Wang, Jiaxin, Renninger, Heidi, & Ma, Qin. (2023). StoManager1: Automated, High-throughput Tool to Measure Leaf Stomata Using Convolutional Neural Networks (v.1.0.0). Zenodo. https://doi.org/10.5281/zenodo.7686022

StoManager1: Automated, High-throughput Tool to Measure Leaf Stomata Using Convolutional Neural Networks

What is StoManager1?

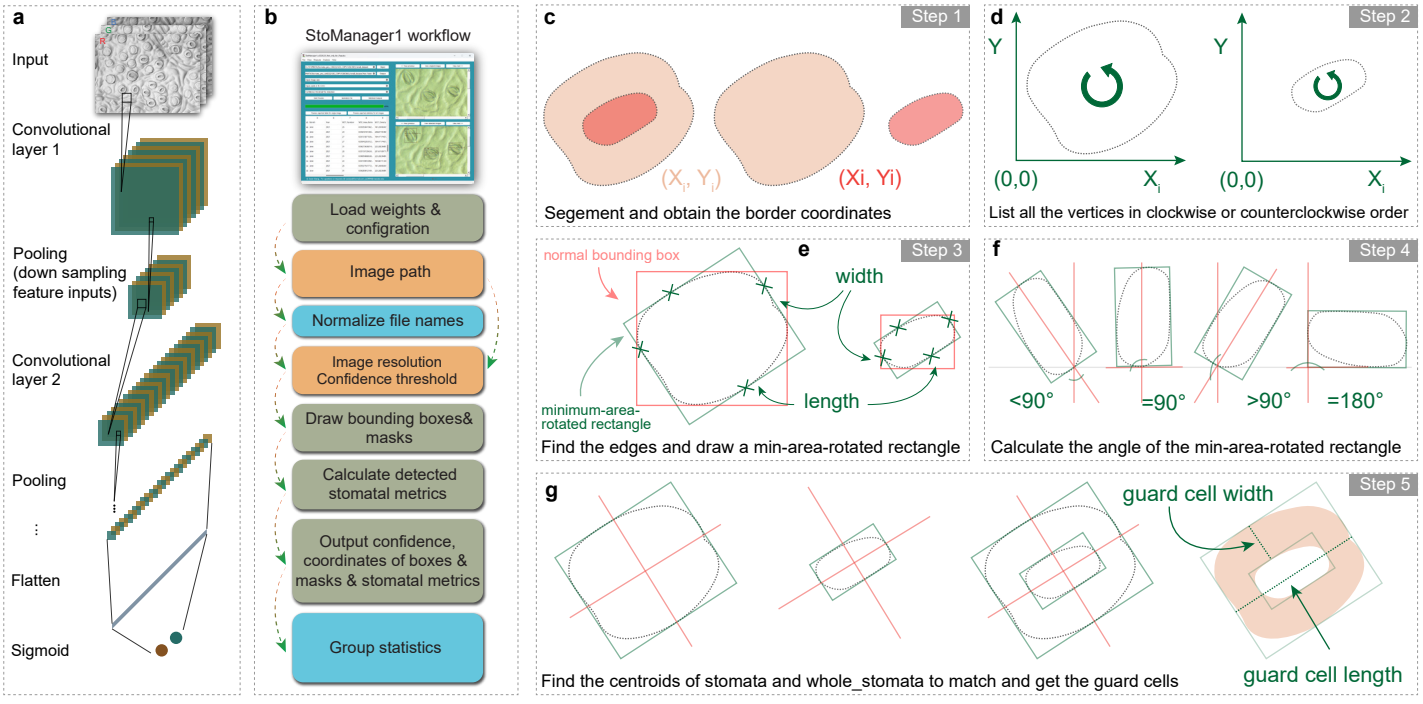
It is a tool created to automatically detect, count, measure, and analyze plant leaf stomatal metrics using geometrical, mathematical, and deep learning algorithms. Measured stomatal metrics include commonly used and newly developed, such as stomatal area, guard cell area, guard cell length, width, ratio of stomatal pore/guard cell area, and stomatal arrangement indices (aggregation, evenness, and divergence).

What can StoManager1 do?

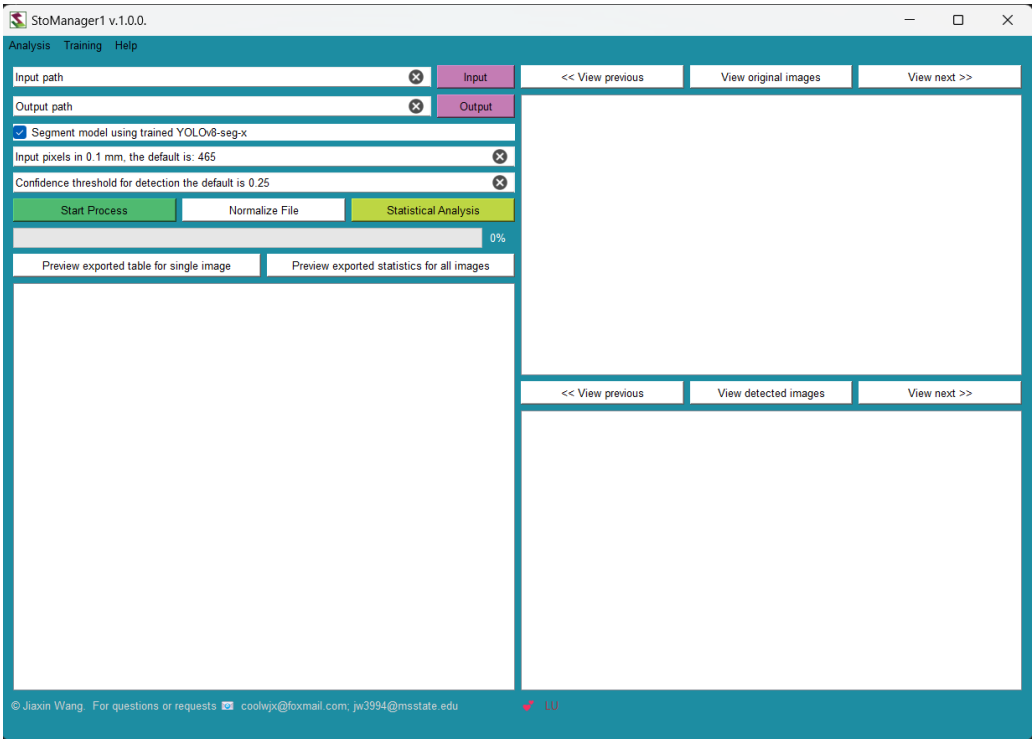
The main function of StoManager1 is for "stomata" and "whole_stomata" detection, segmentation, measurement, and analysis using a trained YOLOv8-seg-x model.
However, using empirical algorithms, StoManager1 also integrates bounding box-based YOLOv3 models for detection, counting, and stomatal metrics inference.
StoManager1 can be used as a pre-trained automatic stomatal labeling tool using its generated labels.
StoManager1 can be used as a YOLOv8 model trainer by using its integrated module-- model_training_in_app.exe.

How to use it?

StoManager1 can be used through Command line-GUI (https://github.com/JiaxinWang123/StoManager1) and a standalone Windows executable app (https://zenodo.org/doi/10.5281/zenodo.7686022).



Using StoManager1 for stomatal metrics measuring:

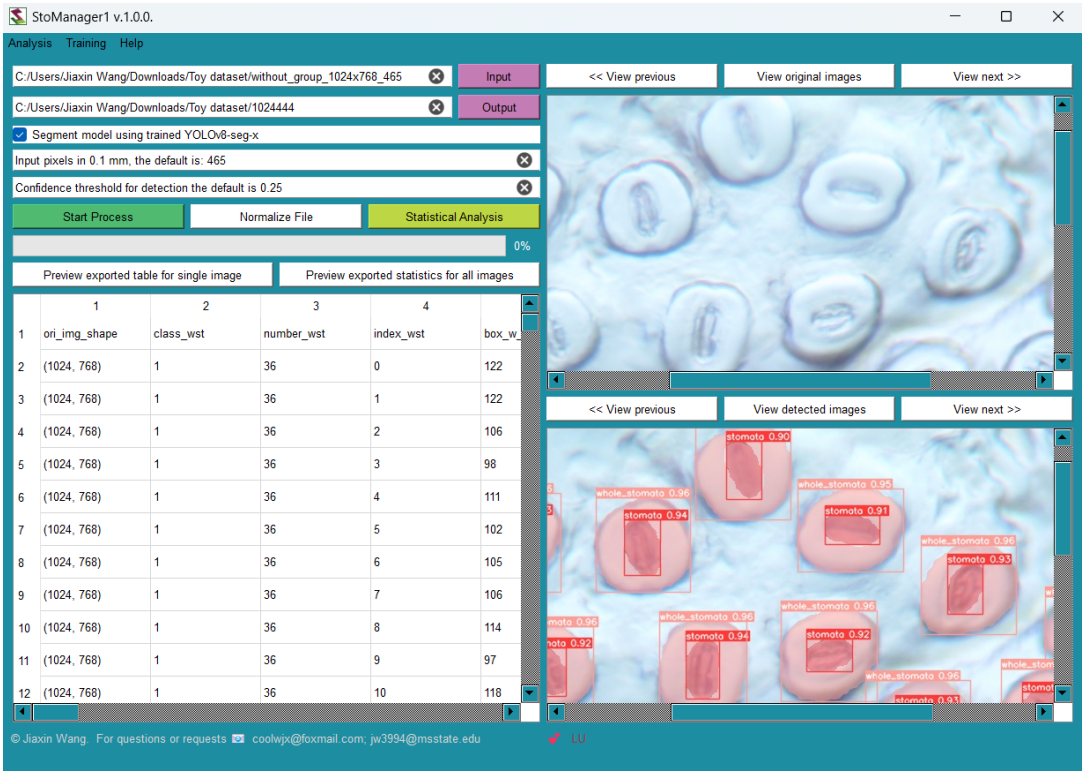


Step 1: Define the Input folder for images to be measured.
Step 2: Check or uncheck the YOLOv8-seg-x model. If checked, the YOLOv8-seg-x model will be used, and more stomatal metrics will be measured based on segmentation results. If unchecked, the bounding-box model and empirical algorithms will be used to estimate stomatal metrics (fewer stomatal metrics will be measured compared with the segment model).
Step 3: Define parameters for stomatal metrics measurement based on your image resolution and detection confidence threshold.
Step 4: Press the "Start Process" button to start the primary function—detection, segmentation, and measurement.
Step 5: Press the "Statistical Analysis" button to start the analysis.
Step 6: Preview your exported results.

Analyzed results will be saved to your output folder and subfolders, depending on which model you used. If you checked the YOLOv8-seg-x model, your analyzed results can be found in "YOUR_OUTPUT_PATH/Predict-output/Output_csv." Under the Output_csv folder, you will find a CSV file and folder named using your image file name.

If you are unchecked the YOLOv8-seg-x model, your analyzed results can be found in "YOUR_OUTPUT_PATH". Under YOUR_OUTPUT_PAT, you will find one CSV, JPG, and TXT file, all named using your image file name.

Supported image formats including 'jpg', 'png', 'tif', and 'jpeg'.



	1	2	3	4	
1	ori_img_shape	class_wst	number_wst	index_wst	box_w
2	(1024, 768)	1	36	0	122
3	(1024, 768)	1	36	1	122
4	(1024, 768)	1	36	2	106
5	(1024, 768)	1	36	3	98
6	(1024, 768)	1	36	4	111
7	(1024, 768)	1	36	5	102
8	(1024, 768)	1	36	6	105
9	(1024, 768)	1	36	7	106
10	(1024, 768)	1	36	8	114
11	(1024, 768)	1	36	9	97
12	(1024, 768)	1	36	10	118

Using StoManager1 for your own model training using your custom dataset:

You need to get your image dataset and corresponding object labels to train YOLO models. Some excellent image labeling tools are available for free, such as LabelImg (https://github.com/HumanSignal/labelImg) and Roboflow online platform (https://roboflow.com/), which you can use to label your image. I strongly recommend Roboflow since you can manage your image dataset easily, such as splitting training, validation, and testing datasets.

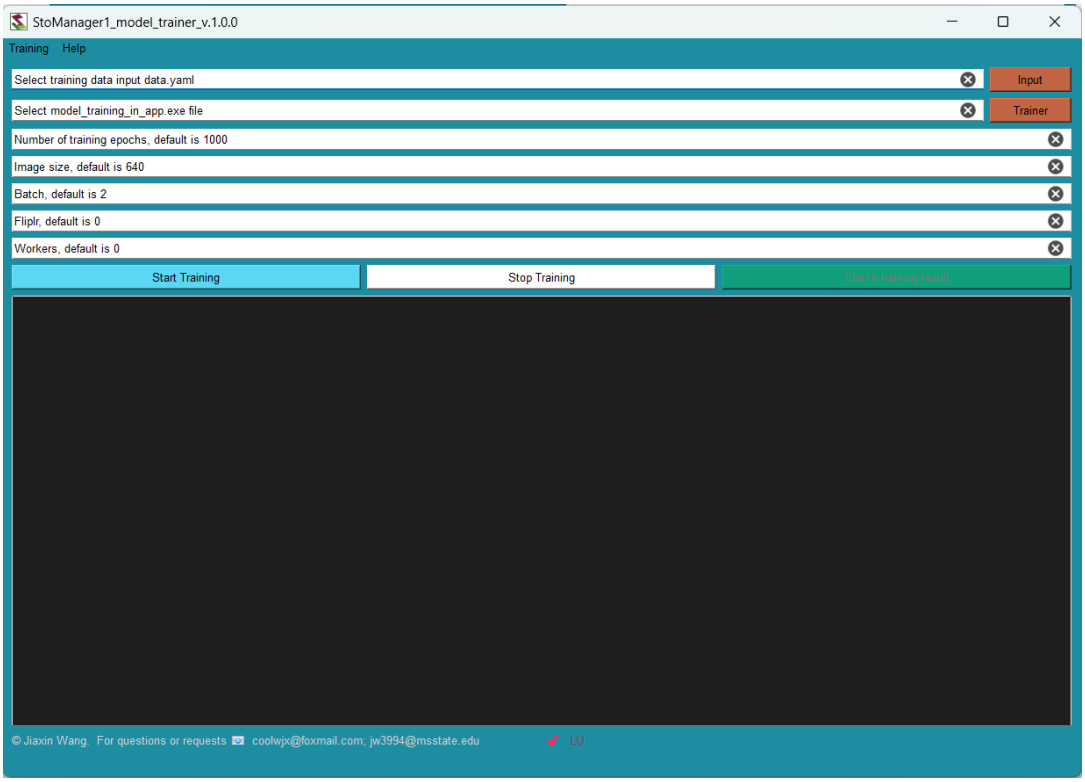
Once you get your labeled image dataset, you will create data.yaml file for dataloader. If you use Roboflow, it will automatically create data.yaml file. You might need to edit the relative path to the absolute path to make sure StoManager1 can find your dataset. For example:

Change:
train: ../train/images
val: ../valid/images
test: ../test/images

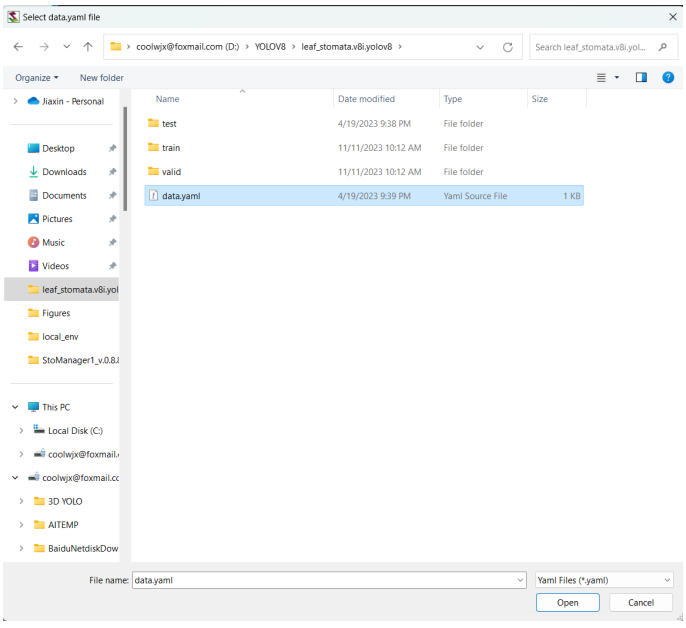
to:
train: D:/YOLOV8/leaf_stomata.v8i.yolov8/train/images
val: D:/YOLOV8/leaf_stomata.v8i.yolov8/valid/images
test: D:/YOLOV8/leaf_stomata.v8i.yolov8/test/images

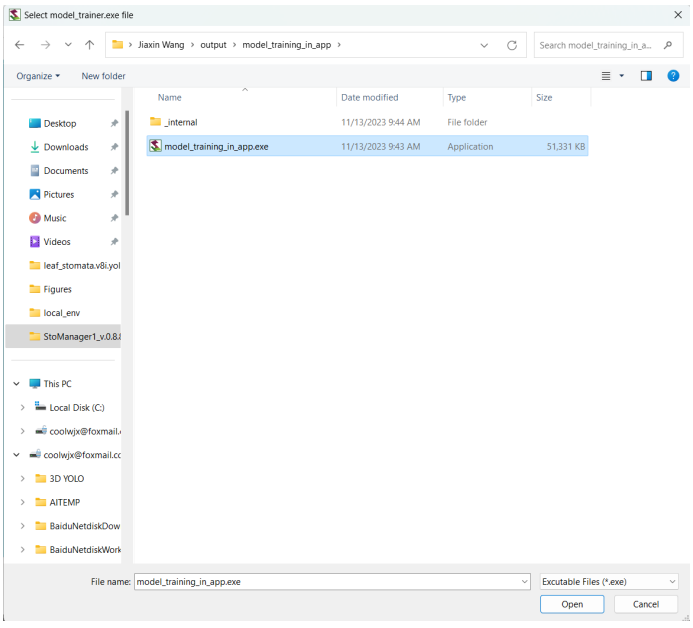
Now, you can start your own model!
To start the model trainer, click “Training” menu and “Train YOLOv8-seg-x” option in the main window.

Then you will see the training window like below:



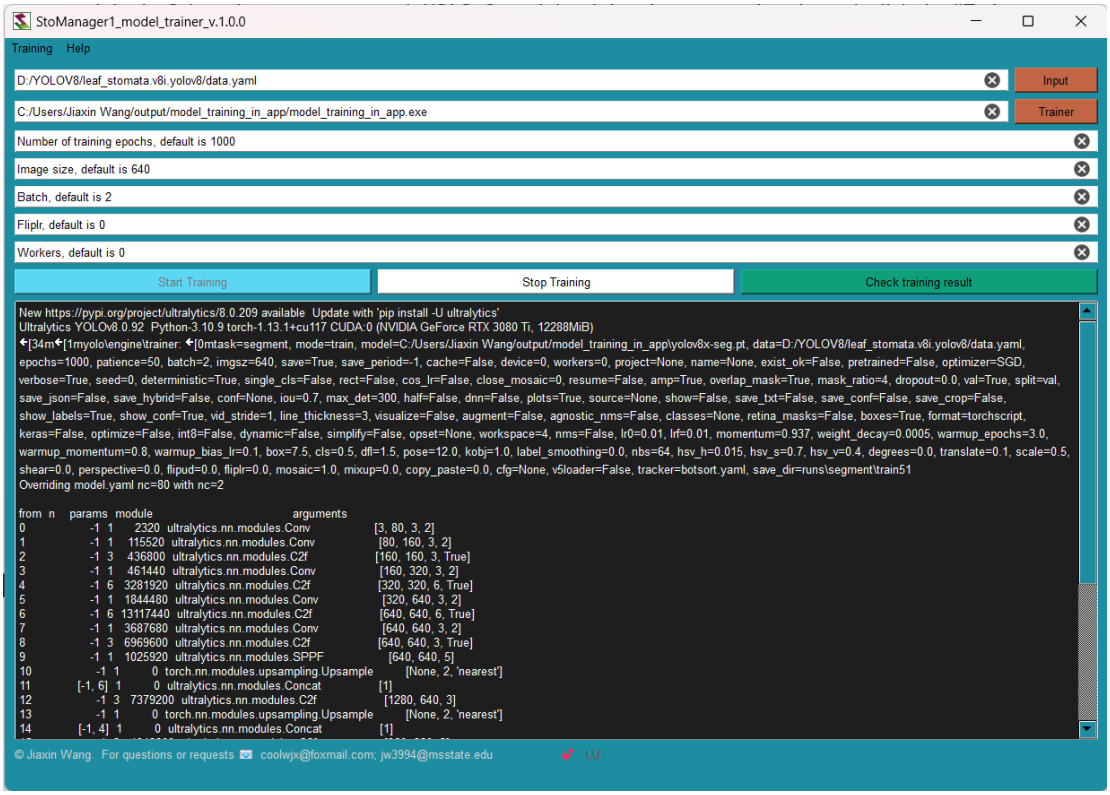
Then you can select your "data.yaml" file and "model_training_in_app.exe" file. Please note that "model_training_in_app.exe" file can be downloaded from the Zenodo page ((https://zenodo.org/doi/10.5281/zenodo.7686022), and **it must work with its _internal folder, which means you cannot delete _internal folder**.



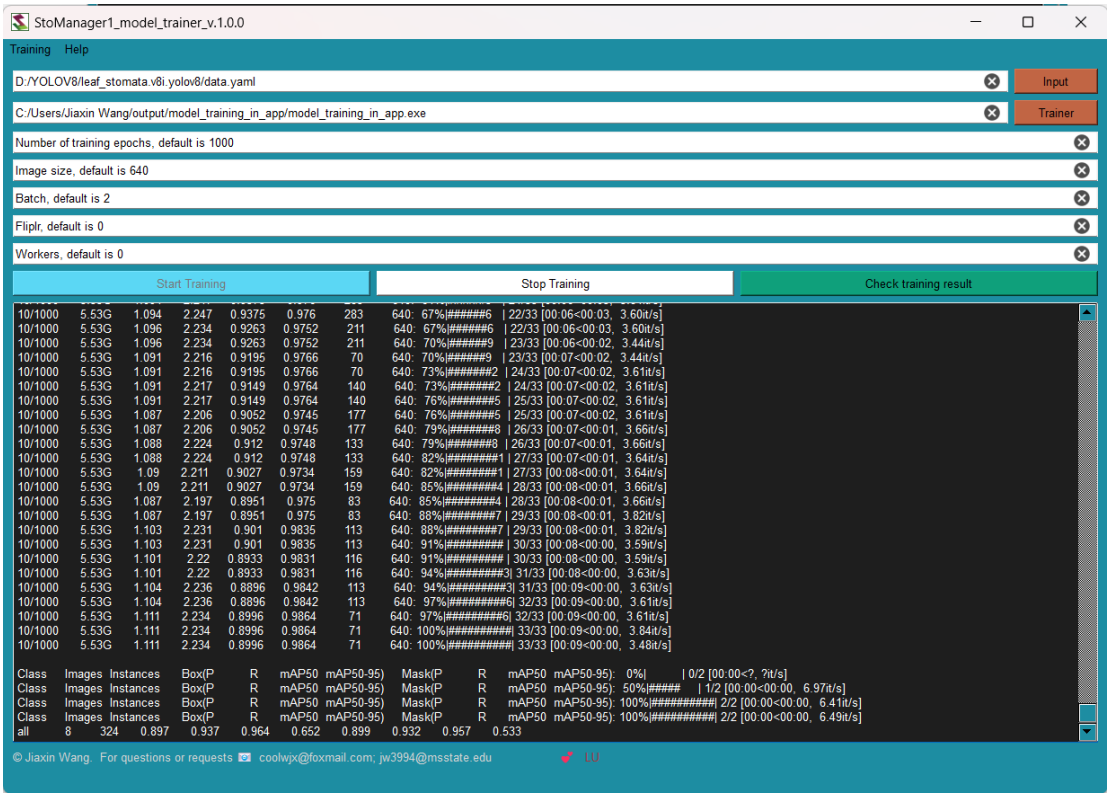


Then, you can define your training parameters. You can leave them as default if you don't know what to set. Otherwise, you can consult YOLOv8 model training documentation by clicking the "Train YOLOv8-seg-x model" option under the "Training" menu in the Trainer window. Note that batch size cannot be set too high if your computer does not have a powerful GPU or you are training a large image dataset with a single image with over 100 objects.

Once you set all parameters, you can press the “Start Training” button to start your training.



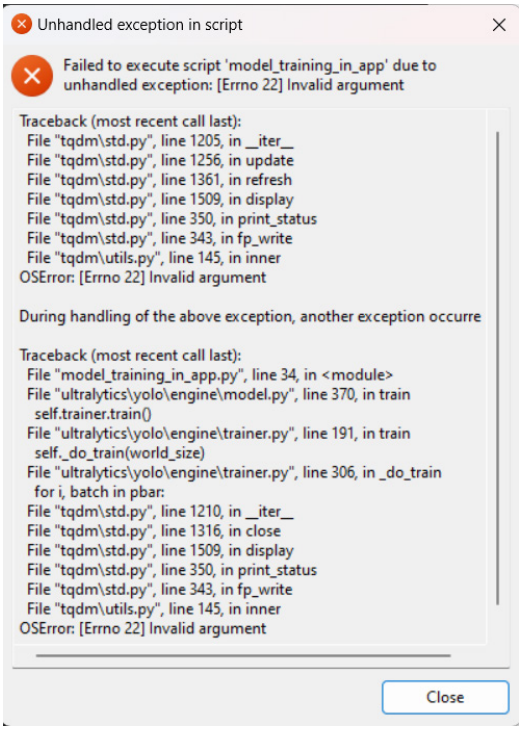
If it is your first training, you must have internet access to download pre-trained weights-- “YOLOv8-seg-x.pt”, or you can manually download and save it into where the “model_training_in_app.exe” file is located.



You can monitor the training process. The training will automatically stop once the best fitting model is obtained based on the precision, recall, mAP50-95, and/or training loss. You can find more information about the YOLOv8 model training metrics online.
Your training results will be saved in the folder of your StoManager1 app under the "runs/segment/train" subfolder, and you can find the training weights in the "weights" folder and other training and validation metrics plots in pdf and results data in CSV.

Name	Date modified	Type	Size
weights	11/13/2023 4:29 PM	File folder	
args.yaml	11/13/2023 4:25 PM	Yaml Source File	2 KB
BoxF1_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	18 KB
BoxP_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	18 KB
BoxPR_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	17 KB
BoxR_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	18 KB
confusion_matrix.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	19 KB
events.out.tfevents.1699914321.JiaxinHome.23408.0	11/14/2023 7:09 AM	0 File	24,645 KB
labels.pdf	11/13/2023 4:25 PM	Adobe Acrobat Docum...	87 KB
labels_correlogram.pdf	11/13/2023 4:25 PM	Adobe Acrobat Docum...	562 KB
MaskF1_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	18 KB
MaskP_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	18 KB
MaskPR_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	17 KB
MaskR_curve.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	17 KB
results.csv	11/14/2023 7:08 AM	Microsoft Excel Comma...	120 KB
results.pdf	11/14/2023 7:09 AM	Adobe Acrobat Docum...	92 KB
train_batch0.jpg	11/13/2023 4:25 PM	JPG File	162 KB
train_batch1.jpg	11/13/2023 4:25 PM	JPG File	253 KB
train_batch2.jpg	11/13/2023 4:25 PM	JPG File	196 KB
val_batch0_labels.jpg	11/14/2023 7:09 AM	JPG File	324 KB
val_batch0_pred.jpg	11/14/2023 7:09 AM	JPG File	284 KB
val_batch1_labels.jpg	11/14/2023 7:09 AM	JPG File	282 KB
val_batch1_pred.jpg	11/14/2023 7:09 AM	JPG File	211 KB
val_batch2_labels.jpg	11/14/2023 7:09 AM	JPG File	338 KB
val_batch2_pred.jpg	11/14/2023 7:09 AM	JPG File	229 KB

You may encounter a Traceback error like the one below (it's because "tqdm" is not working correctly under subprocess; it's function is to estimate the training time for each epoch), and you can ignore it since it will not affect your training process and results.



Using StoManager1 as an automated labeling tool.

Once you have done your measurement, you can use the following code to extract the label files and rename them as same with the image files.

```
import os
import os.path

path =  "C:\Users\YOUR_PATH\F1_training_data\Output\Predict_output\Output_csv"
for dirpath, dirnames, filenames in os.walk(path):
    for filename in [f for f in filenames if f.endswith(".txt")]:
        labelFile = os.path.join(dirpath, filename)
        dir = os.path.dirname(os.path.dirname(labelFile)) ## dir of dir of file
        ## once you're at the directory level you want, with the desired directory as the final path node:
        dirname1 = os.path.basename(dir)
        os.rename(labelFile, f"{dirname1}.txt")
```

Your labels will be saved in your "Output_csv" path, and you can then import those labels and corresponding images into Roboflow for label quality check and modification as needed. This process is beneficial and can save you a lot of time if you have particular stomatal images and need to label a large dataset.

StoManager1 can also be used for other object detection (plants, animals, and objects) when training your own model.

I hope you will find StoManager1 helpful for your research, and please let me know if you have any questions or requests regarding StoManager1. ❤️