1

Mar 18, 2022

RasberryPi-computer based phenotyping for side view image process V.1

Li'ang Yu[1], Magdalena M Julkowska[1]

[1]Boyce Thompson Institute

1

dx.doi.org/10.17504/protocols.io.eq2lynp7pvx9/v1

Stress Architecture and RNA Biology Lab BTI
Tech. support phone: **+1 (607) 279-6002** email: **mmj55@cornell.edu**

Li'ang Yu
Boyce Thompson Institute , University of Illinois at Urbana-...

Imaging of plants can be done in an inexpensive way, using Raspberry Pi based setup. Here - we describe how we are processing the side-view images taken from 7 different sides - using the PlantCV pipeline.

DOI

dx.doi.org/10.17504/protocols.io.eq2lynp7pvx9/v1

Li'ang Yu, Magdalena M Julkowska 2022. RasberryPi-computer based phenotyping for side view image process. **protocols.io** https://dx.doi.org/10.17504/protocols.io.eq2lynp7pvx9/v1

side-view, plant imaging, plant growth rate, plantcv, batch processing

protocol ,

Jan 19, 2022

Mar 18, 2022

57147

Data preparation

1 Download the data from RaspberryPi computer, where the images are being saved into one folder. If you have multiple stands for side-view image collection - download all the images into one folder entitled as your experiment.

NOTE - please make sure that the RapberryPi's were recording the timestamp correctly (i.e. were continuously connected to the internet router - that ensures the right time and date being timestamped on each image).

If your RaspberryPi computer was not connected to the internet throughout the experiment - it is better to save each day of imaging in a separate folder. It will be slightly more work to analyze them but will ensure that your timestamp is correct.

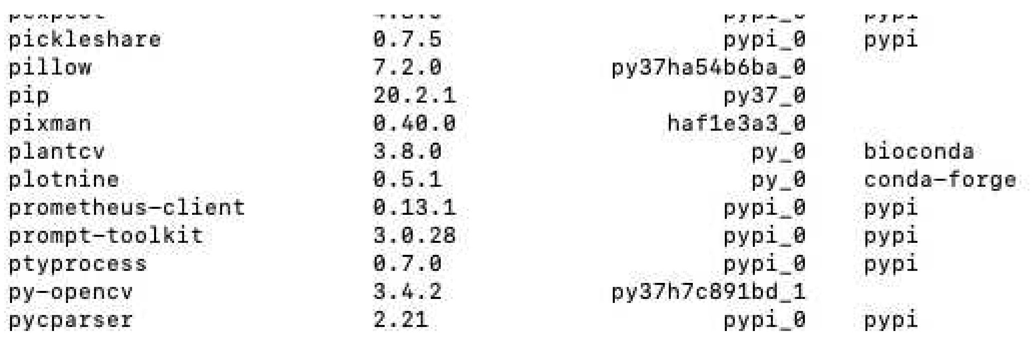
Script development

2 On your desktop/laptop - make sure to have Conda, Jupyter Notebook and PlantCV installed, including all the dependencies. Please see the links for the installation instruction, and follow the instructions by installations through Conda.

IF you are not sure about whether conda, jupyter or plantcv are already installed at your computer - you can check it through typing the following command in your terminal:

```
# to check if conda is installed - and what packages are installed:
conda list
# to check Jupyter Notebook version installed:
jupyter --version
```

To check PlantCV version - search through the list of packages that were delivered to you when prompting conda list. PlantCV should be one of the packages listed alphabetically therein - similar to the screenshot below:



```
pickleshare               0.7.5                    pypi_0    pypi
pillow                    7.2.0            py37ha54b6ba_0
pip                       20.2.1                   py37_0
pixman                    0.40.0               haf1e3a3_0
plantcv                   3.8.0                      py_0    bioconda
plotnine                  0.5.1                      py_0    conda-forge
prometheus-client         0.13.1                   pypi_0    pypi
prompt-toolkit            3.0.28                   pypi_0    pypi
ptyprocess                0.7.0                    pypi_0    pypi
py-opencv                 3.4.2            py37h7c891bd_1
pycparser                 2.21                     pypi_0    pypi
```

List of packages installed in my Conda environment

3 Make sure to download this example Jupyter Notebook - which has been optimized for cowpea/tepary/tomato imaging using the PhenoCage at BTI, and save it in your current working directory.

20220307 PlantCV tepary.ipynb

Please also see pdf version of the same notebook of how individual steps SHOULD look like

20220307 PlantCV tepary.pdf

Just as a reminder - to know what is your current working directory use command

```
pwd
```

whereas to change the directory use the command

```
cd
```

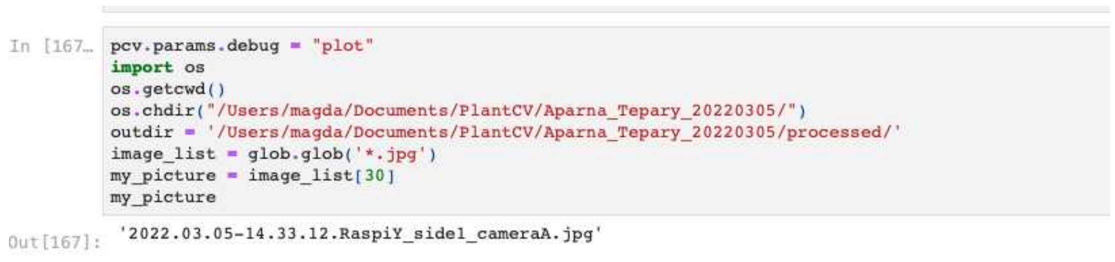
to go one directory up (out of your current folder) use:

```
cd ../
```

4 Open the notebook by typing two commands into your terminal window

```
conda activate plantcv
jupyter notebook
```

5 Navigate to the folder that contains example jupyter notebook - and change theimage directory in the 3rd chunk of the notebook. This is how it looks like:

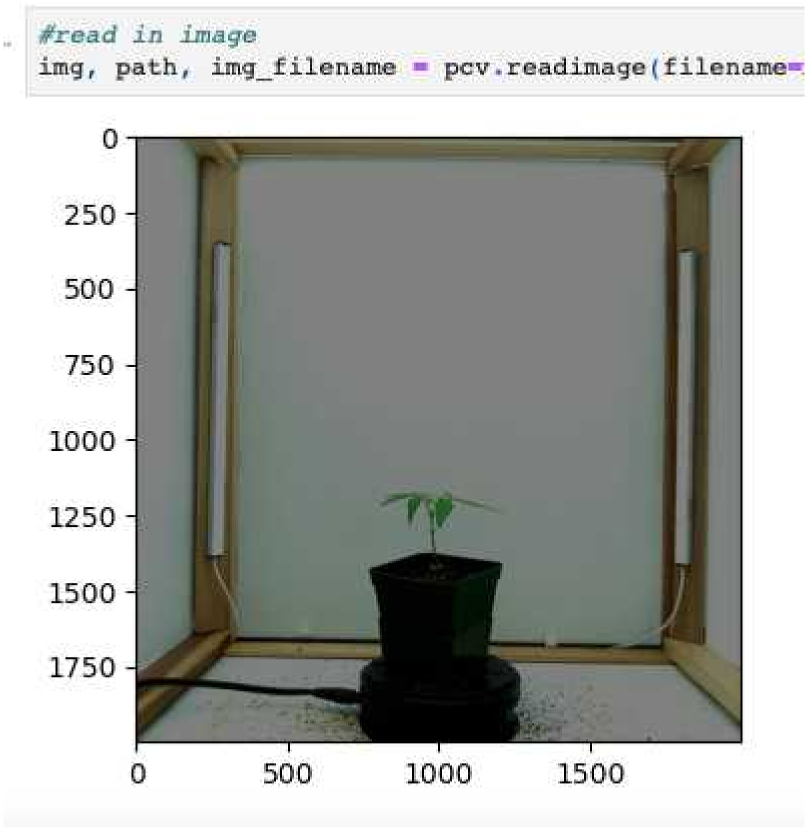


```
In [167…  pcv.params.debug = "plot"
          import os
          os.getcwd()
          os.chdir("/Users/magda/Documents/PlantCV/Aparna_Tepary_20220305/")
          outdir = '/Users/magda/Documents/PlantCV/Aparna_Tepary_20220305/processed/'
          image_list = glob.glob('*.jpg')
          my_picture = image_list[30]
          my_picture

Out[167]:  '2022.03.05-14.33.12.RaspiY_sidel_cameraA.jpg'
```

jupyter notebook chunk with working directory - containing all the images made by Raspberry Pi computer

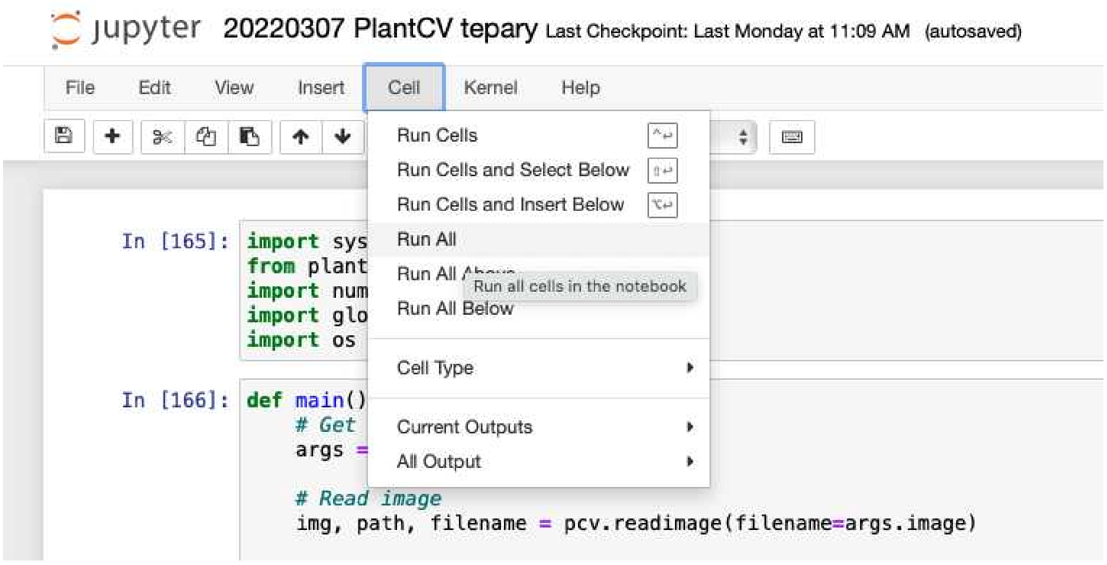
6 Make sure that you load one example image, and that it shows up in 4th chunk of Jupyter notebook. It should look like on the image below, but with your plant.



```
#read in image
img, path, img_filename = pcv.readimage(filename=
```

loaded image in Jupyter notebook

7 Once the image is loaded - you can try and run the entire notebook for your image - to see if the default parameters are working for your imaging setup. You can run the entire notebook by choosing "run all" from the Cell menu in Jupyter Notebook - as on the screenshot below:



jupyter 20220307 PlantCV tepary Last Checkpoint: Last Monday at 11:09 AM (autosaved)

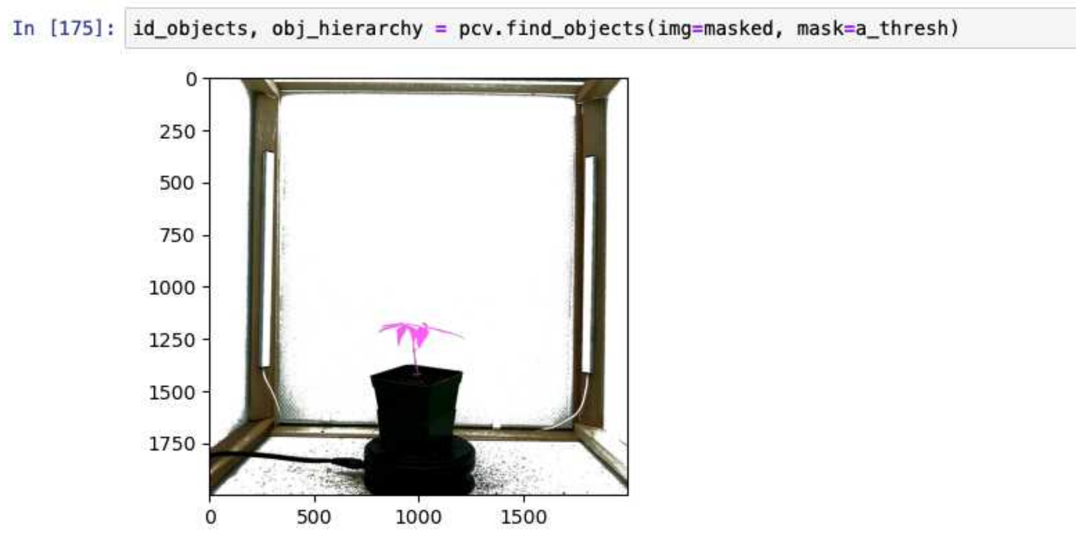
```
In [165]: import sys
          from plant
          import num
          import glo
          import os

In [166]: def main()
              # Get
              args =

              # Read image
              img, path, filename = pcv.readimage(filename=args.image)
```

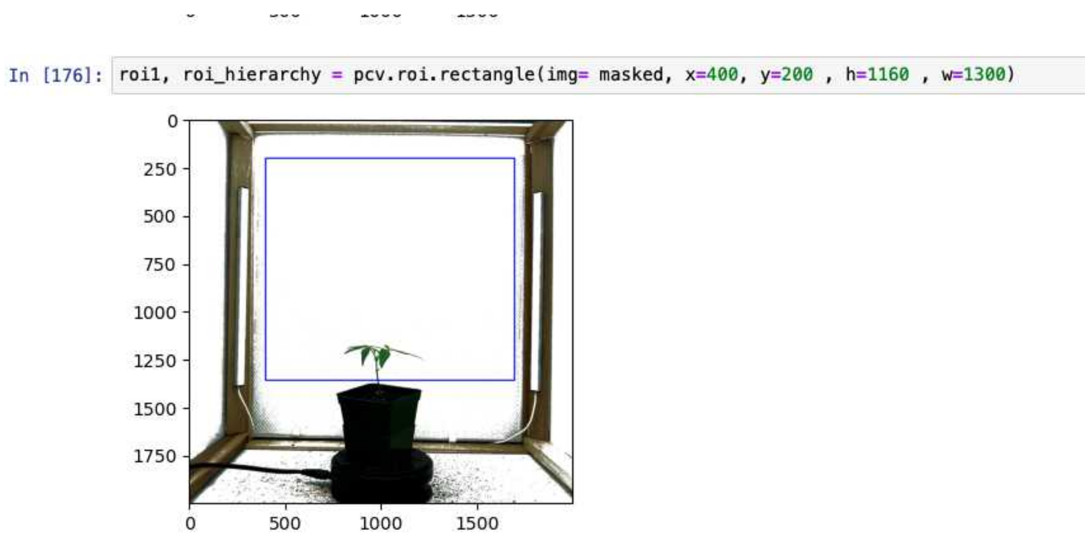
run all the commands within the notebook

8 After you have run throughout the jupyter notebook - please make sure that:
1) the MASK is covering all of the plant parts (chunk #11) - i.e. all of your plant bits are covered with the PINK mask
IF that is not the case - please adjust the threshold values in v_thresh and a_thresh channels
(chunks #6 and #8 respectively)



```
In [175]: id_objects, obj_hierarchy = pcv.find_objects(img=masked, mask=a_thresh)
```

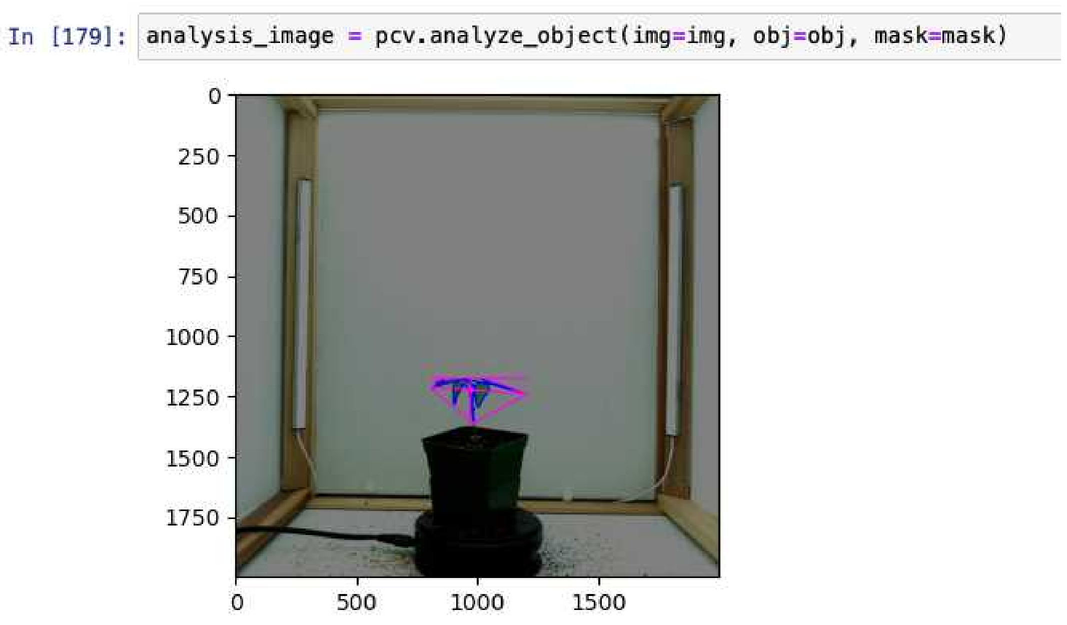
2) if the ROI (Region of Interest) is at least partially covering your plant (chunk #12)



```
In [176]: roi1, roi_hierarchy = pcv.roi.rectangle(img= masked, x=400, y=200 , h=1160 , w=1300)
```

3) if there are any other elements of the picture included in the final measurement - in the last chunk.
It is possible that some stems are going to be too thin to be recognized by the PlantCV. For

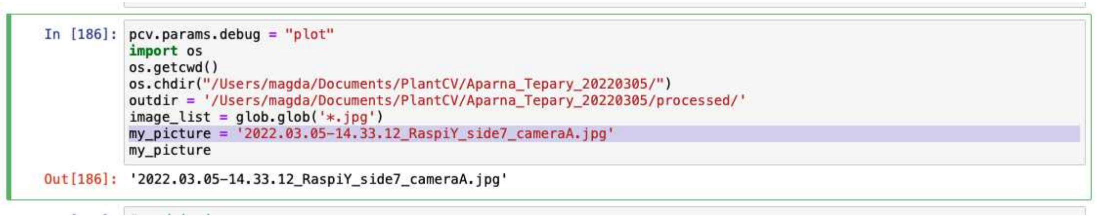
our purpose - estimating the biomass from 7 different side-angles - as the major plant parts are included in the measurements - we should be good to go.



```
In [179]: analysis_image = pcv.analyze_object(img=img, obj=obj, mask=mask)
```

9 Re-run the pipeline on afew example images from your dataset - including young and old plants - and plants that differ in their coloring (e.g. lime green vs. dark green) - by selecting different images in chunk #3.

Please note that you can also use a file name as "my_picture" to select the files that you want to use for testing:



```
In [186]: pcv.params.debug = "plot"
          import os
          os.getcwd()
          os.chdir("/Users/magda/Documents/PlantCV/Aparna_Tepary_20220305/")
          outdir = '/Users/magda/Documents/PlantCV/Aparna_Tepary_20220305/processed/'
          image_list = glob.glob('*.jpg')
          my_picture = '2022.03.05-14.33.12_RaspiY_side7_cameraA.jpg'
          my_picture

Out[186]: '2022.03.05-14.33.12_RaspiY_side7_cameraA.jpg'
```

10 Once you have the correct thresholds - please adjust them in the python code - for which the example is attached below. This is the code that you will be used for batch processing of all images.

NOTE: please adjust the python code only using the code editing software - such as Atom - to make sure that the editing software won't replace your symbols, yielding the code non-functional.

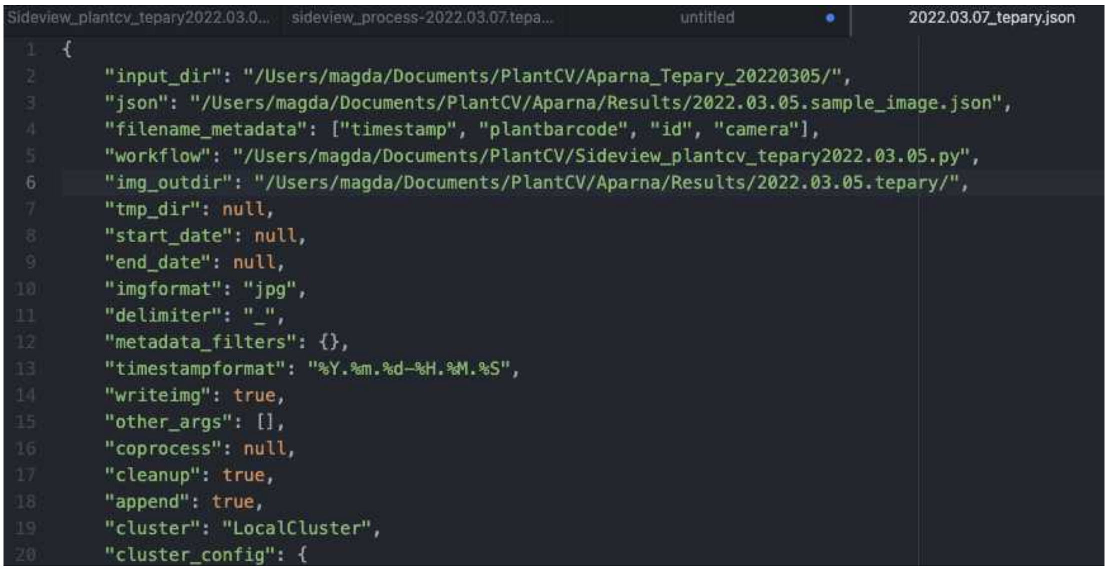
Sideview_plantcv_tepary2022.03.05.py

Batch processing of the side images

11 For processing the images - we need to prepare a json file that will contain the information on:
- the path to the repository containing ALL of your images - indicated as **"input_dir"** in the 2nd line of the json file
- the path to the json output file - indicated as **"json"** in the 3rd line of the json file
- the way that your images are named - the order of timestamp - plant barcode - id - camera - indicated as **"filename_metadata"** in 4th line of the json file
- the path to the python code to be used - indicated as **"workflow"** in the 5th line of the json file
- the path to the repository where all the PROCESSED results will be deposited - indicated as "img_outdir" in the 6th
- delimiter between the individual elements of the file name - we usually use underscore - indicated as **"delimiter"** in the 11th line of the json file
- the timestamp format - indicating the general organization of your time-metadata in the file name - indicated as **"timestampformat"** in line 13 of the json file

NOTE: Please adjust json file ONLY in the code-processing software - such as Atom. Otherwise, you are running a risk that the individual symbols are going to be changed by your text editing software - and damage the json file.

NOTE: the output directories should exist PRIOR to going on with the next steps. Please make sure to generate the correct folders.


Sideview_plantcv_tepary2022.03.0...
sideview_process-2022.03.07.tepa...
untitled
2022.03.07_tepary.json

```
1  {
2      "input_dir": "/Users/magda/Documents/PlantCV/Aparna_Tepary_20220305/",
3      "json": "/Users/magda/Documents/PlantCV/Aparna/Results/2022.03.05.sample_image.json",
4      "filename_metadata": ["timestamp", "plantbarcode", "id", "camera"],
5      "workflow": "/Users/magda/Documents/PlantCV/Sideview_plantcv_tepary2022.03.05.py",
6      "img_outdir": "/Users/magda/Documents/PlantCV/Aparna/Results/2022.03.05.tepary/",
7      "tmp_dir": null,
8      "start_date": null,
9      "end_date": null,
10     "imgformat": "jpg",
11     "delimiter": "_",
12     "metadata_filters": {},
13     "timestampformat": "%Y.%m.%d-%H.%M.%S",
14     "writeimg": true,
15     "other_args": [],
16     "coprocess": null,
17     "cleanup": true,
18     "append": true,
19     "cluster": "LocalCluster",
20     "cluster_config": {
```

The part of the json file that needs to be adjusted to individual experiments

The example json file is attached below

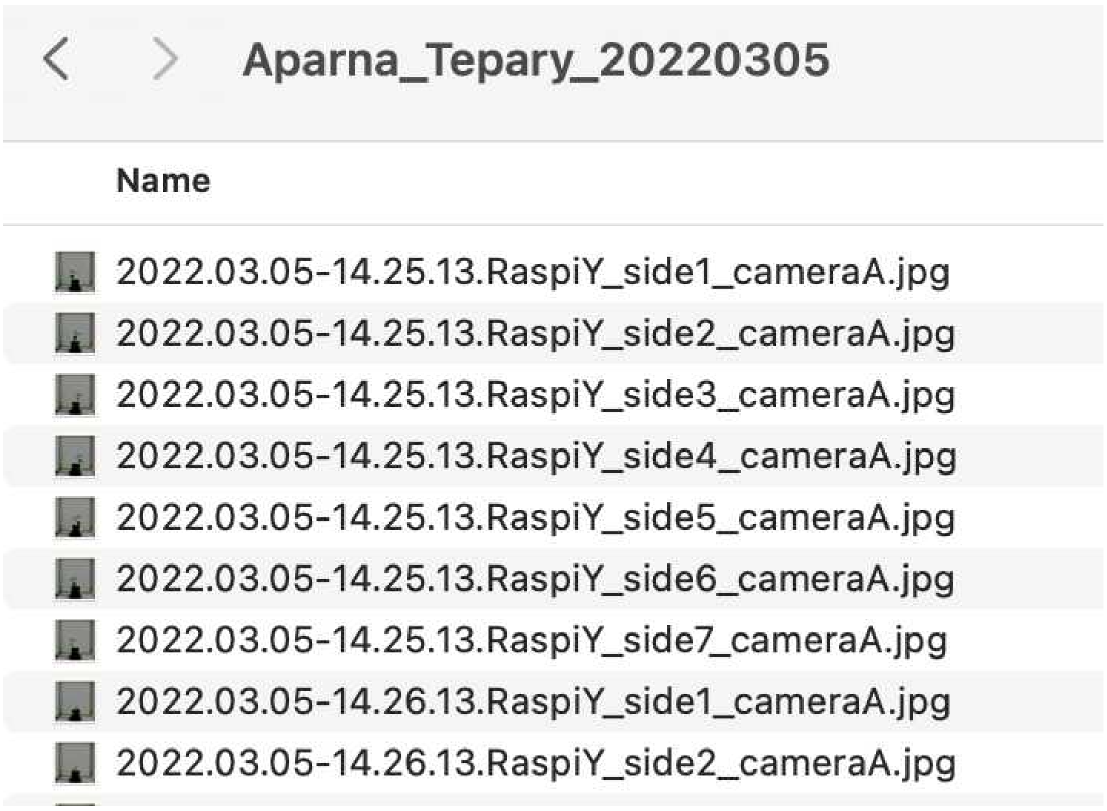
2022.03.07_tepary.json

12 PROBLEMS with file naming? - Check out our SOLUTION here. NOTE - this step is OPTIONAL - and only applicable if you have oddly-named files:

Recently - we were recording the images with the error in their names - i.e. the timestamp was separated by a "." but the Raspberry Pi was also separated by a ".", instead of an "_". If something similar happens to you - that you have to rename your files prior to batch processing - follow this step:

1) determine the UNIQUE part of your name that you would like to change - such as ".Raspi" - rather than just "." or "_" -

In our case - we had to rename ".RaspiY" to "_RaspiY" for correct delimiter between "timestamp" and "plantbarcode" - like on the screenshot below:


Aparna_Tepary_20220305

Name

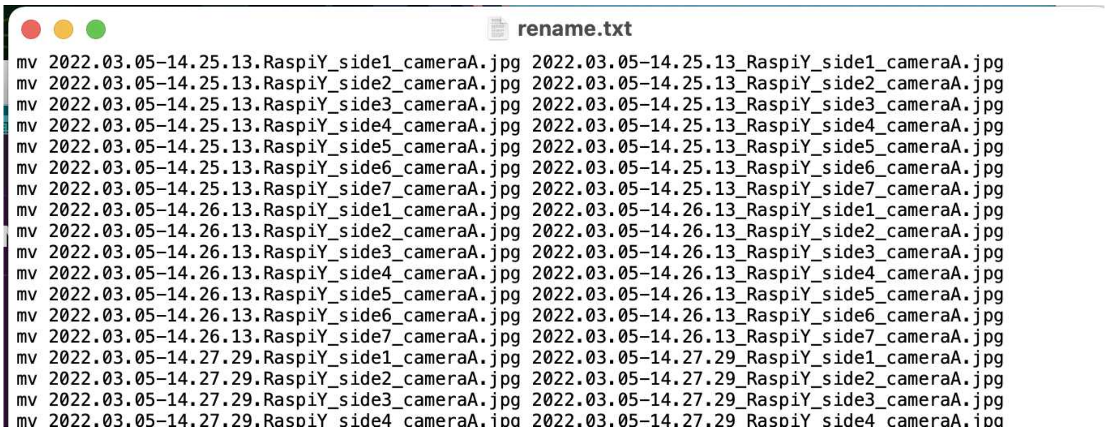
2022.03.05-14.25.13.RaspiY_side1_cameraA.jpg
2022.03.05-14.25.13.RaspiY_side2_cameraA.jpg
2022.03.05-14.25.13.RaspiY_side3_cameraA.jpg
2022.03.05-14.25.13.RaspiY_side4_cameraA.jpg
2022.03.05-14.25.13.RaspiY_side5_cameraA.jpg
2022.03.05-14.25.13.RaspiY_side6_cameraA.jpg
2022.03.05-14.25.13.RaspiY_side7_cameraA.jpg
2022.03.05-14.26.13.RaspiY_side1_cameraA.jpg
2022.03.05-14.26.13.RaspiY_side2_cameraA.jpg

wrongly named images

2) in the terminal - navigate to your directory containing the wrongly named images - and run the following command line - which will take ALL images containing in their name "RaspiY", and replace a specific part of the file name indicated in the "sed" part of the command. This code will generate a file called "rename.txt" that will be saved in the same folder as your original images.

```
ls -1 *RaspiY* | awk '{print("mv "$1 " " $1)}' | sed
's/.RaspiY/_RaspiY/2' > rename.txt
```

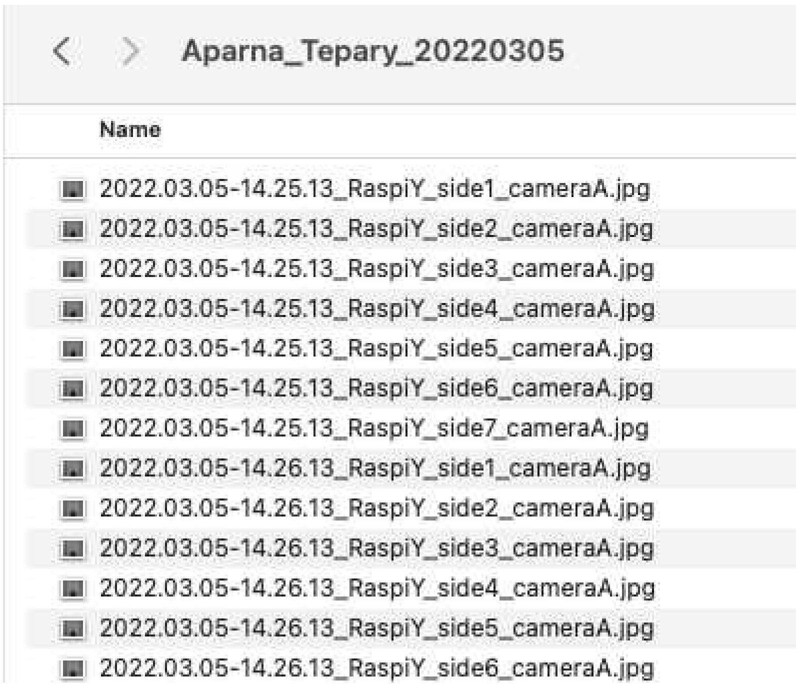
3) open the rename file which should contain ALL the "move" commands (mv) for all the files within your directory with your wrongly named images. Should look something like the screenshot below:


rename.txt

```
mv 2022.03.05-14.25.13.RaspiY_side1_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side1_cameraA.jpg
mv 2022.03.05-14.25.13.RaspiY_side2_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side2_cameraA.jpg
mv 2022.03.05-14.25.13.RaspiY_side3_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side3_cameraA.jpg
mv 2022.03.05-14.25.13.RaspiY_side4_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side4_cameraA.jpg
mv 2022.03.05-14.25.13.RaspiY_side5_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side5_cameraA.jpg
mv 2022.03.05-14.25.13.RaspiY_side6_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side6_cameraA.jpg
mv 2022.03.05-14.25.13.RaspiY_side7_cameraA.jpg 2022.03.05-14.25.13_RaspiY_side7_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side1_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side1_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side2_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side2_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side3_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side3_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side4_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side4_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side5_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side5_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side6_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side6_cameraA.jpg
mv 2022.03.05-14.26.13.RaspiY_side7_cameraA.jpg 2022.03.05-14.26.13_RaspiY_side7_cameraA.jpg
mv 2022.03.05-14.27.29.RaspiY_side1_cameraA.jpg 2022.03.05-14.27.29_RaspiY_side1_cameraA.jpg
mv 2022.03.05-14.27.29.RaspiY_side2_cameraA.jpg 2022.03.05-14.27.29_RaspiY_side2_cameraA.jpg
mv 2022.03.05-14.27.29.RaspiY_side3_cameraA.jpg 2022.03.05-14.27.29_RaspiY_side3_cameraA.jpg
mv 2022.03.05-14.27.29.RaspiY_side4_cameraA.jpg 2022.03.05-14.27.29_RaspiY_side4_cameraA.jpg
```

4) copy ALL the contents of the rename.txt file - and paste them into your terminal window. This will run ALL the rename commands - that will replace the plant name in your repository with the wrongly made images.

5) check the repository if the names are indeed changed to the correct names.


Aparna_Tepary_20220305

```
Name
2022.03.05-14.25.13_RaspiY_side1_cameraA.jpg
2022.03.05-14.25.13_RaspiY_side2_cameraA.jpg
2022.03.05-14.25.13_RaspiY_side3_cameraA.jpg
2022.03.05-14.25.13_RaspiY_side4_cameraA.jpg
2022.03.05-14.25.13_RaspiY_side5_cameraA.jpg
2022.03.05-14.25.13_RaspiY_side6_cameraA.jpg
2022.03.05-14.25.13_RaspiY_side7_cameraA.jpg
2022.03.05-14.26.13_RaspiY_side1_cameraA.jpg
2022.03.05-14.26.13_RaspiY_side2_cameraA.jpg
2022.03.05-14.26.13_RaspiY_side3_cameraA.jpg
2022.03.05-14.26.13_RaspiY_side4_cameraA.jpg
2022.03.05-14.26.13_RaspiY_side5_cameraA.jpg
2022.03.05-14.26.13_RaspiY_side6_cameraA.jpg
```

13 Now - you have prepared ALL the files - make sure to also have plantcv-workflow.py and plantcv-utils.py codes in your directory - it does NOT have to be the same directory as your experiment/images. Just have them somewhere with the identifiable path to them

plantcv-utils.py **plantcv-workflow.py**

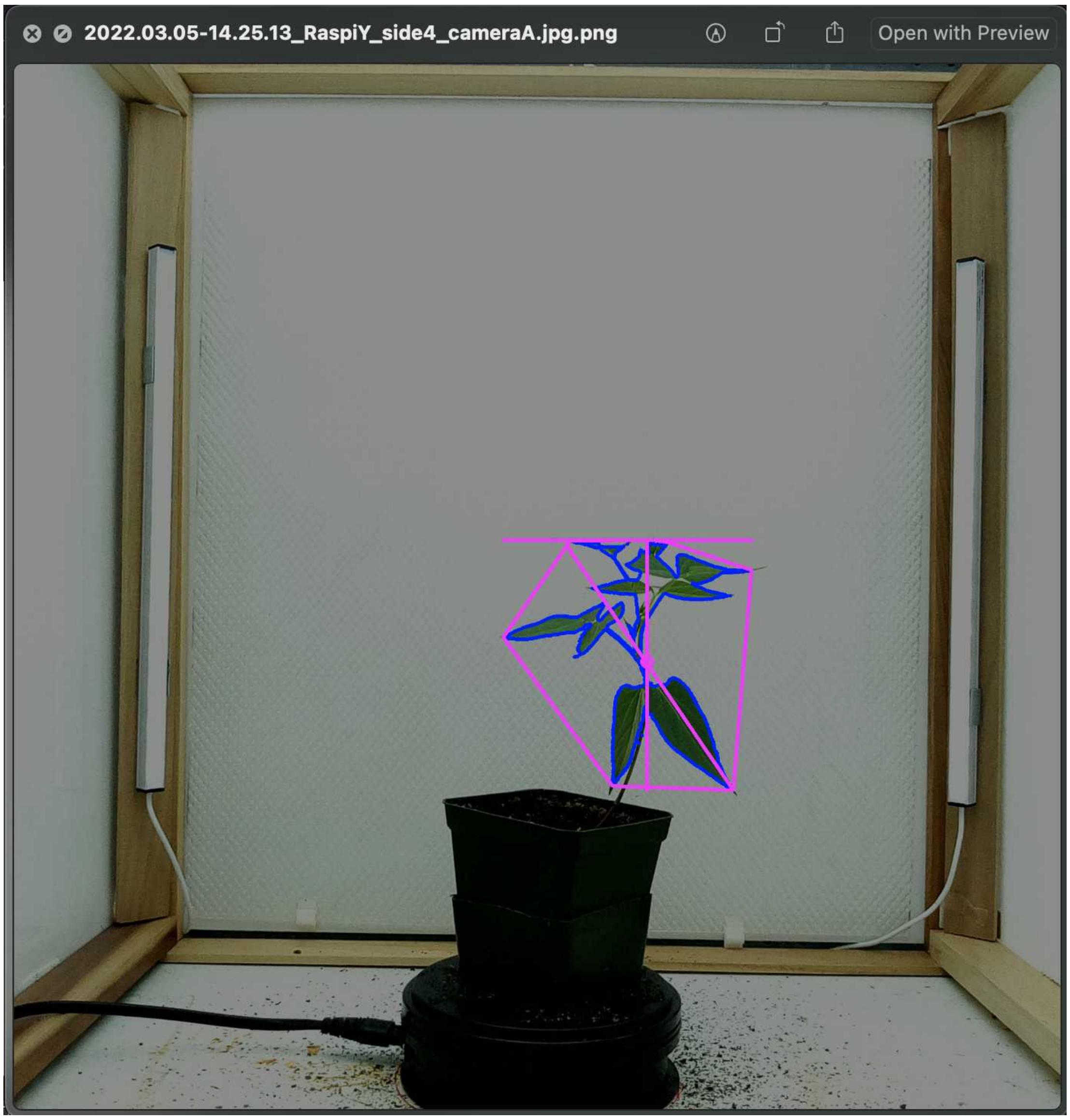
14 Run the following command - adjusting the path to your **"plantcv-workflow.py"** file and path to your "json" file prepared in step 11 of THIS protocol.

```
python /Users/magda/Documents/PlantCV/plantcv-workflow.py --config
/Users/magda/Documents/PlantCV/2022.03.07_tepary.json
```

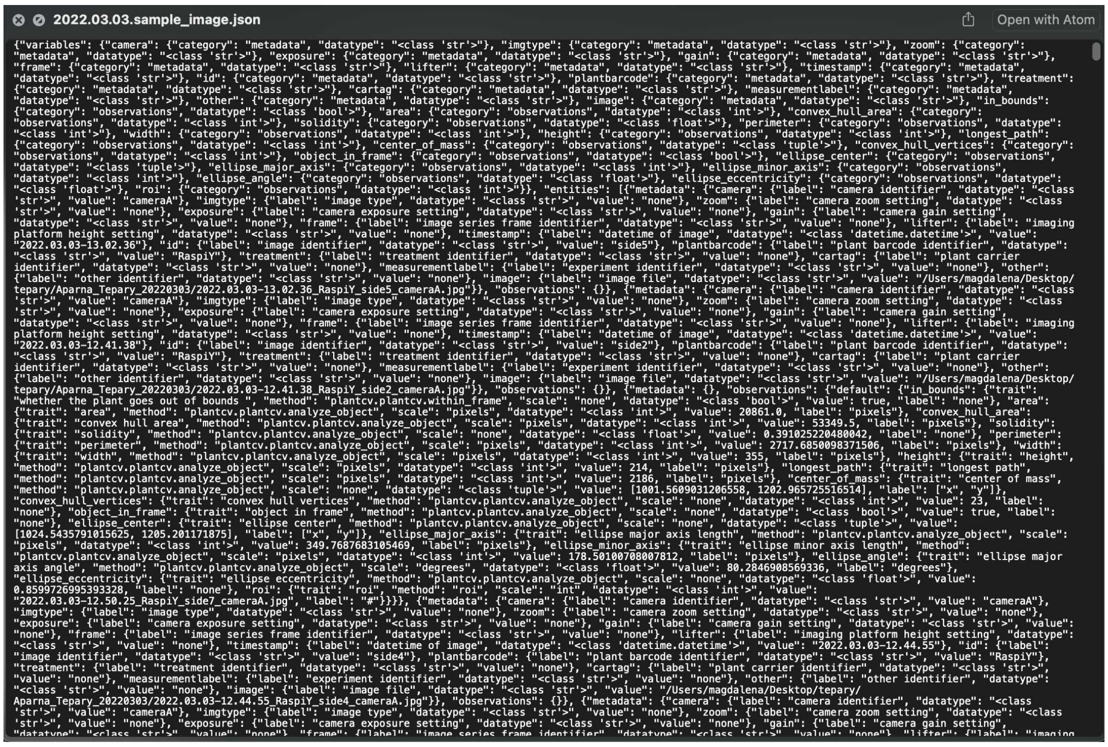
NOTE: if you are running it on your Desktop - this step might take some time (it took me +/- 20 minutes to process 980 images of cowpea)

15 Once the script is finished working - you should see the output files in the output directory specified in the json file generated in Step 11 of THIS protocol.

Please take time to go through the output files and make sure to go through the processed images - possibly identifying any problems that the python code had with identifying the plant parts in your images.



16 The script will also create a json file with the results in the output folder. It will look something like the screenshot below:



```
{"variables": {"camera": {"category": "metadata", "datatype": "<class 'str'>"}, "imgtype": {"category": "metadata", "datatype": "<class 'str'>"}, "zoom": {"category":
"metadata", "datatype": "<class 'str'>"}, "exposure": {"category": "metadata", "datatype": "<class 'str'>"}, "gain": {"category": "metadata", "datatype": "<class 'str'>"},
"frame": {"category": "metadata", "datatype": "<class 'str'>"}, "lifter": {"category": "metadata", "datatype": "<class 'str'>"}, "timestamp": {"category": "metadata",
"datatype": "<class 'str'>"}, "id": {"category": "metadata", "datatype": "<class 'str'>"}, "plantbarcode": {"category": "metadata", "datatype": "<class 'str'>"}, "treatment":
{"category": "metadata", "datatype": "<class 'str'>"}, "cartag": {"category": "metadata", "datatype": "<class 'str'>"}, "measurementlabel": {"category": "metadata",
"datatype": "<class 'str'>"}, "other": {"category": "metadata", "datatype": "<class 'str'>"}, "image": {"category": "metadata", "datatype": "<class 'str'>"}, "in_bounds":
{"category": "observations", "datatype": "<class 'bool'>"}, "area": {"category": "observations", "datatype": "<class 'int'>"}, "convex_hull_area": {"category":
"observations", "datatype": "<class 'int'>"}, "solidity": {"category": "observations", "datatype": "<class 'float'>"}, "perimeter": {"category": "observations", "datatype":
"<class 'int'>"}, "width": {"category": "observations", "datatype": "<class 'int'>"}, "height": {"category": "observations", "datatype": "<class 'int'>"}, "longest_path":
{"category": "observations", "datatype": "<class 'int'>"}, "center_of_mass": {"category": "observations", "datatype": "<class 'tuple'>"}, "convex_hull_vertices": {"category":
"observations", "datatype": "<class 'int'>"}, "object_in_frame": {"category": "observations", "datatype": "<class 'bool'>"}, "ellipse_center": {"category": "observations",
"datatype": "<class 'tuple'>"}, "ellipse_major_axis": {"category": "observations", "datatype": "<class 'int'>"}, "ellipse_minor_axis": {"category": "observations",
"datatype": "<class 'int'>"}, "ellipse_angle": {"category": "observations", "datatype": "<class 'float'>"}, "ellipse_eccentricity": {"category": "observations", "datatype":
"<class 'float'>"}, "roi": {"category": "observations", "datatype": "<class 'int'>"}}, "entities": [{"metadata": {"camera": {"label": "camera identifier", "datatype": "<class
'str'>", "value": "cameraA"}, "imgtype": {"label": "image type", "datatype": "<class 'str'>", "value": "none"}, "zoom": {"label": "camera zoom setting", "datatype": "<class
'str'>", "value": "none"}, "exposure": {"label": "camera exposure setting", "datatype": "<class 'str'>", "value": "none"}, "gain": {"label": "camera gain setting",
"datatype": "<class 'str'>", "value": "none"}, "frame": {"label": "image series frame identifier", "datatype": "<class 'str'>", "value": "none"}, "lifter": {"label": "imaging
platform height setting", "datatype": "<class 'str'>", "value": "none"}, "timestamp": {"label": "datetime of image", "datatype": "<class 'datetime.datetime'>", "value":
"2022.03.03-13.02.36"}, "id": {"label": "image identifier", "datatype": "<class 'str'>", "value": "side5"}, "plantbarcode": {"label": "plant barcode identifier", "datatype":
"<class 'str'>", "value": "RaspiY"}, "treatment": {"label": "treatment identifier", "datatype": "<class 'str'>", "value": "none"}, "cartag": {"label": "plant carrier
identifier", "datatype": "<class 'str'>", "value": "none"}, "measurementlabel": {"label": "experiment identifier", "datatype": "<class 'str'>", "value": "none"}, "other":
{"label": "other identifier", "datatype": "<class 'str'>", "value": "none"}, "image": {"label": "image file", "datatype": "<class 'str'>", "value": "/Users/magdalena/Desktop/
tepary/Aparna_Tepary_20220303/2022.03.03-13.02.36_RaspiY_side5_cameraA.jpg"}}, "observations": {}}, {"metadata": {"camera": {"label": "camera identifier", "datatype": "<class
'str'>", "value": "cameraA"}, "imgtype": {"label": "image type", "datatype": "<class 'str'>", "value": "none"}, "zoom": {"label": "camera zoom setting", "datatype": "<class
'str'>", "value": "none"}, "exposure": {"label": "camera exposure setting", "datatype": "<class 'str'>", "value": "none"}, "gain": {"label": "camera gain setting",
"datatype": "<class 'str'>", "value": "none"}, "frame": {"label": "image series frame identifier", "datatype": "<class 'str'>", "value": "none"}, "lifter": {"label": "imaging
platform height setting", "datatype": "<class 'str'>", "value": "none"}, "timestamp": {"label": "datetime of image", "datatype": "<class 'datetime.datetime'>", "value":
"2022.03.03-12.41.38"}, "id": {"label": "image identifier", "datatype": "<class 'str'>", "value": "side2"}, "plantbarcode": {"label": "plant barcode identifier", "datatype":
"<class 'str'>", "value": "RaspiY"}, "treatment": {"label": "treatment identifier", "datatype": "<class 'str'>", "value": "none"}, "cartag": {"label": "plant carrier
identifier", "datatype": "<class 'str'>", "value": "none"}, "measurementlabel": {"label": "experiment identifier", "datatype": "<class 'str'>", "value": "none"}, "other":
{"label": "other identifier", "datatype": "<class 'str'>", "value": "none"}, "image": {"label": "image file", "datatype": "<class 'str'>", "value": "/Users/magdalena/Desktop/
tepary/Aparna_Tepary_20220303/2022.03.03-12.41.38_RaspiY_side2_cameraA.jpg"}}, "observations": {}}, {"metadata": {}, "observations": {"default": {"in_bounds": {"trait":
"whether the plant goes out of bounds ", "method": "plantcv.plantcv.within_frame", "scale": "none", "datatype": "<class 'bool'>", "value": true, "label": "none"}, "area":
{"trait": "area", "method": "plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 20861.0, "label": "pixels"}, "convex_hull_area":
{"trait": "convex hull area", "method": "plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 53349.5, "label": "pixels"}, "solidity":
{"trait": "solidity", "method": "plantcv.plantcv.analyze_object", "scale": "none", "datatype": "<class 'float'>", "value": 0.391025220480042, "label": "none"}, "perimeter":
{"trait": "perimeter", "method": "plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 2717.6850098371506, "label": "pixels"}, "width":
{"trait": "width", "method": "plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 355, "label": "pixels"}, "height": {"trait": "height",
"method": "plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 214, "label": "pixels"}, "longest_path": {"trait": "longest path",
"method": "plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 2186, "label": "pixels"}, "center_of_mass": {"trait": "center of mass",
"method": "plantcv.plantcv.analyze_object", "scale": "none", "datatype": "<class 'tuple'>", "value": [1001.5609031206558, 1202.965725516514], "label": ["x", "y"]},
"convex_hull_vertices": {"trait": "convex hull vertices", "method": "plantcv.plantcv.analyze_object", "scale": "none", "datatype": "<class 'int'>", "value": 23, "label":
"none"}, "object_in_frame": {"trait": "object in frame", "method": "plantcv.plantcv.analyze_object", "scale": "none", "datatype": "<class 'bool'>", "value": true, "label":
"none"}, "ellipse_center": {"trait": "ellipse center", "method": "plantcv.plantcv.analyze_object", "scale": "none", "datatype": "<class 'tuple'>", "value":
[1024.5435791015625, 1205.201171875], "label": ["x", "y"]}, "ellipse_major_axis": {"trait": "ellipse major axis length", "method": "plantcv.plantcv.analyze_object", "scale":
"pixels", "datatype": "<class 'int'>", "value": 349.7687683105469, "label": "pixels"}, "ellipse_minor_axis": {"trait": "ellipse minor axis length", "method":
"plantcv.plantcv.analyze_object", "scale": "pixels", "datatype": "<class 'int'>", "value": 178.50100708007812, "label": "pixels"}, "ellipse_angle": {"trait": "ellipse major
axis angle", "method": "plantcv.plantcv.analyze_object", "scale": "degrees", "datatype": "<class 'float'>", "value": 80.2846908569336, "label": "degrees"},
"ellipse_eccentricity": {"trait": "ellipse eccentricity", "method": "plantcv.plantcv.analyze_object", "scale": "none", "datatype": "<class 'float'>", "value":
0.8599726995393328, "label": "none"}, "roi": {"trait": "roi", "method": "roi", "scale": "int", "datatype": "<class 'int'>", "value":
"2022.03.03-12.50.25_RaspiY_side7_cameraA.jpg", "label": "#"}}}}, {"metadata": {"camera": {"label": "camera identifier", "datatype": "<class 'str'>", "value": "cameraA"},
"imgtype": {"label": "image type", "datatype": "<class 'str'>", "value": "none"}, "zoom": {"label": "camera zoom setting", "datatype": "<class 'str'>", "value": "none"},
"exposure": {"label": "camera exposure setting", "datatype": "<class 'str'>", "value": "none"}, "gain": {"label": "camera gain setting", "datatype": "<class 'str'>", "value":
"none"}, "frame": {"label": "image series frame identifier", "datatype": "<class 'str'>", "value": "none"}, "lifter": {"label": "imaging platform height setting", "datatype":
"<class 'str'>", "value": "none"}, "timestamp": {"label": "datetime of image", "datatype": "<class 'datetime.datetime'>", "value": "2022.03.03-12.44.55"}, "id": {"label":
"image identifier", "datatype": "<class 'str'>", "value": "side4"}, "plantbarcode": {"label": "plant barcode identifier", "datatype": "<class 'str'>", "value": "RaspiY"},
"treatment": {"label": "treatment identifier", "datatype": "<class 'str'>", "value": "none"}, "cartag": {"label": "plant carrier identifier", "datatype": "<class 'str'>",
"value": "none"}, "measurementlabel": {"label": "experiment identifier", "datatype": "<class 'str'>", "value": "none"}, "other": {"label": "other identifier", "datatype":
"<class 'str'>", "value": "none"}, "image": {"label": "image file", "datatype": "<class 'str'>", "value": "/Users/magdalena/Desktop/tepary/
Aparna_Tepary_20220303/2022.03.03-12.44.55_RaspiY_side4_cameraA.jpg"}}, "observations": {}}, {"metadata": {"camera": {"label": "camera identifier", "datatype": "<class
'str'>", "value": "cameraA"}, "imgtype": {"label": "image type", "datatype": "<class 'str'>", "value": "none"}, "zoom": {"label": "camera zoom setting", "datatype": "<class
'str'>", "value": "none"}, "exposure": {"label": "camera exposure setting", "datatype": "<class 'str'>", "value": "none"}, "gain": {"label": "camera gain setting",
"datatype": "<class 'str'>", "value": "none"}, "frame": {"label": "image series frame identifier", "datatype": "<class 'str'>", "value": "none"}, "lifter": {"label": "imaging
```

In our lab, we typically prefer to work with the .csv files rather than json files. To transform the json file into csv files - please run the below code adjusting the path to your **"plantcv-utils.py"** file and path to your "json" file containing the results - produced by Step 15 of THIS protocol.

```
python /Users/magda/Documents/PlantCV/plantcv-utils.py json2csv --json /Users/magda/Documents/PlantCV/Aparna/Results/2022.03.05.sample_image.json --csv /Users/magda/Documents/PlantCV/Aparna/Results
```

The script will result in two files being produced:
- Results-multi-value-traits.csv
- Results-single-value-traits.csv

NOTE: if you are processing individual days separately - you might want to add a DATE identifier to these files before moving on with the next day processing - otherwise they will be renamed when you will run the next itteration of the code. If that happens - you can always re-run this step for the specific json file - as that will NOT be renamed - since the names are going to be unique (if you change them in the json file for each day in STEP 11!)

Analysis of the data

17 Processing the data further to calculate the cumulative plant area from all 7 angles as well as growth rates can be done in R. Please follow the example pipeline developed by Li'ang and Hayley in R, which can be accessed here.

NOTE: it is a good practice to run a correlation analysis between your cumulative area and the fresh weight (if recorded) on the final day of the experiment.