Programming Project 03

Francel Lamprecht, Christine Robinson, Regina Wehler

Winter Semester 2018/19

Contents

1 Introduction

REMOVE LATER!This is a dummy sentence that shows how citations work (Adams et al., 2018).

Plant phenotyping is a research area in the plant sciences that focuses on the quantitative measurement of both functional and structural properties of plants. During the last decade plant phenotyping evolved to non-destructive, image-analysis-based phenotyping. This computer driven approach allows the characterization of plants in a high-throughput manner (Walter et al., 2015).

The focus of this plant phenotyping research project was on the processing, segmentation and classification of Arabidopsis, to gain valuable insight on how the image-analysis-based phenotyping in the life sciences works.

Arabidopsis is often used as a model plant, as the plant is small in size, with a rapid life cycle of about six weeks (Koornneef and Meinke, 2010). Arabidopsis is a rosette plant, with the individual leaves being on separate, distinguishable stems. This characteristic of Arabidopsis makes leaf separation from each other less complex. Given that we were only exploring viable methods in this research area, Arabidopsis was our chosen plant, as it gave us the opportunity to explore with different algorithms in an easier way than what the other given data set, Tobacco, would have allowed us to do.

2 Methodology

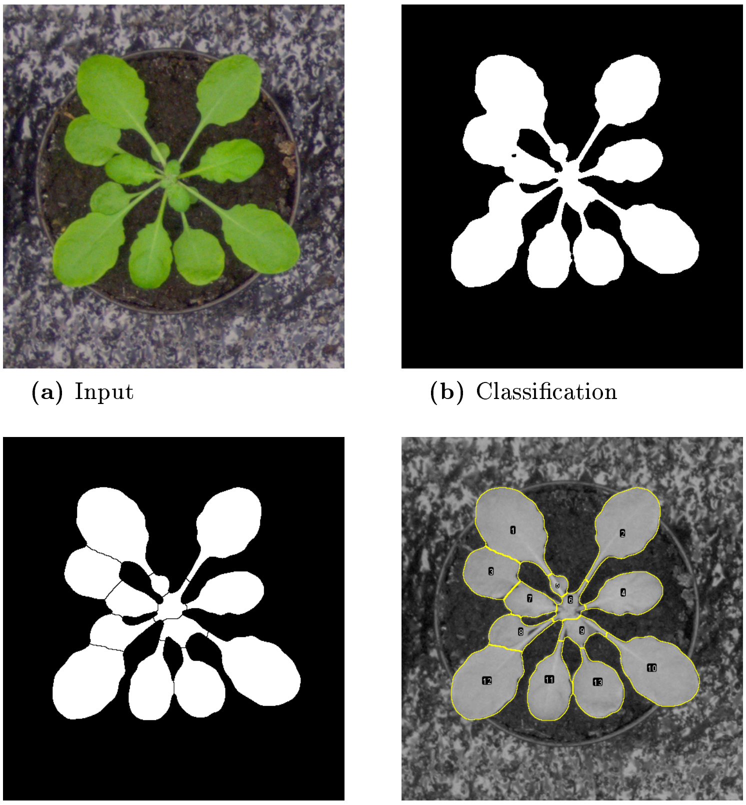
The image analysis approach consists of three main steps (figure 1). First, the input RGB input image is segmented into two class, plant and background by a trainable Weka segmentation classifier. Second, the binary class output is used to identify single leaf objects by watershed segmentation. The resulting binary mask is utilized to analyze properties of the leaf objects in the single red, green and blue channel, respectively.

2.1 Segmentation

The plant is separated from the background with the help of a trainable Weka segmentation classifier, which is available as a plugin for Fiji. Weka provides a GUI to train machine learning algorithms to produce pixel-based segmentations. The user can add traces to classes and train the classifier with those. Afterwards, traces/regions of interest can be adjusted and the classifier can be re-trained to improve classification. Six representative pictures are selected from the 2017 data set for training (plant029, plant145 and plant159 from A1; plant032, plant034 and plant037 from A2). Pictures are chosen to cover the whole range of plant green shades and the range of background characteristics of the given data sets. The Weka Experimenter was used to assess the performance of different machine learning algorithms. Based on these results (table 1), FastRandomForest was used as a classifier. By applying the trained classifier to the RGB input images, for each input image a binary classification image is obtained (figure 1-b).

2.2 Objects Recognition

A plant consists of leaves that are attached to each other. To be able to analyze leaves individually, they need to be separated. Here, the watershed algorithm was used to separate touching objects. The algorithm first calculates an Euclidean distance map and determines center points as points which are, from a topological view, the ultimate eroded points, As the algorithm's name indicates, this topological map is "flooded" with water and at each



(a) Input (b) Classification

(c) Watershed (d) Particle Analysis

Figure 1 – The analysis of plant leaves is conducted in three steps. The RGB input image (a) is classified into plant and background by a trainable Weka segmentation classifier (b). Single leaf objects are identified by watershed segmentation (c). The binary mask is used to analyze properties in the split red, green and blue channel (d).

Table 1 – Comparison of classifier performance by Weka Experimenter. The default algorithm parameters were kept.

Algorithm	Percent correct	Precision	Recall	F score	Matthews correlation	AUC
FastRandomForest	99.93	1.00	1.00	1.00	1.00	1.00
SMO	92.52	0.89	0.93	0.91	0.85	0.93
k-nearest Neighbors	95.38	0.95	0.93	0.94	0.90	0.95
RandomSubSpace	99.85	1.00	1.00	1.00	1.00	1.00
Bagging	99.77	1.00	1.00	1.00	1.00	1.00
DesicisonTable	99.05	0.99	0.9	0.99	0.98	1.00

collision of two "watersheds", a line separating two objects is drawn. The output of the algorithm is the binary image with added watershed lines (figure 1-c).

2.3 Object Analysis

After an image was subjected to aforementioned steps, the resulting watershed image was then analyzed with Fiji's "Analyze Particles" command. The measurements that were selected to be performed during the particle analysis are:

- Area - Measures the area of selection in square pixels. This was used to obtain the object's size.
- Standard deviation - Used to generate the mean gray value. It is the standard deviation of the grey values.
- Mean gray value - Average gray value of the selection, which is used in the calculation of the average brightness.
- Modal gray value - Most frequently occurring gray value within the selection. This will always be 255 when the analysis is done on the watershed image.
- Min and Max gray value - The minimum and maximum gray value of the selection.
- Perimeter - An alternative way to define the size of an object as it is the length of the outside boundary of the selection.
- Centroid - This gives us the center point of the selection and uses the average X and Y coordinates for all the pixels in the selection or image.
- Bounding rectangle - This smallest possible rectangle that encloses the selection. It gives the X and Y coordinates in respect of the upper left corner of the rectangle, as well as the height and width, from which we will calculate the height-width ratio from.
- Shape descriptors - These descriptors are used to attain the roundness of each object.
- Feret's diameter - Can be used to find the longest distance between any two points on the selection boundary.
- Integrated density- Used to indicate the sum of the brightness, as it sums the values of the pixels in the image or selection
- Median - Gives the median value of the pixels in the image or selection.
- Display labels - Selected to have an easy way to find the image name, as well the object ID.

Particles with a pixel size less than 70 were excluded from the "Analyze Particles" command. The value of 70 was established by trial and error, by inspecting the lines drawn by the watershed algorithm to separate objects.

Fiji automatically generates a Results Table, with the selected measurements. These results can be accessed by using the *getResultsTable()* method in the source code, to select the appropriate results for further use. By using this method a csv file with all the results deemed necessary was created.

Table 2 – Measurement options, Results Table results and the corresponding value in the csv file.

Measurement option	Results Table results	Corresponding value in the csv file
Display Labels	Index and Label	objectID and imageID
Area	Area	size
Centroid	X and Y	pos x and pos y
Shape descriptors	Round	roundness
Integrated density	RawIntDen	brightness sum
Mean gray value	Mean	brightness average
Bounding rectangle	Width and Height	width and height ratio

The generated Results Table results ("columns") that were included in the csv file can be seen in Table 2.

As brightness values calculated from binary watershed images is insufficient for plants, Fiji's "Analyze Particles" command was also applied to the original RGB images. The brightness (both the sum and the average) of each of the three color channels (red, green and blue) were individually calculated. These individual channel results were then added together to calculate the sum of the brightness and average brightness for each object. To ensure computational efficiency not all the measurements were selected again, as this would have resulted in the duplication of results, eg. the size of the objects will remain constant, it is independent of the type of input image. The measurements that were necessary to be applied to the original RGB images, are the mean gray value and the integrated density. Results regarding the brightness of each color channel were attained and added to the csv file.

An additional requirement of the project was to include the width to height ratio. As this is not a built-in measurement option in Fiji, the width and the height that were included in the Results Table, as part of the "Bounding rectangle" measurement were extracted and the ratio was calculated, by dividing the leaf width by the leaf height. The calculated width to height ratio was added to the csv file to be available for further use.

2.4 Explorative Data Analysis

Everything in Python

3 Results

4 Discussion & Outlook

- watershed is working best for circular objects, that's why it sometimes fails on leaves with long stems

Most of the decisions made about which measurements to include in the Fiji generated Results Table became clear when reading the ImageJ documentation. (Ferreira and Rasband, 2012). One decision that was needed to be made, was whether to take an object's position from the the top left appearing pixel or from the centroid. In this project we used the centroid to indicate the object's position, as the top left appearing pixel position has no advantage over the centroid. By using the centroid positions, we ensured that the position coordinates are more robust to rotations. If the top left appearing pixel was used, problems could arrive if the leaves were to be rotated.

5 References

Adams, T., Dörpinghaus, J., Jacobs, M., and Steinhage, V. (2018). Automated lung tumor detection and diagnosis in CT Scans using texture feature analysis and SVM. In *Communication Papers of the 2018 Federated Conference on Computer Science and Information Systems*, Annals of Computer Science and Information Systems, pages 13–20. PTI.

Ferreira, T. and Rasband, W. (2012). *ImageJ User Guide*.

Koornneef, M. and Meinke, D. (2010). The development of arabidopsis as a model plant. *The Plant Journal*, 61(6):909–921.

Walter, A., Liebisch, F., and Hund, A. (2015). Plant phenotyping: from bean weighing to image analysis. *Plant Methods*, 11(1):14.