



SEMESTER PROJECT - MASTER IN ROBOTICS

Agriculture: Automated Phenotyping

AUTHOR:

YANIS BOUADI (283606)

PROFESSOR:

JOSIE HUGHES

DOCTORAL ASSISTANTS:

MAX POLZIN

SHIV ASHUTOSH



January, 2022

Acknowledgements

I would like to take this opportunity to express my gratitude to Professor Josie Hughes, head of CREATE Lab at EPFL as well as Doctoral assistants Max Polzin and Shiv Ashutosh for their precious weekly feedback and inputs on the project. Thanks for your interest and passion.

Contents

1	Introduction	1
2	Method Implementation	2
3	Results	4
3.1	Influence of the dilation on the metrics	5
3.2	Influence of the contrast on the metrics	5
3.3	Influence of the brightness on the metrics	6
4	Discussion & Conclusion	6
5	References	7
6	Appendix	8
6.1	Results on Image 1: Using ImageJ software versus my approach	8
6.2	Results on Image 2: Using ImageJ software versus my approach	12
6.3	Github: Link to the code	14
6.4	UNIL Dataset	15
6.5	Protocol to plant seeds and monitor the growth	15

1 Introduction

A plant phenotype is the result of the interactions between the genome of a stationary plant and all the micro- and mega-environments encountered during its life span. Modern plant phenotyping is often using non-invasive technologies and digital technologies to provide essential information on how genetics and environmental pressures can guide selection toward productive plants suitable for their environment. More specifically, plant phenotyping is the assessment of complex plant traits such as growth, development, tolerance, resistance, architecture, physiology, ecology and yield. This study focuses only on roots morphology such as length and depth of main and lateral roots. *ImageJ* is the software currently used in biology lab at UNIL to extract measurements from scanned images of the roots. By manually segmenting the roots, ImageJ outputs a measurement that is accurate enough for phenotyping. The inconvenient of this method is that it's time-consuming to point at the roots one by one manually specially when you have hundreds of roots to analyse.

That's why is important to automate the phenotyping. Several methods can be used. The following were tried during the semester but were not robust enough to be reliable.

- **KNN algorithm:** This method is based on computing all the distances between all the points. After recomputing several times the algorithm, the point are segmented into K different clusters. The issue with this method is that, even if the roots are well separated along the horizontal axis, the clustering is not accurate enough. Indeed, lateral roots are assigned to the wrong root.
- **Region growing algorithm:** This algorithm is based on exploring the pixel around a chosen pixel. Therefore, one idea can be, once a binary image is get (white pixels for roots and black pixels for the background), take randomly X points among the white pixels and then explore all the pixels around. Recompute until X separate regions are get.
- **First and second derivatives:** As the roots are not straight lines, the curvature changes all along the root. By computing first and second derivatives and look at their positive or negative sign, it will be possible to get the position of these changes in curvatures and therefore have a linear model of the root.

In this report, first part will be dedicated in explaining each steps of the algorithm implemented. Then, the results will be presented with a focus on the robustness of the algorithm regarding the contrast and the brightness of the input image. Finally, these results will be discussed with a critical point of view.

2 Method Implementation

Before beginning the image processing, these parameters needs to be entered:

- **Input image:** RGB image
- **Mode of use:** 'Fast' / 'Recommended' / 'Accurate'
- **Number of seeds:** that are planted in the petri dish
- **Stage:** 'early' if early-stage roots / 'late' if late-stage roots

Step 1: Load the images and crop them:

As the aim of the project is to automated the phenotyping process, it makes sense to assume that the setup for taking the images will be always the same. In addition to that, the seeds are always aligned at the same distance from the top of the petri dish (cf. red line in Figure 1). Starting from there, once all the images are taken, they can be cropped in order to exclude the border of the petri dish and reduce the noise and therefore ease the process.



Figure 1: Initial image (left) and cropped image (right)

Step 2: Get a binary image and contour the roots:

It's important to separate the background from the foreground (roots). Once this binary is get, the contouring can start using a '*measure*' library from '*skimage*'. It's based on **edge detection**. More specifically, it uses marching cubes algorithm [1]. As shown in Figure 2, it works well as it's used on binary image. Each color represents a contour detected by the algorithm.

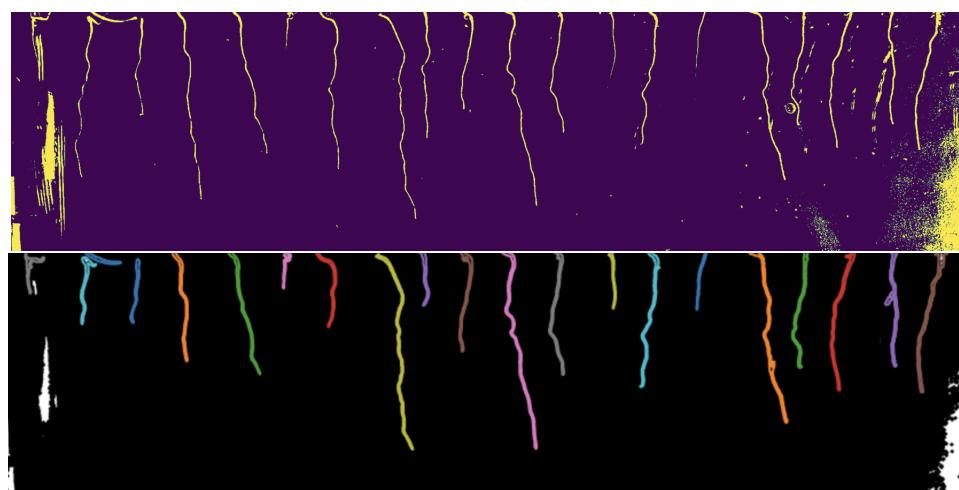


Figure 2: Binary image (top) and contouring (bottom)

Step 3: Noise reduction / Fine tuning:

Adaptive thresholds: The binary image got in step 2 is based on an adaptive threshold. For each image, the binary threshold is initialized as a function of the ratio between the white pixels (roots) and the black pixels (background). Then, another threshold, the one regarding the minimal size of a contour, increases or decreases depending on how many contours have been detected.

Dilation: 3 mode of use have been set in the algorithm: Mode 'Fast', Mode 'Recommended' and Mode 'Accurate'. In the first mode ('Fast'), there is no fine tuning, the image is analyzed right after the binary process. This mode can be used when the resolution of the image is high and the image processing seems easy. In the two other modes ('Recommended' and 'Accurate'), the image is dilated. The dilation **fills the gaps between the white pixels**. Indeed, when there are gaps along a root, the contouring process will detect it as different roots. By using dilation, most of these gaps are filled and the root contouring is therefore more accurate. Figure 3 illustrates the dilation.

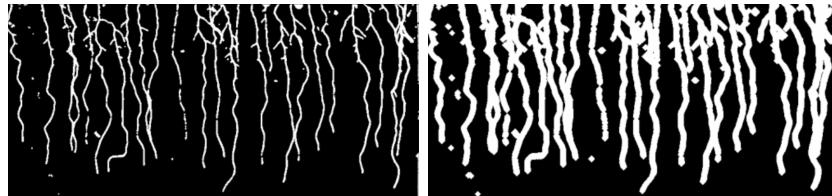


Figure 3: From left to right, dilation of the image: fill in the gaps between white pixels

Step 4: Compute the depth:

For each contour previously detected, which means each root, it's possible to know the position of the **lowest and highest points of the contour** along the vertical axis. Therefore, the depth of the main root is computed after converting the distance in pixels into a distance in centimeters.

Step 5: Compute the length:

When there are **no lateral roots detected**, the length is computed as half of the cumulative distance between each point along the entire contour.

When **lateral roots are detected**, as shown in Figure 4, the entire root is segmented vertically in parts where there is only the main root (red lines) and parts where lateral roots are detected (colorful lines). The segmentation is done by going through the whole contour point by point. As long as the contour is in the up to down direction, it is considered as the main root. As soon as a rising edge is detected, it means either that the main root is finished or that it is a side root. The total length of the main root is the sum between the total length of parts with no lateral roots (red lines) and the depth of the part with lateral roots (colorful lines). Indeed, in the black circled area in Figure 4, the total length of the main root is the sum of the depth of the green contour, the length of the red contour and the depth of the orange contour.

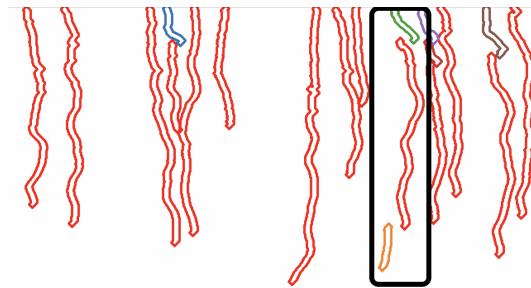
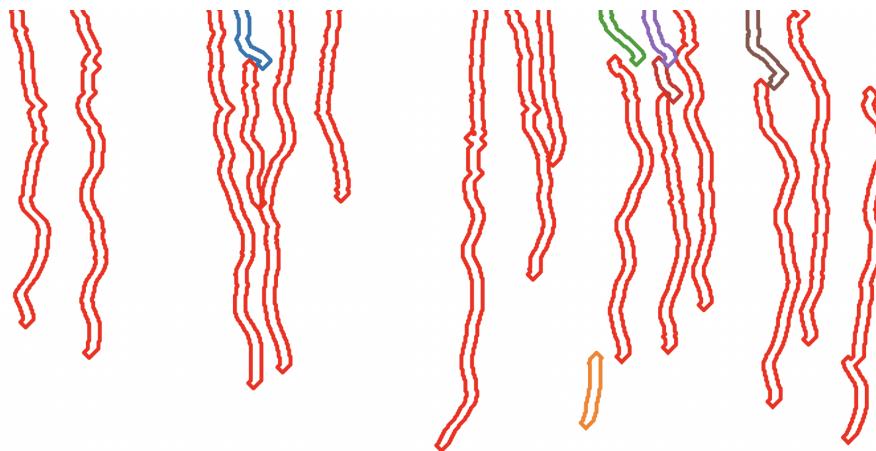


Figure 4: Red lines are parts with no lateral roots / Colorful lines represent parts with lateral roots

3 Results

All the work done in the previous steps needs to be evaluated. Therefore, the error, between the ground truth and the output of the algorithm, is computed for depths and lengths. Note that all the complete results are available in Appendix. In order to go deeper in the analysis of the results, the algorithm will be challenged by changing the contrast and brightness of the input image. Here is an example of the output of the algorithm:



	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	3.55	4.10	0	
Root 2	3.88	4.46	0	
Root 3	4.22	4.75	0	
Root 4	1.64	1.87	1	[0.69]
Root 5	4.04	4.48	0	
Root 6	2.15	2.48	0	
Root 7	4.91	5.56	0	
Root 8	3.02	3.43	0	
Root 9	1.76	1.94	0	
Root 10	3.39	3.92	2	[0.84, 0.64]
Root 11	2.87	3.25	2	[0.5, 0.66]
Root 12	3.36	3.88	0	
Root 13	3.62	4.10	1	[0.91]
Root 14	3.74	4.26	0	
Root 15	3.93	4.48	0	
Mean values	3.34	3.80	0.4	[-]
GT Mean values	3.75	3.93	[-]	[-]
Error [%]	10.97	3.38	[-]	[-]

Figure 5: Outputs of the algorithm

3.1 Influence of the dilation on the metrics

In this subsection, let's analyze the importance of the Step 3 described in section 2 by quantifying the error for the 3 modes of use: 'Fast', 'Recommended' and 'Accurate'.

Figure 6 shows the results for an image of early stage roots while Figure 7 shows the results for an image of a more complex root network. For both early-stage and late-stage roots, the error for the modes 'Recommended' and 'Accurate' are smaller than the one in mode 'Fast'. It means that the dilation method is efficient. The difference is more noticeable for late stage roots when the roots network is more complex (Figure 7).

	Mode "Fast"	Mode "Recommended"	Mode "Accurate"
Error on the depth measurement [%]	9.4	9.4	7.8
Error on the length measurement [%]	1.5	1.5	2.6

Figure 6: Influence of the dilation on the error for image 1 (Create Lab - early stage roots)

	Mode "Fast"	Mode "Recommended"	Mode "Accurate"
Error on the depth measurement [%]	50.8	11.0	11.0
Error on the length measurement [%]	46.5	3.4	3.4

Figure 7: Influence of the dilation on the error for image 2 (UNIL - late stage roots)

3.2 Influence of the contrast on the metrics

Figure 8 shows the robustness of the algorithm when the contrast changes. Indeed, using an online software that changes uniformly the contrast of a given image, one can output an image more or less contrasted. The following results are for a change in contrast within the range $\pm 80\%$ using the mode 'Accurate' as it's the one that outputs more accurate results.

	-80%	-50%	-20%	0%	+20%	+50%	+80%
Error on the depth measurement [%]	3.8	3.8	3.8	7.8	18.0	23.8	14.1
Error on the length measurement [%]	17.8	18.0	18.0	2.6	33.5	39.6	27.2

Figure 8: Influence of the contrast on the error for image 1 (Create Lab - early stage roots)

	-80%	-50%	-20%	0%	+20%	+50%	+80%
Error on the depth measurement [%]	8.0	10.0	21.8	11.0	18.0	48.2	35.5
Error on the length measurement [%]	1.1	3.3	15.3	3.4	10.0	41.3	24.9

Figure 9: Influence of the contrast on the error for image 2 (UNIL - late stage roots)

3.3 Influence of the brightness on the metrics

Figure 10 shows the robustness of the algorithm when the brightness changes. Indeed, using the same software as for the contrast, one can output an image with more or less brightness. The following results are for a change in brightness within the range $\pm 80\%$ using the mode 'Accurate' as it's the one that outputs more accurate results.

	-80%	-50%	-20%	0%	+20%	+50%	+80%
Error on the depth measurement [%]	6.5	6.6	6.6	7.8	1.9	4.2	8.7
Error on the length measurement [%]	5.9	6.8	5.9	2.6	10.7	5.7	21.4

Figure 10: Influence of the brightness on the error for image 1 (Create Lab - early stage roots)

	-80%	-50%	-20%	0%	+20%	+50%	+80%
Error on the depth measurement [%]	26.1	30.0	29.1	11.0	19.1	16.8	33.6
Error on the length measurement [%]	18.5	22.9	23.1	3.4	12.5	10.0	27.4

Figure 11: Influence of the brightness on the error for image 2 (UNIL - late stage roots)

4 Discussion & Conclusion

All the results in section 3 give us an indication on how to not setup the lightning of the image. Indeed, the accuracy of the results decreases when the contrast and the brightness are increased. It does not seem to be the case when the contrast and the brightness are decreased. I did not had the opportunity to fine tune the algorithm after these results to make it more robust but these results are very promising as the algorithm is quite robust to contrast and brightness.

However, even if these two parameters have been varied from $\pm 80\%$, the brightness and the contrast have been changed uniformly on the entire image. It's important to mention that one limitation of the algorithm can be when one spot is much brighter than another one. All the adaptive thresholds will be biased by the brighter spot and affects the detection of darker spots. This is not an issue when the lightning environment is controlled. Indeed, the aim of this project is to fully automate the phenotyping process once the seeds are planted. Therefore, the setup with a robotic gripper that will bring petri dishes one at the time to a camera will allow to take pictures in a controlled environment in term of contrast and brightness. Once all this setup is done, the results should be much better and reach less than 10% for all kind of contrast and brightness tests.

Dataset, coding, results and final presentation are available here:
https://github.com/bouadiyanis/SemesterProject_CreateLab_YanisBouadi

5 References

1. Lorensen, William and Harvey E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", Computer Graphics SIGGRAPH 87 Proceedings, Vol.21 N°4, July 1987, p. 163-170.
<https://people.eecs.berkeley.edu/~jrs/meshpapers/LorensenCline.pdf>
2. Thomas Burrell, "The Microphenotron: a robotic miniaturized plant phenotyping platform with diverse applications in chemical biology", Burrell et al. Plant Methods, 2017.
https://www.researchgate.net/publication/314165418_The_Microphenotron_A_robotic_miniaturized_plant_phenotyping_platform_with_diverse_applications_in_chemical_biology
3. Jonas A Ohlsson, "SPIRO – the automated Petri plate imaging platform designed by biologists, for biologists", Department of Molecular Sciences, Uppsala BioCenter, Swedish University of Agricultural Sciences and Linnean Center for Plant Biology, Sweden, 2021.
https://www.researchgate.net/publication/350109832_SPIRO--the_automated_Petri_plate_imaging_platform_designed_by_biologists_for_biologists
4. Ajay Mudgil, "Plant Growth Monitoring System", International Journal of Engineering Research Technology (IJERT), Vol. 1 Issue 4, June - 2012.
<https://www.ijert.org/research/plant-growth-monitoring-system-IJERTV1IS4185.pdf>
5. S. Ducournau, "An image acquisition system for automated monitoring of the germination rate of sunflower seeds", Computers and Electronics in Agriculture 44 (2004) 189–202.
<https://www.sciencedirect.com/science/article/pii/S0168169904000791>
6. Hsiang Sing Naik, "A real-time phenotyping framework using machine learning for plant stress severity rating in soybean", Naik et al. Plant Methods, 2017.
https://www.researchgate.net/publication/315829448_A_real-time_phenotyping_framework_using_machine_learning_for_plant_stress_severity_rating_in_soybean

6 Appendix

6.1 Results on Image 1: Using ImageJ software versus my approach

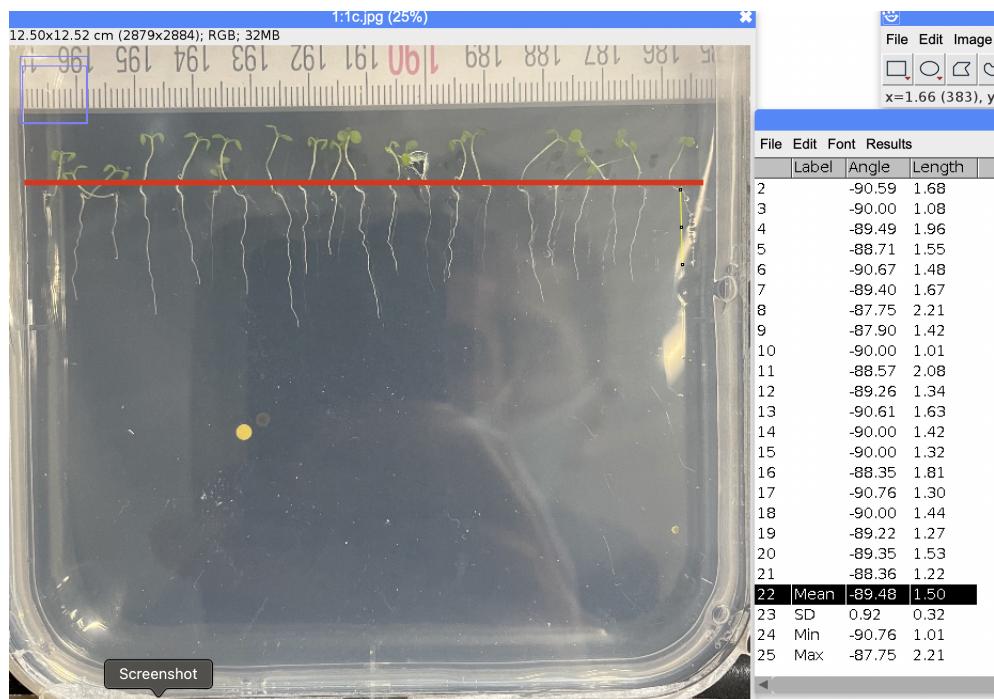


Figure 12: Results on image 1 (Create Lab - early stage roots): Measuring the depths using ImageJ

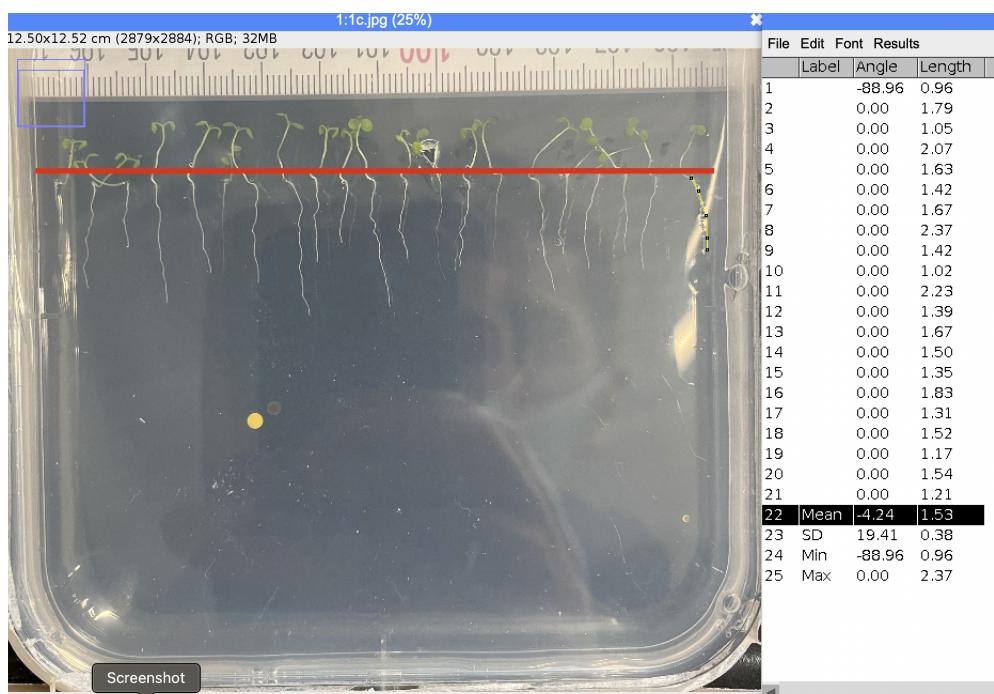


Figure 13: Results on image 1 (Create Lab - early stage roots): Measuring the lengths using ImageJ

	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	0.42	0.52	0	[]
Root 2	1.36	1.48	0	[]
Root 3	1.53	1.70	0	[]
Root 4	0.94	1.15	0	[]
Root 5	0.68	0.74	0	[]
Root 6	1.24	1.39	0	[]
Root 7	2.44	2.73	0	[]
Root 8	1.53	1.69	0	[]
Root 9	0.71	0.76	0	[]
Root 10	1.68	1.90	0	[]
Root 11	0.73	0.79	0	[]
Root 12	2.12	2.38	0	[]
Root 13	1.44	1.62	0	[]
Root 14	1.72	1.92	0	[]
Root 15	1.43	1.57	0	[]
Root 16	1.75	2.06	0	[]
Root 17	2.29	2.88	2	[0.23, 1.02]
Root 18	0.46	0.67	0	[]
Mean values	1.36	1.55	0.1	[·]
GT Mean values	1.50	1.53	[·]	[·]
Error [%]	9.37	1.49	[·]	[·]

Figure 14: Results on image 1 (Create Lab - early stage roots): Mode 'Fast'

	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	0.42	0.52	0	[]
Root 2	1.36	1.48	0	[]
Root 3	1.53	1.70	0	[]
Root 4	0.94	1.15	0	[]
Root 5	0.68	0.74	0	[]
Root 6	1.24	1.39	0	[]
Root 7	2.44	2.73	0	[]
Root 8	1.53	1.69	0	[]
Root 9	0.71	0.76	0	[]
Root 10	1.68	1.90	0	[]
Root 11	0.73	0.79	0	[]
Root 12	2.12	2.38	0	[]
Root 13	1.44	1.62	0	[]
Root 14	1.72	1.92	0	[]
Root 15	1.43	1.57	0	[]
Root 16	1.75	2.06	0	[]
Root 17	2.29	2.88	2	[0.23, 1.02]
Root 18	0.46	0.67	0	[]
Mean values	1.36	1.55	0.1	[-]
GT Mean values	1.50	1.53	[-]	[-]
Error [%]	9.37	1.49	[-]	[-]

Figure 15: Results on image 1 (Create Lab - early stage roots): Mode 'Recommended'

	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	0.62	0.66	0	[]
Root 2	1.25	1.40	0	[]
Root 3	2.38	2.59	0	[]
Root 4	1.81	2.03	0	[]
Root 5	0.52	0.62	0	[]
Root 6	2.01	2.25	0	[]
Root 7	2.62	2.98	0	[]
Root 8	1.62	1.76	0	[]
Root 9	1.26	1.41	0	[]
Root 10	2.46	2.77	0	[]
Root 11	1.55	1.72	0	[]
Root 12	0.88	0.95	0	[]
Root 13	1.70	1.93	0	[]
Root 14	0.75	0.83	0	[]
Root 15	1.16	1.37	1	[0.79]
Root 16	1.15	1.30	0	[]
Root 17	1.74	1.99	0	[]
Root 18	1.00	1.12	0	[]
Root 19	1.57	2.01	1	[0.59]
Root 20	0.71	0.90	1	[0.41]
Root 21	0.28	0.38	0	[]
Mean values	1.38	1.57	0.1	[]
GT Mean values	1.50	1.53	[]	[]
Error [%]	7.81	2.61	[]	[]

Figure 16: Results on image 1 (Create Lab - early stage roots): Mode 'Accurate'

6.2 Results on Image 2: Using ImageJ software versus my approach

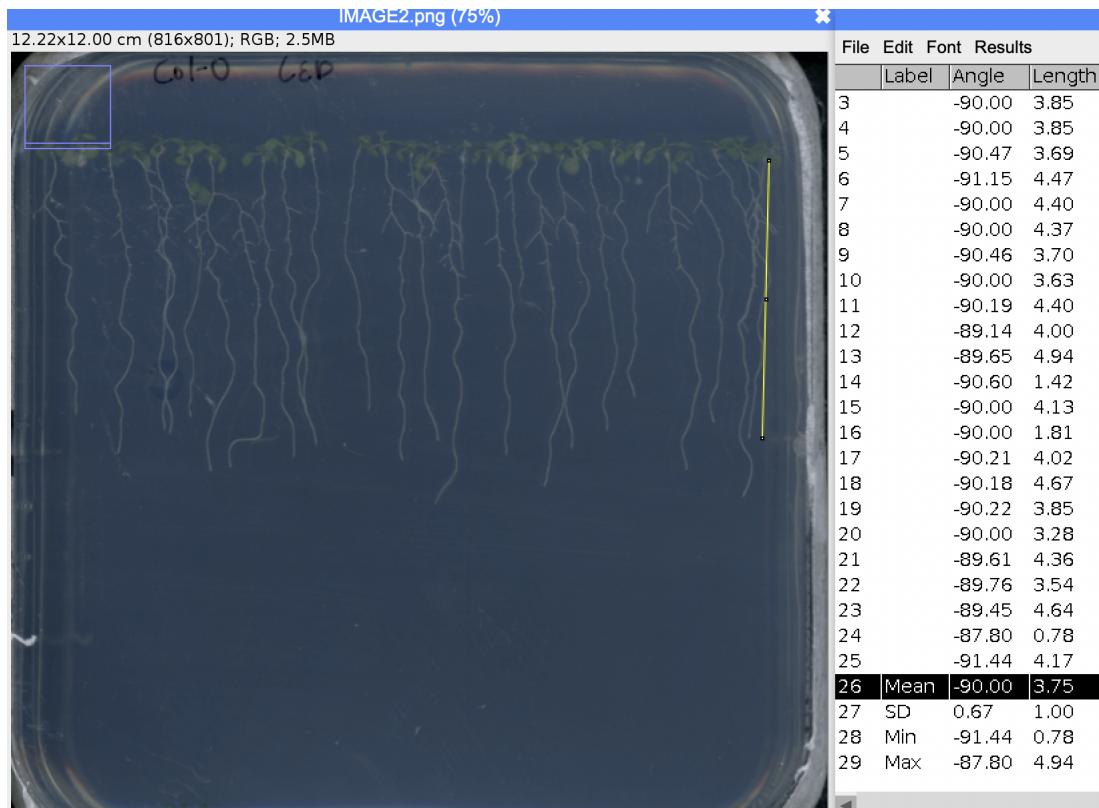


Figure 17: Results on image 2 (UNIL - late stage roots): Measuring the depths using ImageJ

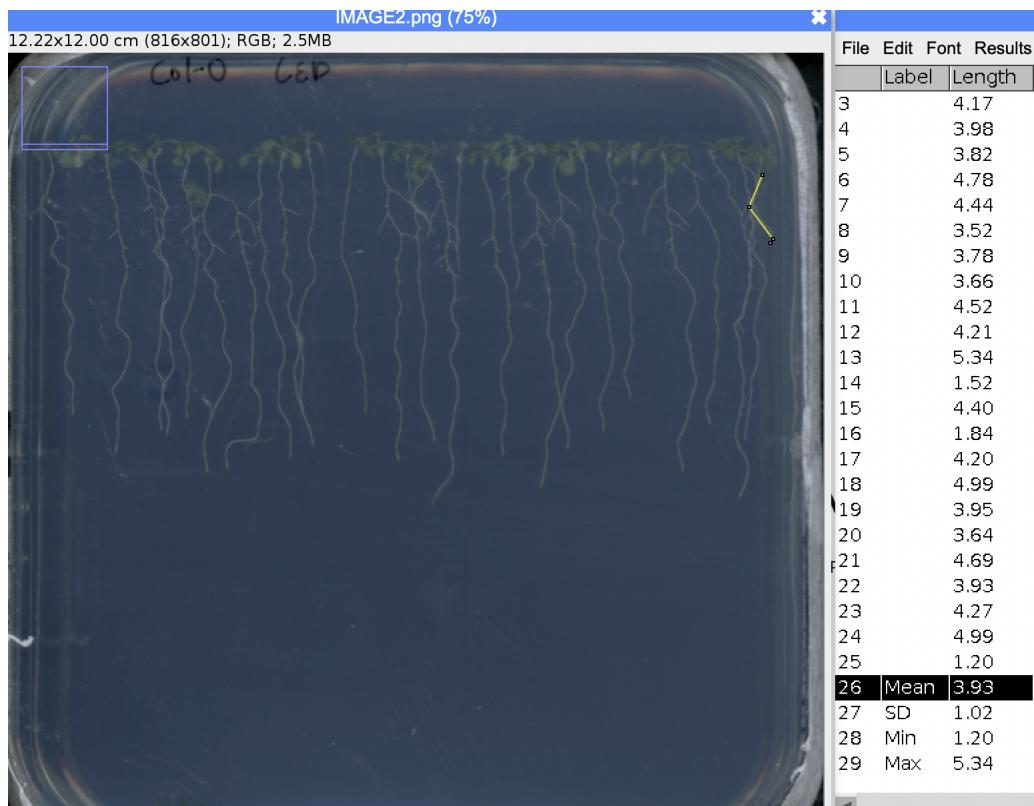


Figure 18: Results on image 2 (UNIL - late stage roots): Measuring the lengths using ImageJ

	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	0.40	0.46	0	[]
Root 2	2.96	3.41	0	[]
Root 3	3.31	3.72	1	[1.26]
Root 4	1.14	1.33	1	[0.79]
Root 5	0.13	0.18	0	[]
Root 6	3.62	4.09	1	[0.88]
Root 7	0.13	0.18	0	[]
Root 8	3.08	3.45	1	[0.95]
Mean values	1.85	2.10	0.5	[-]
GT Mean values	3.75	3.93	[-]	[-]
Error [%]	50.77	46.50	[-]	[-]

Figure 19: Results on image 2 (UNIL - late stage roots): Mode 'Fast'

	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	3.55	4.10	0	[]
Root 2	3.88	4.46	0	[]
Root 3	4.22	4.75	0	[]
Root 4	1.64	1.87	1	[0.69]
Root 5	4.04	4.48	0	[]
Root 6	2.15	2.48	0	[]
Root 7	4.91	5.56	0	[]
Root 8	3.02	3.43	0	[]
Root 9	1.76	1.94	0	[]
Root 10	3.39	3.92	2	[0.84, 0.64]
Root 11	2.87	3.25	2	[0.5, 0.66]
Root 12	3.36	3.88	0	[]
Root 13	3.62	4.10	1	[0.91]
Root 14	3.74	4.26	0	[]
Root 15	3.93	4.48	0	[]
Mean values	3.34	3.80	0.4	[-]
GT Mean values	3.75	3.93	[-]	[-]
Error [%]	10.97	3.38	[-]	[-]

Figure 20: Results on image 2 (UNIL - late stage roots): Mode 'Recommended'

	Main Root Depth [cm]	Main Root Length [cm]	Number of lateral roots	Lateral root depths [cm]
Root 1	3.55	4.10	0	[]
Root 2	3.88	4.46	0	[]
Root 3	4.22	4.75	0	[]
Root 4	1.64	1.87	1	[0.69]
Root 5	4.04	4.48	0	[]
Root 6	2.15	2.48	0	[]
Root 7	4.91	5.56	0	[]
Root 8	3.02	3.43	0	[]
Root 9	1.76	1.94	0	[]
Root 10	3.39	3.92	2	[0.84, 0.64]
Root 11	2.87	3.25	2	[0.5, 0.66]
Root 12	3.36	3.88	0	[]
Root 13	3.62	4.10	1	[0.91]
Root 14	3.74	4.26	0	[]
Root 15	3.93	4.48	0	[]
Mean values	3.34	3.80	0.4	[]
GT Mean values	3.75	3.93	[]	[]
Error [%]	10.97	3.38	[]	[]

Figure 21: Results on image 2 (UNIL - late stage roots): Mode 'Accurate'

6.3 Github: Link to the code

Dataset, coding, results and final presentation are available here:

https://github.com/bouadiyanis/SemesterProject_CreateLab_YanisBouadi

6.4 UNIL Dataset

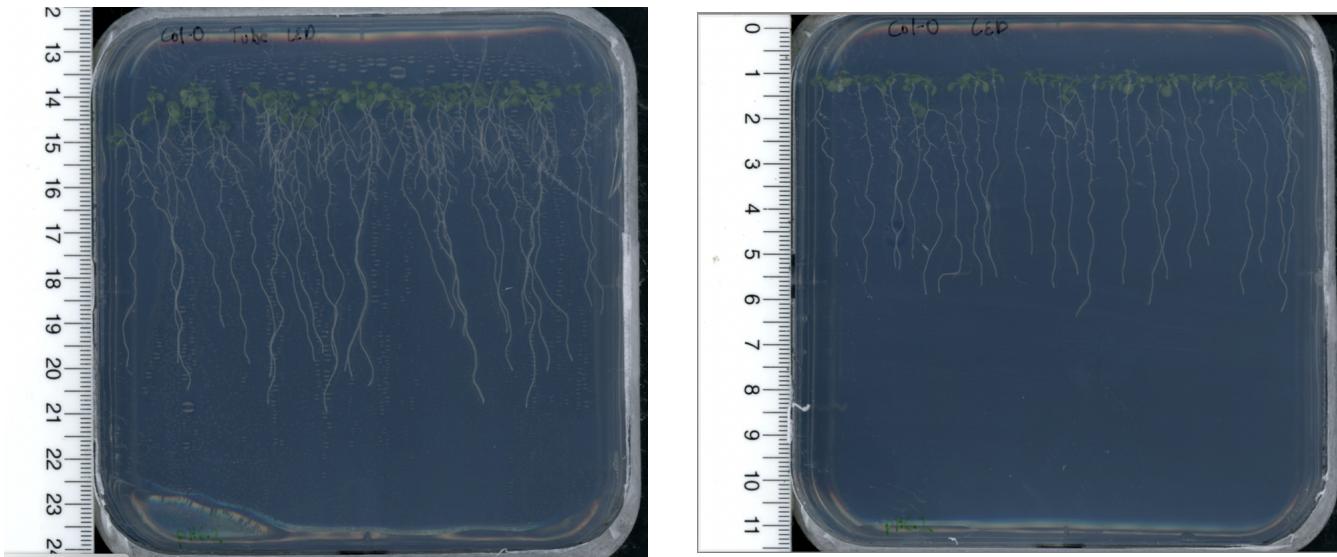


Figure 22: Pictures provided by UNIL Lab

6.5 Protocol to plant seeds and monitor the growth

Seeds sterilization Petri Dish:

1. Add 1mL of ethanol (75%) + Triton (0.05%) in a tube containing the seeds.
2. Shake the tube for 10 minutes maximum.
3. Remove the liquid using a pipette without touching the seeds.
4. Add 1mL of ethanol (100%) and shake gently for 20 minutes maximum.
5. Pour the content of the tube (seeds and ethanol) on a filter under a laminar flow hood and wait until the ethanol evaporates.
6. Use a sterilized toothpick to pick and place the seeds from the filter to the Petri dish

Seeds Growth: The following material is needed:

- Scanner with a high resolution
- Support to stack vertically the Petri dishes
- Room with an ambient temperature of 20°C