OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

BIOFLUIDMECHANIK

AI Nuclei Detection

User Manual

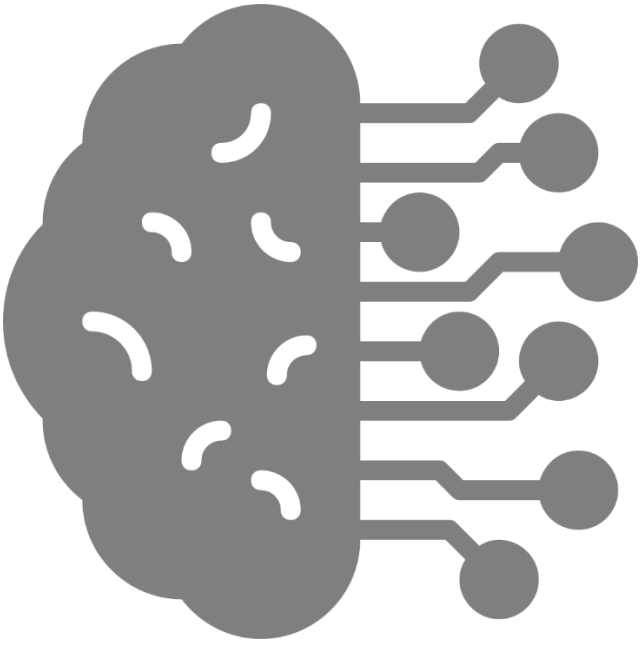


Table of Contents

Application

The present program is a complete desktop application for automatic detection and evaluation of nuclei deposed on gas exchanging fibers of ECMO membrane lungs. Nuclei are stained with 4',6-diamidin-2-phenylindole (DAPI) and then visualized using a fluorescence microscope. The microscopic images are automatically z-staggered by the microscope and exported as "enhanced depth of field" (EDoF) gray scale image. Condensed nuclei and decondensed nuclei are automatically detected based on a Mask-RCNN Deep Learning (DL) model with ResNet50 + FPN as backbone. After DL detection and evaluation, the generated original data is exported as CSV, XLSX, and as JSON file. The JSON file is primarily used for later postprocessing, e.g. automatic creation of bar charts and boxplots which are included in a PDF report. In addition, the JSON file includes meta data for improved traceability of the conducted research.

Computer Requirements

5.3 GB of hard disk space is required to run the software as a packed Windows executable. The software was developed for PCs running Microsoft Windows 10 or later. It is recommended to use a CUDA-supported GPU (NVIDIA GPU) as this boosts the software performance significantly and a CPU with a minimum of 16 GB of RAM. The software automatically checks whether such a GPU is available and then outsources the evaluation process to the GPU automatically. If no CUDA-supported GPU is detected, the evaluation is carried out by the CPU. Depending on the CPU and the GPU, the software adapts its evaluation processes to work as efficient as possible on the machine.

Starting the Software

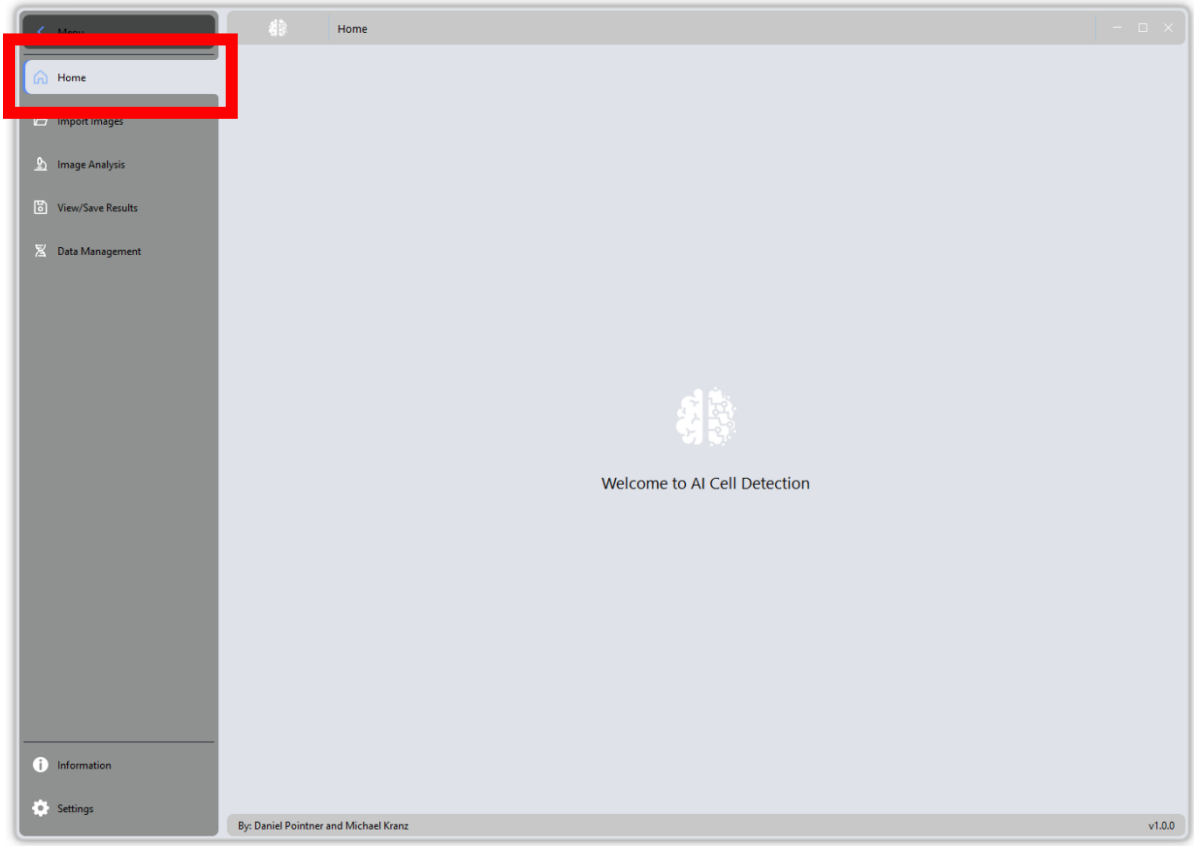
To start the software, double click on the AI Nuclei Detection.exe or the desktop shortcut. After that, a loading screen (Splash screen) is displayed while the software is booting in the background. The loading screen automatically disappears when the software is initialized correctly and the graphical user interface (GUI) of the software appears.

Control Panels

The GUI is used for interaction between the user and the program. It comprises of control panels (pages). The welcome page is shown at first. Text information on each individual page guide the user step-by-step throughout the program.

Welcome Page

After starting the application, the Welcome Page appears. The Welcome Page is always accessible by clicking the house icon (red marker) in the left menu.



Import Page

Clicking the folder icon navigates the user to the import page. The software indicates if images are already imported. The user must insert a trial name for the project before an import is possible. The trial name is later used for the analysis, export, and data representation. After name the trial, the images can be imported to that trial by clicking the “Import Images” button. A dialog window is opened, and an image directory can be selected. The impot can also be aborted by clicking the “Stop Import” button. After a successful image import, the central icon changes to a green check mark. Multiple trials can be defined by inserting another trial name and importing images to that trial.

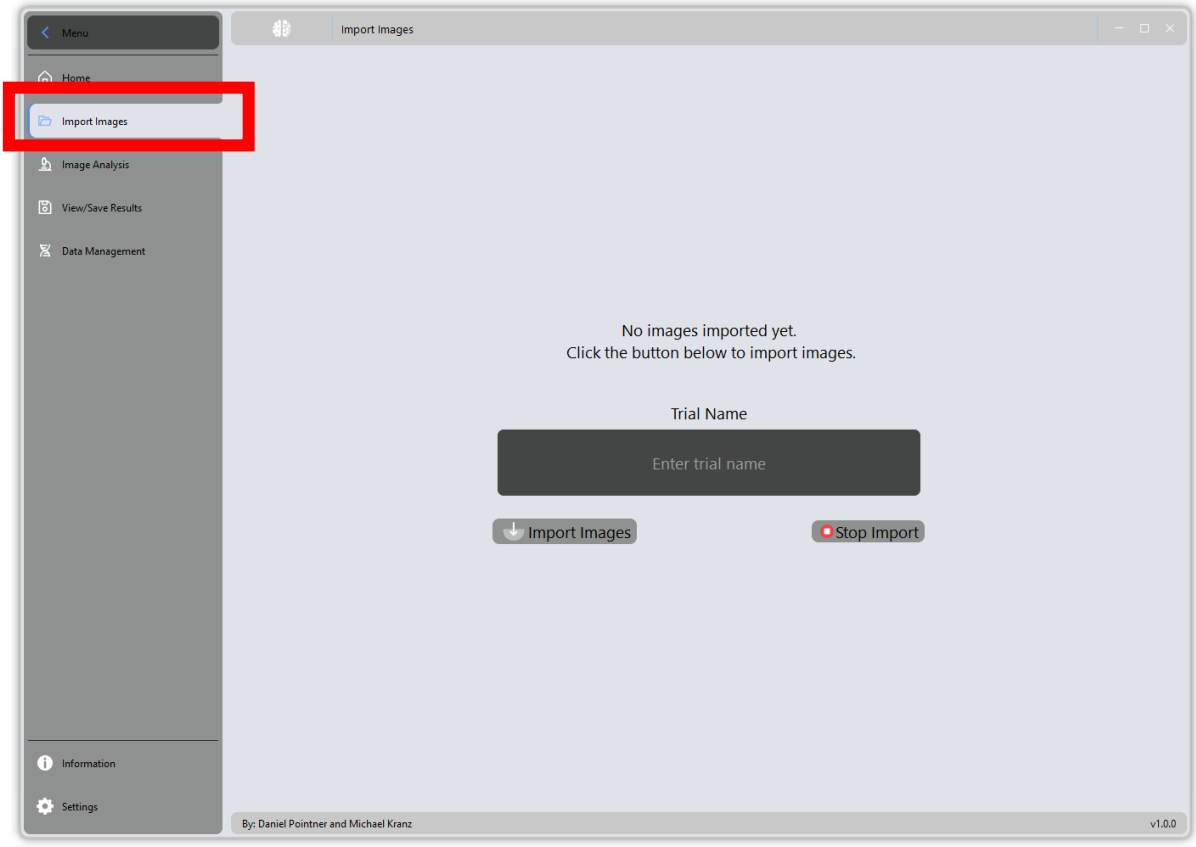
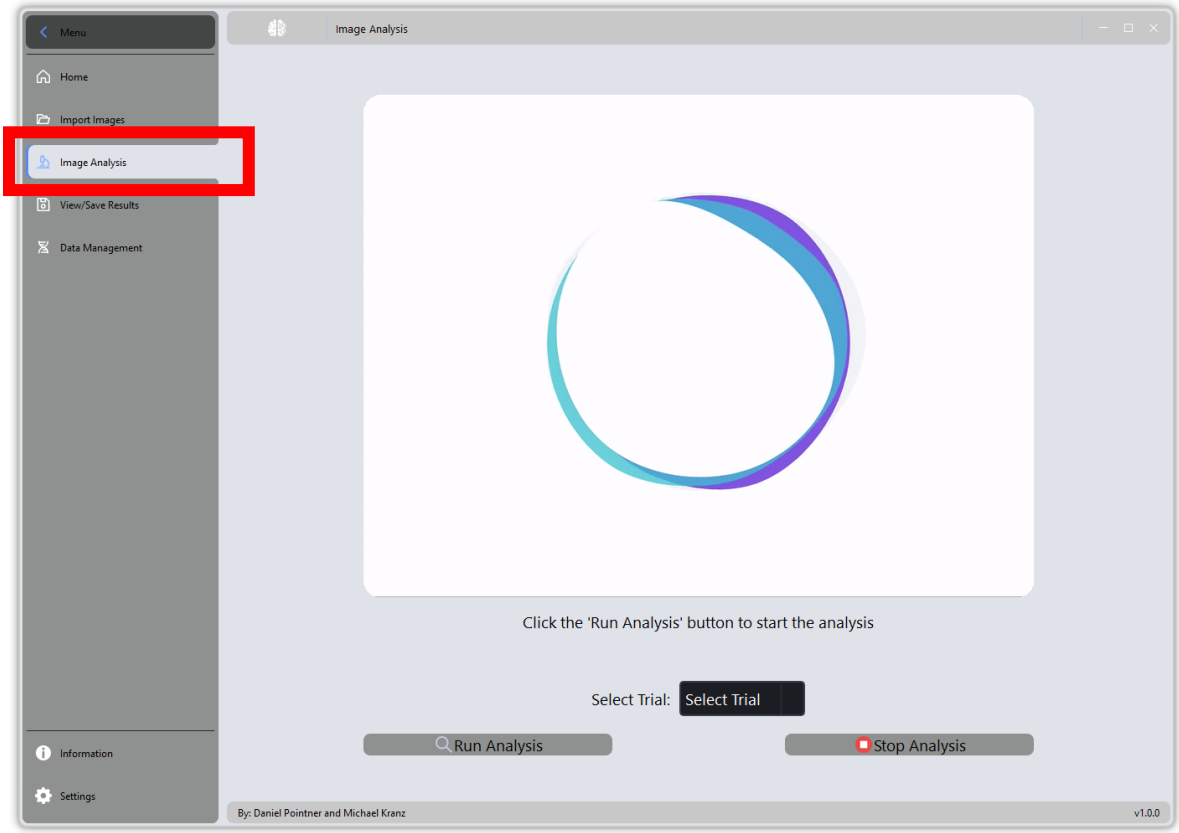


Image Analysis Page

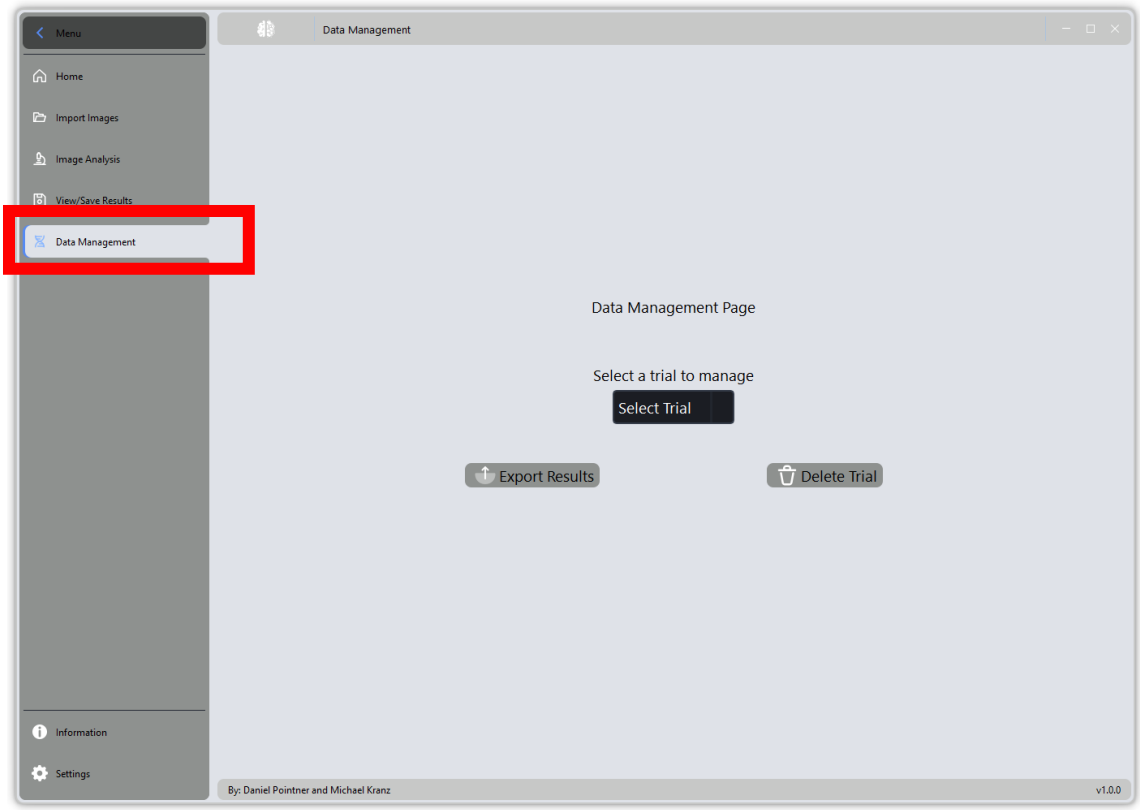
The Image Analysis Page is the key page for performing DL image inference. The user must select the trial name from the drop-down menu and then click the "Run Analysis" button. In addition, it is possible to select "All Trials" in the drop-down menu to run inference with all imported trials. Note, this will analyze all trials in the dropdown menu. If there are remaining trials from other analyses, these are also considered. Beware of potentially overwriting the image analysis results of old trials!
How to remove irrelevant trials: see the "Data Management" page.



Results Page

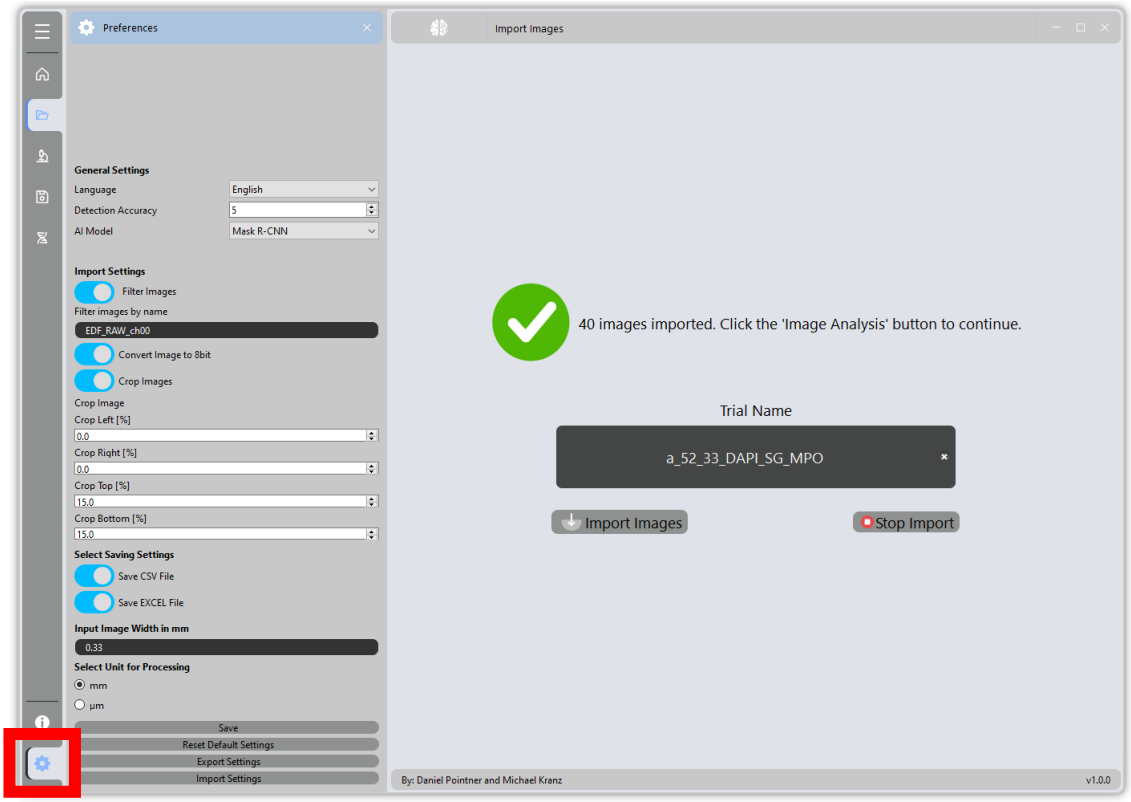
On this page, you can view the results of the image analysis by selecting the corresponding trail in the drop-down menu and clicking on "Show results". By clicking on "Save results", the predictions made by the AI, which you can see in the viewer, are saved in the project folder (internally to the program). All data belonging to the project can be exported or deleted via the data management page. It is also possible to click on "Generate report" to automatically generate a PDF report that clearly aggregates all the data for the selected project. The report is automatically opened in the PDF viewer once generation is complete. This is also included when exporting the project data in the exported project folder.

Data Management Page



You can use this menu to manage, i.e. export or delete, all data associated with a created project.
Select the project you want to save from the drop-down menu and click on "Export results". You can now define the location of the export. The export then takes place in this folder, whereby a subfolder with the name of the selected project is automatically created. All data belonging to the project (including data such as the number of cells and AI predictions as well as the PDF report - if generated via the results viewer!). You will find the AI predictions in the "plots" subfolder and the PDF at the top level of the project folder.
To delete an experiment, also select the corresponding experiment from the drop-down menu and then click on Delete experiment. You must then confirm the deletion process.

Settings Menu



The settings menu is displayed when you click on the gear wheel icon at the bottom left of the GUI window. Following settings can be made:

General Settings

You can change the language of the program if you wish. Changing the language requires the software to be reopened for the changes to take effect.
Another option is to set the "Detection accuracy", where a higher value means a more restrictive analysis policy. Setting higher values internally increases the threshold for the AI's confidence score, meaning that low contrast cores or instances for which the AI is uncertain whether they are nuclei will be rejected.
It is also possible to change the AI model used for the analysis.
Note: The default model is Mask R-CNN, which is validated for accurate detection of nuclei. YOLOv8 is a lightweight model with significantly lower inference time, but it is not yet validated and should be considered a beta version.

Image Compression

To reduce processing effort and assure AI model accuracy, a compression of the image bit depth to 8 bit can be made. If the images are already in a 8 bit format, the software detects it and thus will not perform compression during import.

Image Cropping

In this certain use case of z-staggered images of a three-dimensional specimen, a significant distortion exists in the top and the bottom area of the image. These areas cannot be investigated adequately. For this reason, an image cropping function was integrated in the software. To activate the cropping, the user must click the corresponding toggle switch. The user can crop up to 25 % from the image width from the left side and the right side as well as up to 25 % from the top and bottom of the image. Selecting 0 % means that no cropping from this side is applied. After setting the cropping values the used must click the “Save” button to apply the settings.

Select Saving Settings

The user can select the files for export by clicking the corresponding toggle switch. The generated data can be saved as CSV and/or formatted XLSX file. An automatic PDF report with boxplots and stacked bar charts can also be created. After inference, the cells detected by the AI model are highlighted in the analyzed images for representation in the “Results” panel. The analyzed images with additional information e.g. cell count, cell area etc. can be stored by clicking on the corresponding button “Store Results”, which will store the in a subfolder called “plots” in the project folder.
The same applies to the PDF Generation. Clicking on “Generate Report” will invoke the PDF generator and stores the PDF in the project directory. When exporting the project on “Data management” Page all files created during analysis, as well as (if generated the “plots” and PDF report.

Image Size

The software calculates the area of detected cells. For reference, it is necessary to insert the image width. Only numerical entries are accepted by the software.

Processing Unit

For data representation in the generated report, it is necessary to select the correct unit. The used can select between “mm” and “µm”.

Export Settings

The complete configuration of the software, the import, cropping and compression settings as well as the processing settings can be exported for storage or for later reloading. Settings are stored as a JSON file with current timestamp.

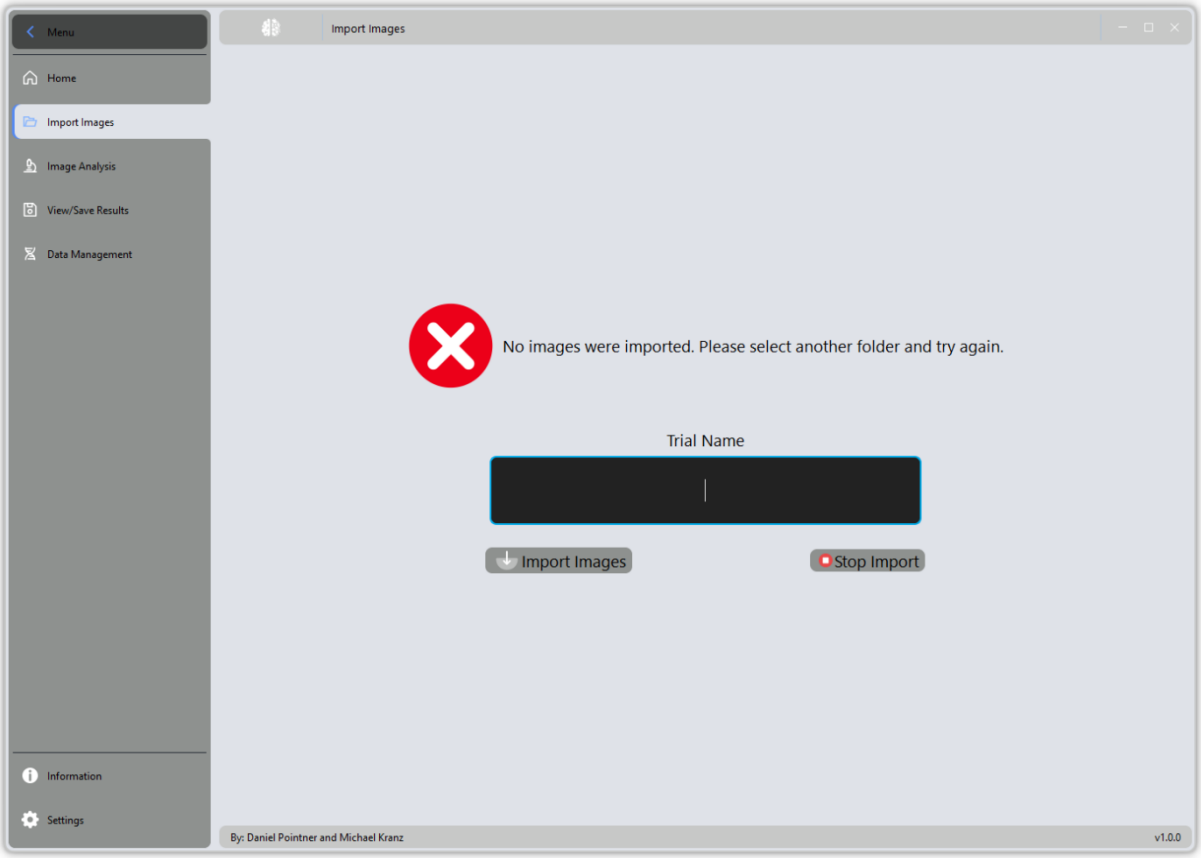
Import Settings

Click on this button to select a JSON file to load preconfigured settings of the application. The software checks the integrity of the settings file and warns in case of corrupted file.

Error Handling

This section contains all potential error messages the software will indicate in case of unintended usage or in case of processing errors and possible solutions to solve the issue.

ERROR: "No images were imported. Please select another folder and try again."

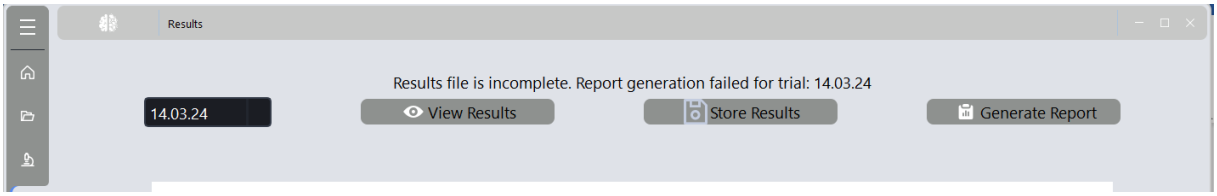


Upon entering a trial name and selecting a folder with no images or no images matching the filter set in preferences the import will fail resulting in the error message above. Consider selecting another folder for import or try adapting the filter "Image name filter" in the Preferences tab.

ERROR: "No results to display for the selected trial. Please run the analysis first."

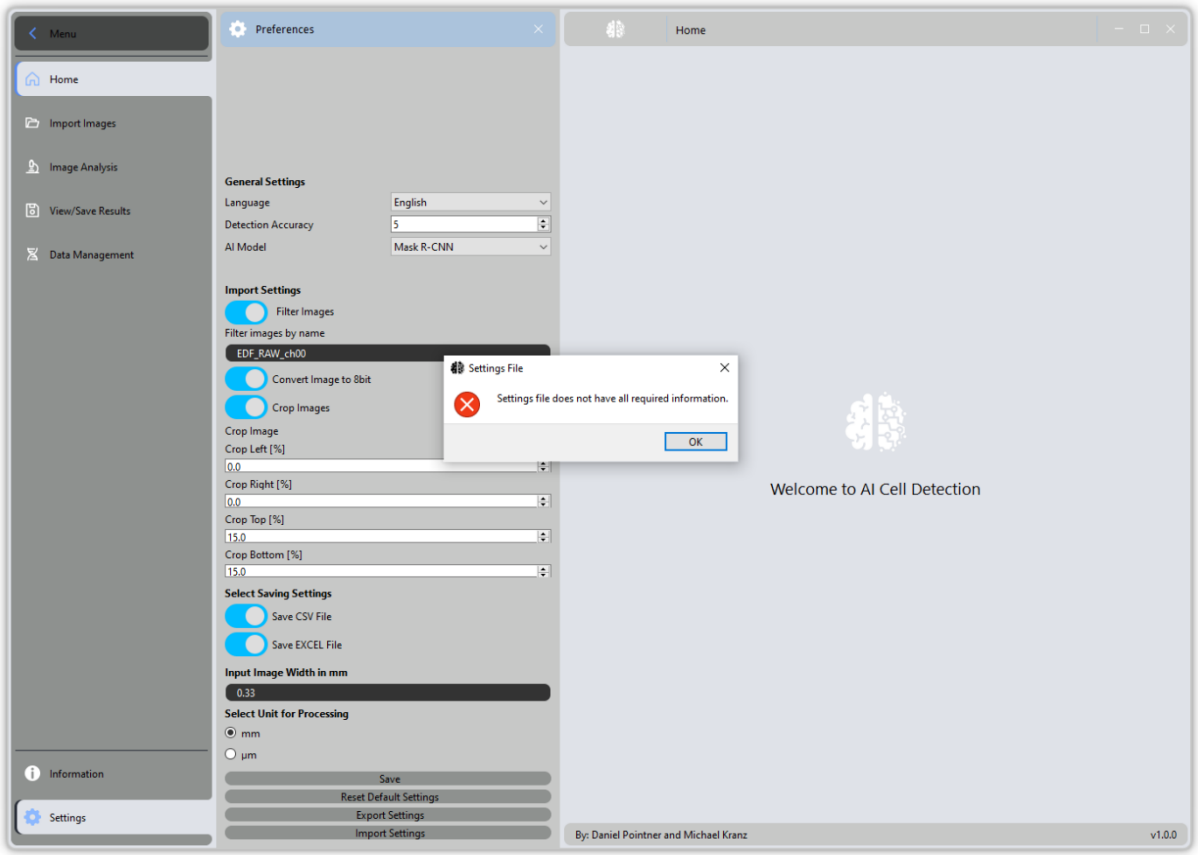
You will get this error if you have imported images but have not run analysis on them yet. Instead, you proceeded to view results, which are not existent at this stage. Switch back to "Image Analysis" page and let the software analyze the complete import to view its results and to be able to store results and/or generate PDF report.

ERROR: "Results file is incomplete. Report generation failed for trial: X"



When trying to generate the PDF report for a trial/project which does not have a results file (temporary instances_results.json contained in the project folder / dynamically generated during inference) or incomplete results this information will appear as a status message on the "Results" page.
This is most likely the result of having interrupted the process and attempting to generate a report for the selected import. You may switch back to the Image Analysis Page, select the trial in question and rerun the analysis to solve this issue.

ERROR: "Settings file does not have all required information."



If you try to import a JSON file which does not match the format expected by the software, you will receive this warning. Ultimately, the current setting will NOT be changed.

HOW-TOs

Workflow for Train Data Generation

- Generate weak labels using Nuclei Annotation Toolbox
- Export these labels (images and annotations) contained in output folder
- Open Computer Vision Annotation Tool (CVAT) and create a project which must contain at least one class (e.g. class 1: "nucleus"; class 2: "decondensed nucleus")
- Create a new task in CVAT and import the images (Note: the web viewer can only render 8-bit tiff files, if you generated 16-bit tiff files during image capturing you may want to convert them automatically during import to Nuclei Annotation toolbox by setting the checkmark "Convert to 8-bit" in the Import Dialog)
- Upon creating a new task in CVAT upload the auto generated annotation file (instances_default.json) to CVAT using the "Actions" > "Upload Annotation" button and selecting COCO v1 as file format)
- This will load annotations for class 1 (nucleus) and show them as polygons
- Modify polygons by removing false predictions generated during auto-annotation, correct overlapping nuclei and defining cell classes (e.g. selecting bigger nuclei and assign them to class 2 (Note: selected AI model to train MUST be able to perform instance segmentation to segment AND classify different nuclei types) or adding missing nuclei by drawing new polygons
- Upon correction export the Task Dataset by clicking on "Actions" > "Export task dataset"- be sure to select "Export Images" as this will lead to export both the already converted 8-bit images in "images" and COCO annotations in "annotations" folder of your task dataset

Workflow for AI training

- Create a virtual environment with Python >=3.10 (**`python -m venv venv`**) and installing the packages from requirements.txt (**`pip install -r requirements.txt`**)
- Put your train data into the following structure (on the same level of "**`train.py`**")

  |- **train_data**
  |--- ***annotations***
  |------ instances_default_train.json
  |------ instances_default_val.json
  |--- ***train***
  |------ all images to train the model on
  |--- ***val***
  |------ all images to test the model on

- In order to obtain instances_default_train.json, instances_default_val.json you may need to split the COCO file downloaded from CVAT to your specific usecase;

Besure to balance your dataset if it contains multiple classes (you may use "**balance_coco_dataset.py**" included in the train utilities folder to split and balance your dataset)

- Run the training by opening the terminal and "cd" into the directory of the "**train.py**"
- Next type:
    - **python train.py --data_path="path/to/your/train_data/" -dataset coco**
- **--data_path** is most likely the train_data folder on the same level of train.py; however, provide the absolute path to train_data to ensure the training script detects your data
- train.py uses arg parse, which means that you can define the number of epochs using the **--epochs** flag for instance; even if its recommended to leave the script in the "as-is" state, you can change the default arguments by modifying the "**get_args_parser()**" function.
- Set the batch size (the number of images to be processed) to match your GPU setup; **--batch-size 2** is recommended.
- To evaluate model performance the script will generate a "metrics" subfolder inside the run folder which contains JSON files for both training and validation losses and accuracy; you may want to use "**plot_losses_accuracy.py**" utility script to visualize them; you must point to the correct json paths inside your run directory/directories.