different annotations. This is a common use-case in the context of deep learning. In general, a network should be trained with perfect data. Since every expert will label images more or less differently than another expert, there exists no ground truth for labelled microscopy images. Nevertheless, we can use the labels of different experts and estimate a ground truth. For this, several methods exist and Mistos utilizes the STAPLE algorithm.

We will simulate three experts’ opinions and use the three label layers we created in Demo Experiment 1 to estimate a ground truth. Then, we will export the images (as tiffs) and the labels (as binary masks) so we could utilize them as training data (e.g., for training a DeepFlash model).

## Ground Truth Estimation

* In the top navigation bar, click on DeepFlash
* Choose the tab “Ground Truth Estimation”
* Set the number of experts to 3
* Add the four images we annotated in Demo Experiment 1
* Click “Create new Form”
* Choose your manual segmentations for expert 0, the computer assisted segmentation for expert 1 and the StarDist segmentation for expert 2

## Export results

* Go to Experiments and create a new experiment called “Demo Experiment 2”
* Create a group and name it “ground truth estimation”
* Add the images to the group
* Add the layer ground truth estimation to the images
* Click “Generate Result Report”
* Click “Export Results”
* Check all boxes and click Export
* Note: if you inspect the exported masks and they appear black, open them in a “professional” image viewer (e.g., ImageJ). Pixel values in masks are always 0 or 1. If your image viewer does not adapt its contrast, the mask will be not or only barely visible.

# Demo Experiment 3:

Let us assume we used many of the images from Demo Experiment 2 and trained a DeepFlash model with it. To import a model, go to “Import” and select the tab “Deepflash Models”. Now enter the path to the folder and click on “Import from filepath” (you can just open the folder in your explorer, click on the bar where you see the file path, and copy and paste it from there; **use the provided model**). As soon as you see an “OK” message you may click on classifiers. Verify that there is a classifier with the name of the folder you imported and the type “deepflash\_model”. To classify images, perform the following steps:

* Click on “Deepflash” in the top navigation bar
* Select your model
* Click on “Add Images” and select the four images
* Now click predict, you can track the progress in the “Mistos Backend” console.
* As soon as the process is finished, go to “Experiments”, and create “Demo Experiment 3”
* Create a new group called “DeepFlash” and add the four images.
* Select the layers starting with “df\_seg\_”
* Generate the result report and export the results

# Demo Experiment 4:

Lastly, we want to archive complete Mistos experiments and images including all layers and import them again. If experiments are exported, all corresponding images are also exported. During import, they are also imported again, even if the original images are still in the database. Currently, file sizes of exported images and experiments are quite large, and the same image gets imported multiple times if it’s in the experiment more than once. We are working on a more optimized solution.

## Image Export

* Go to Images
* Go to an images detail page by clicking select in the image’s row and then on detail on the right side
* Click on “Export to Export Folder”
* Verify that your export folder contains a file with the following name: “mistos\_image\_{id}\_{image\_name}.pkl

## Image Import

* Go to “Import”
* Go to “Mistos Importer”
* Enter the complete filepath into the input field below “Import archived Mistos Images”
  + E.g., “C:\users\{username}\export\mistos\_image\_{id}\_{image\_name}.pkl”
* Verify the image is now available in the “Images” screen

## Experiment Export

* Go to “Experiments”
* Select an experiment and go to its detail page
* Click on “Export Mistos Experiment”
* Verify that your export folder contains a file with the following name: “mistos\_experiment\_{experiment\_name}.pkl”

## Experiment Import

* Follow the same steps as for image import, but use the input field below “Import archived Mistos Experiments”