Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images

Jean-Michel Pape
pape@ipk-gatersleben.de

Department of Molecular Genetics,
Leibniz Institute of Plant Genetics
and Crop Plant Research (IPK),
06466 Gatersleben, Germany

Christian Klukas
christian.klukas@lemnatec.de

LemnaTec GmbH,
52076 Aachen, Germany

Abstract

The segmentation of individual leaves in plant images is still a challenging task, especially in case of leaf overlaps. The exact determination of individual leaf areas could improve the biomass estimation which is a good indicator for plant performance. In addition, the number of leaves is directly related to plant development, leaf counts give insight into changing plant development stages. Machine learning is a powerful tool in vision tasks. Here we propose an approach including image analysis (based on the software IAP) for extraction of a comprehensive set of image features to predict the number of leaves for *Arabidopsis thaliana* and tobacco plants supplied by the organizers of the Leaf Counting Challenge (LCC) of the Computer Vision Problems in Plant Phenotyping (CVPPP) workshop in conjunction with the British Machine Vision Conference (BMVC) in 2015. In addition, we developed a method to detect exact leaf borders for resolving inaccurate leaf segmentation in case of leaf overlaps. For classifier training we evaluate a broad set of different colour and texture features. The predicted leaf borders are used as input for further image processing methods to complete the leaf segmentation. The results show the methods ability for improved leaf count estimations and for predicting leaf overlap borders, which helps to improve the segmentation of individual leaves.

1 Introduction

The results from the 2014 leaf segmentation challenge (LSC) in 2014 showed that there is still a need for improvement of individual leaf segmentation. Our previous approach [12] showed inaccuracies in resolving leaf overlaps in the training and test data sets (data sets and approaches for data analysis are described in detail in [15] and [10]). Here, the exact determination of the leaf shapes and leaf positions, in case of an overlap to determine which leaf is in front, is still a problem. In our previous work the problem was resolved by calculating straight split lines which do not take the real leaf shape into account. Even for techniques like so called matching techniques or shape modelling approaches, it is hard to estimate the real

Pages 3.1-3.12
https://dx.doi.org/10.5244/C.29.CVPPP.3

leaf position in case of leaf overlaps without additional information. Known approaches use time-lapse image data, 3-D information as input or images are acquired under well-defined conditions [1, 9, 18]. Additional information about the leaf borders would be essential to enhance the robustness in case of diverse imaging circumstances. Here, we try to resolve this problem by an enhanced image pre-processing. The extraction of the leaf borders would give an information about the real leaf shape, also in case of leaf overlaps. By utilizing a machine learning approach different features could be included for learning leaf border features. For general use cases machine-learning-based edge detection approaches have been developed before, but these do not focus on leaf borders in leaf overlap regions (e.g. [2]). Another problem we found in our previous work, is the low sensitivity for very small leaves, and we therefore systematically under-estimated the number of leaves [12]. In this work we present a machine-learning-based regression approach, which utilizes a larger number of phenotypic features, in this case calculated with our IAP software [7], to improve the prediction of leaf counts. The results of the leaf count predictions are further used to modify the sensitivity of our leaf labelling methods. In this paper we focus on combining approaches for predicting leaf counts, individual leaf labelling and the detection of leaf overlap lines, to improve the labelling. Based on the extracted leaf borders common image processing algorithms will be used to perform the leaf segmentation. The transformation of multiple image channels into the text-based ARFF file format allows us to use the Weka [3] machine learning toolkit not only for leaf count prediction, but also for the prediction of leaf border lines.

2 Methods

The overall image analysis approach contains three main steps: (1) The analysis of the general plant phenotypic appearance, using our IAP analysis pipeline. The phenotypic result data table is used to train a data-set-specific predictive regression model to better predict leaf counts. (2) We analyse colour- and texture-specific plant features, to create a prediction model for leaf borders in the regions of leaf overlaps. (3) The predicted leaf count numbers and the leaf border prediction is utilized to improve the reconstruction and labelling of individual leaves.

2.1 Leaf-Count Regression Model

This step of the workflow is based on a machine learning approach utilizing Weka for machine learning tasks and IAP for calculating a set of geometric features, based on the plant shape. At first, a feature evaluation and classifier training is performed by using several image features. In a second step, the best classifier is used to predict the leaf count on unknown testing data. The complete workflow is depicted in figure 3 on the right. For the extraction of a set of geometric features we used the IAP software. Most of the features are described in the reference paper for this software system [7], an overview is given in table 1. In addition, we added a feature which was developed and used for the direct prediction leaf counts in our LSC analysis pipeline of the previous year [12]. This leaf count feature, which is based on the Euclidian distance map maxima peak search, gives a good approximation for the number of leaves, especially for rosette plants, such as *Arabidopsis thaliana*. Also, a image moment calculation was added, which gives information about the orientation and shape of the plant (eccentricity). The features are calculated by using the ground truth foreground images included within the given data sets which were provided for the LCC 2015.

Table 1: List of extracted geometric image features using our IAP software, as basis for improved leaf count predictions.

Rank	Feature	Description
1	leaf count (distance map peaks)	Edm based peak number
2	border length	foreground border length
3	hull circularity	convex hull circularity
4	min rectangle (length small side)	size of smallest bounding rectangle
5	skeleton branch count	number of skeleton branch points
6	skeleton length	length of skeleton
7	convex hull farthest distance (PC1)	maximum distance within convex hull
8	circumcircle diameter	foreground circumcircle diameter
9	compactness	foreground compactness
10	skeleton endpoints	number of skeleton endpoints
12	2nd moment minor	2nd main axis
13	hull PC2	max. dist. in opposite direction of PC1
14	2nd moment major	first main axis
15	area / skeleton length	ratio between area and skeleton length
16	min rectangle area	area of minimum bounding rectangle
17	area	foreground area
18	hull area	area of convex hull
19	hull points	number of convex hull extreme points
20	hull fill grade	covered pixels by foreground
21	skeleton length/skeleton tips	ratio of skeleton length and # of endpoints

2.1.1 Feature Evaluation

To gain a deeper understanding on the importance of selected features in relation to the leaf counts, we performed a feature evaluation using the Weka [3] feature selection mode. We used the ReliefF-Attribute-Evaluation method (weka.attributeSelection.ReliefFAttributeEval) [6, 8, 14] and the Attribute-Selection-Ranker Search method, to access the average merit of the calculated features. The ReliefF algorithm evaluates the worth of attributes by repeatedly sampling an instance and then by accessing the distance to the nearest instance of the same class in comparison to a different class. An advantage of this method is that it can work on discrete as well as on continuous data. In this case we aimed for the prediction of continuous data, as our prediction goal should be an estimation of leaf-counts. A classification approach with a given set of categorical target values would require more training data for all possible leaf counts and was therefore regarded as not suitable. For parameterization, we used the default settings in Weka v3.6.12, all instances were sampled and evaluated, and 10 nearest neighbours were used to estimate the attribute relevance. The Attribute-Selection-Ranker ranks the attributes by their individual evaluations. The average rank was then used to order the features as shown in table 2. The overall rank table column was calculated by adding-up the average merit values for A1, A2 and A3, and by ordering the features accordingly. Overall, our previous approach of counting leaves by accessing the peaks of the foreground-distance-map, was found to be most valuable in our testing to predict the real leaf counts. After that the plant outline length, hull circularity and geometric bounds such as the size of the minimum enclosing rectangle, as well as the number of skeleton branch counts were found to be useful features. For the different data sets, the spread of the average merit values was quite small, and the ordering of the different features showed no clear pattern, when comparing the results for A1, A2, and A3. As the importance of the features didn't suggested a clear cut-off threshold for the A1 data set, we continued to use all calculated features in the leaf count prediction step (see section 2.1.2 and figure 2 for details).

Table 2: Feature evaluation, ranks of the top five features for A1, A2, A3 and overall

Rank	A1	A2	A3	Overall
1	hull fill grade	skeleton branch cnt	dist.map peaks	dist.map peaks
2	2nd moment minor	dist.map peaks	hull circularity	border length
3	leaf tip corner angle	compactness	border length	hull circularity
4	2nd moment major	skeleton length	hull PC1	min rectangle (b)
5	leaf tip width skewn.	2nd moment major	min rectangle (b)	skeleton branch cnt

2.1.2 Classifier Evaluation

To compare the general classifier performance, we compared all regression approaches (see figure 1), available by default in Weka (no additional packages were installed) using the experimenter interface. After loading the training data result tables (generated using IAP version 2.0.2), we used the colour-count data column as the target-column for the regression approach. The colour-count in the ground-truth label images corresponds to the ground-truth leaf count values minus one (background colour should be ignored).

For this leaf segmentation challenge situation, we selected for each data set the classifiers which showed the lowest error. For A1 the IBk algorithm was selected, for A2 the DecisionTable, and for A3 the RandomSubSpace (using the REPTree) algorithm. From the available and tested approaches Meta-Classifier RandomSubSpace [4], which uses the REPTree algorithm by default, showed the overall lowest classification errors. We did not consider the M5Rules [5, 13, 17] algorithm as it showed fluctuating performance for different testing situations and data bases.

The evaluations in Weka are based on floating point regression results, so in practice the prediction results will be slightly less accurate, as the leaf counts would need to be rounded.

To further find out, how influential the number of phenotypic features is to the prediction accuracy, we compared the performance of the approaches, when using the single best feature (from table 2), when using the top five features (as listed in the table), and when using all phenotypic features as extracted using IAP (see table 1 for details). Figure 2 illustrates the results. In all cases the classifier performance increased by incorporating the top five features, when compared to using a single predictive variable. For dataset A1 incorporating more than five features still significantly improved the error rate (mean absolute error), for A2 and A3 no further improvement could be achieved in these cases.

These comparisons are based in each case on a 10-fold cross of the training data and a 10 times repeated testing. More repetitions could be done and further statistical evaluations of the significance of the differences would be possible, but were not pursued within this work.

2.2 Individual Leaf Segmentation

The pipeline to detect individual leaves consists of two main parts, as depicted in figure 3. In the first part of our pipeline we utilize machine learning techniques for edge extraction to resolve the exact contours or borders of leaves in case of touching leaves or leaf overlaps. In the second part we use several image processing steps to reconstruct the detailed leaf shapes, supported by the predicted leaf borders. The predicted leaf counts from the previous section are also used in the leaf segmentation and region merging processes.

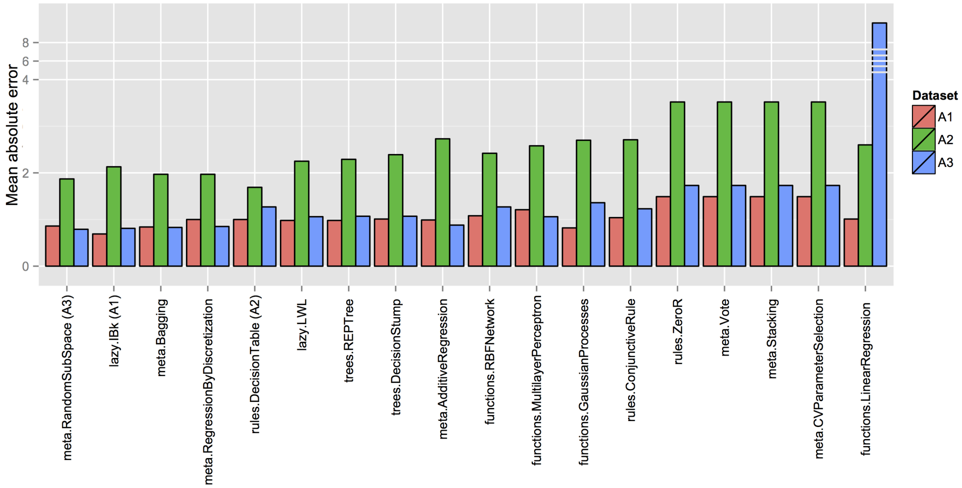

Figure 1: Comparison of prediction results for leaf counts of the A1, A2 and A3 data sets. Calculated are the differences (mean absolute error) of ground truth leaf counts to the prediction results. Table 4 shows the values in detail.

2.2.1 Leaf Border Detection

The exact leaf borders which will be used for the classifier training are extracted from the given ground truth image data (pre-labeled leaf image, see middle part of figure 4). The leaf borders are extracted by performing an edge detection (sobel filter) on the pre-labeled ground truth images. The result are fine lines, representing the leaf edges in the area of leaf-overlap. This result is dilated by a few pixels, as the edge region is somewhat blurry in the input images. By applying the ground truth foreground information two images per plant are generated. One image contains the borders of touching leaves. The second image contains the inner leaf area pixels. These images are used for classifier training. For an optimal training result different features are extracted. The given RGB images are converted into several colour-spaces (CMYK, HCY, HSI, HSL, I1T2T3, Lab, LCHLab, LCHLuv, LSHLab, LSHLuv, Luv, YCbCr, YIQ, YQ1Q2, Yuv, YUV, Yxy), since it is known that the RGB colour-space is often not optimal suited for segmentation tasks. By using the different Weka feature evaluation modes, we determined, that the Luv and Lab colour spaces were in most cases best suited to separate the border pixels from the inner leaf pixels. As the ranking of the channels was not fully consistent and often highly correlated channels were grouped together in the ranking (e.g. several colour channels encode the brightness, only slightly different), we choose the best colour-related channels (Lab.a and Luv.v), which encode opposite colour information (Lab.a red-green and Luv.v blue-yellow), as the basis for the generation of additional channels. Therefore, based on these two channels a further feature extraction is performed utilizing different image- and texture-filters. A Gaussian blur, median and sharpen filter are applied by using ImageJ [16] functions. The texture features are calculated based on the GLCM approach [11]. The median filter uses a mask size of 4 pixels in x and y direction, for the Gaussian blur, sharpen and GLCM texture calculations a 3x3 mask size was chosen. A random forest RF classifier is trained by utilizing the WEKA framework (version 3.6.12). The depth of the trees is set to unlimited and 100 trees have been calculated, random

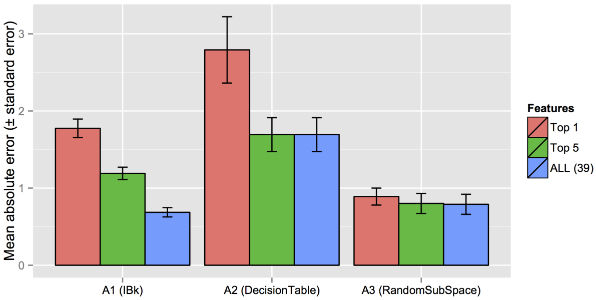

Figure 2: Comparison of prediction results for leaf counts of the A1, A2 and A3 data sets, when using the most important feature (see table 1), the best five, and all features for leaf-count prediction (10-fold cross-validation, shown are mean absolute errors, error bars show standard error based on 10 repetitions).

seed value is '1' (the command line parameters are '-K 0 -I 100 -S 1', default values were used). In a next step, the trained model is used to predict the leaf borders on the testing data. Weka automatically chooses the most likely class, in case of a binary decision (leaf border or inner leaf pixels), the threshold is 0.5. The actual class probability can also be included in the output and transformed into gray-scale images, as shown for illustration in image 3 of figure 4.

2.2.2 Leaf Area Labelling

The leaf border prediction image which is calculated by the first pipeline part and the results from the leaf count estimation are used as input for the individual leaf segmentation. An overview of the used methods can be found in figure 3 (gray box at the bottom). For estimating the positions of the leaf center points a Euclidean distance map is calculated on the predicted leaf edge image. The machine learning model, trained by Weka (for all methods we used the default parameters as provided by Weka version 3.6.12) is used to segment the test data into 'inner leaf areas' and 'leaf border' pixels. In case of touching or overlapping leaves the peaks in the distance map will be at least partially separated by the detected leaf borders. The peak detection gives the positions of the leaf centres. The distance map includes still noise and artefacts. To optimize the peak search, the sensitivity for the peak detection is adjusted using the leaf count results obtained in the leaf count prediction process. The sensitivity is decreased iteratively as long as the detected number of peaks stays larger or is equal to the estimated leaf count. The detected peaks, which approximate the position of the leaf centres, are then used as starting points for a flood-filling like approach. Before the labelling starts, segments are separated by split-lines, which are calculated as detailed in [12]. The flood-filling labelling-approach therefore stops at the split-lines, the shortest line which splits any two regions. The flood filling at this stage just processes the predicted leaf inner areas and will not process the detected leaf borders (see image 4, 5, and 6 in figure 4). In addition, small foreground areas which contain no detected distance-map peaks will initially not be labeled. To complete the labelling of the whole plant (in this case given from the provided foreground masks), unlabeled pixels will be assigned according to the nearest labeled foreground pixel. The nearest labeled pixel is determined according to a shortest path search, which ignores background pixels. Finally, in case more regions are found and

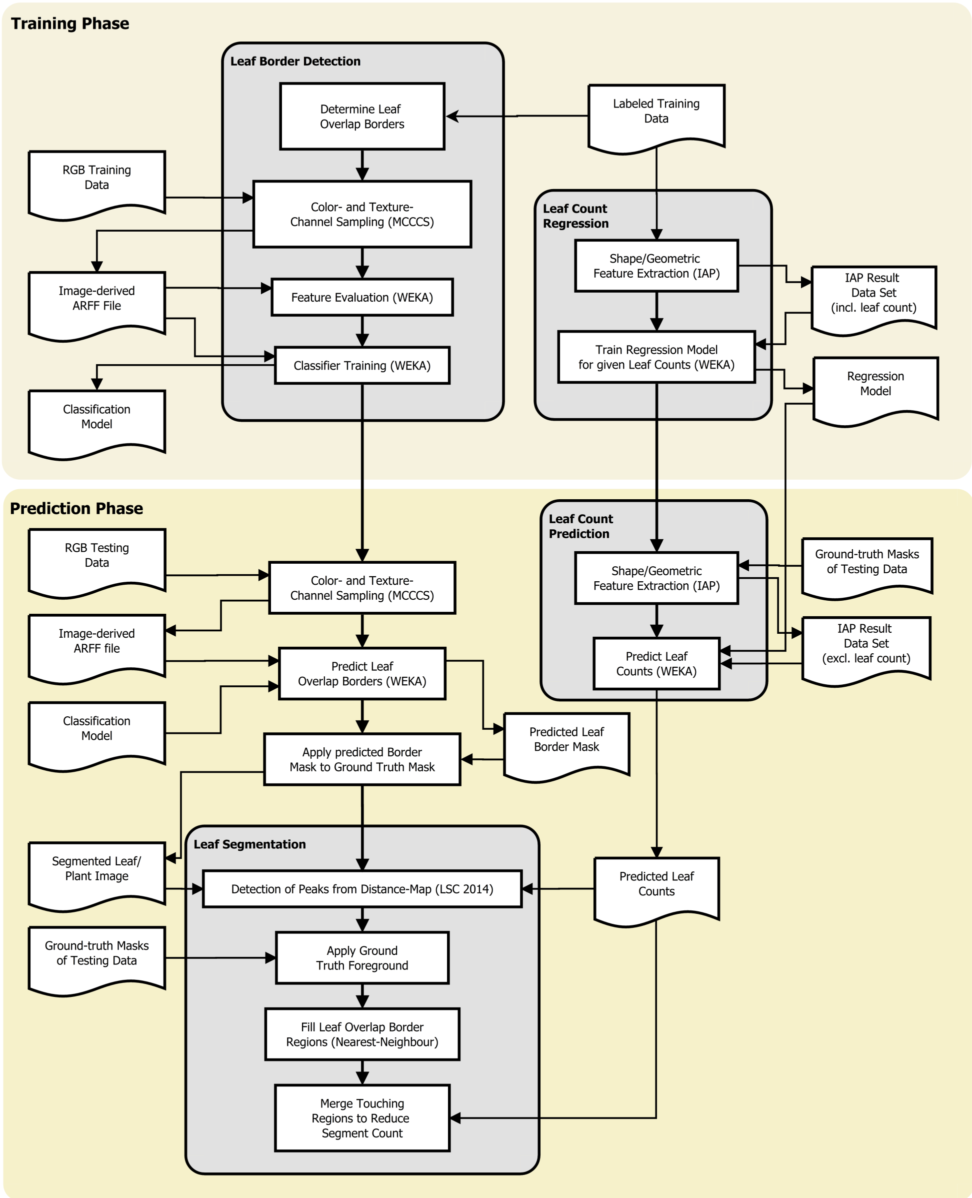

Figure 3: Suggested workflow for leaf count prediction and leaf segmentation. The first part depicts the training phases for the leaf border detection and for the leaf count regression, both utilizing machine learning techniques. The information for the predicted leaf borders and leaf count is used to perform an individual leaf segmentation. Rectangles represent processing steps, the other shapes represent data files.

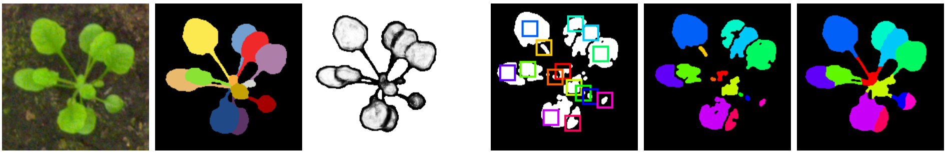

Figure 4: Results for the prediction result for the leaf edge detection and leaf labeling (example image from data set A1, plant 007). (1) input image, (2) ground truth label image, (3) prediction result (black indicates high probability for leaf borders), (4) detected leaf centres, (5) result of performed region filling, (6) labeled missing pixels by nearest neighbour approach and region merging.

labeled than predicted, in each iteration step two adjacent regions are merged using a greedy step-wise approach to maximize the compactness of the merged areas. The merging process will stop if the number of regions equals the estimated leaf count.

3 Results and Discussion

Table 3 shows the error rates of our approach for predicting the leaf counts of the given training data sets. Calculation based on the classifiers which give the best performance (see table 4).

Table 3: Error rates for the leaf count prediction for the training set.

	A1	A2	A3
Correlation coefficient	0.84	0.79	0.79
Mean absolute error	0.69	1.69	0.79
Root mean squared error	0.96	1.95	0.93
Relative absolute error	47.66	57.90	52.47
Root relative squared error	55.95	56.88	56.20
Total Number of Instances	128	31	27

Overall, the quality of the leaf count estimation has been significantly improved compared to the LSC 2014 results. The geometric features give a good basis and could eventually be further improved by other features such as colour and texture related or additional geometric features. Here, it would probably be sensible, to extend the list of features by adding the results of other phenotypic image analysis tools. We expect that our approach will work for other species too, but the actual trained models will most likely not be applicable for diverse phenotypes. For different phenotypes, the influential phenotypic properties will also most likely vary. Comprehensive training data will therefore be required for creating leaf-count prediction models for plants with different phenotypic appearance. Such training data would also be useful to make it possible to work on generalization of the presented methods.

The processing pipeline for leaf border detection and leaf labelling is designed by utilizing Bash scripts and different command line based image processing commands, utilizing our MCCCS [1] framework. This system also connects machine learning techniques with image analysis approaches, by converting images to ARFF files, which can be processed by Weka, and by converting result ARFF files into grayscale or RGB images. Image processing utilizes IAP [7] and ImageJ [16] as libraries for this system.

[1]mcccs.sourceforge.net *(unpublished)*

Table 4: Comparison of classifier performance (shown are algorithms, ordered by the average mean absolute error for A1, A2, A3, default algorithm parameters in Weka were unchanged from default. These values are diagrammed in figure 1.

Algorithm	A1	A2	A3	Average Mean Absolute Error
meta.RandomSubSpace	0.86	1.87	**0.79**	1.17
lazy.IBk	**0.69**	2.13	0.81	1.21
meta.Bagging	0.84	1.97	0.83	1.21
meta.RegressionByDiscretization	1.00	1.97	0.85	1.27
rules.DecisionTable	1.00	**1.69**	1.27	1.32
lazy.LWL	0.98	2.25	1.06	1.43
trees.REPTree	0.98	2.29	1.07	1.45
trees.DecisionStump	1.01	2.39	1.07	1.49
meta.AdditiveRegression	0.99	2.73	0.88	1.53
functions.RBFNetwork	1.08	2.42	1.27	1.59
functions.MultilayerPerceptron	1.21	2.58	1.06	1.62
functions.GaussianProcesses	0.82	2.70	1.36	1.63
rules.ConjunctiveRule	1.04	2.71	1.23	1.66
rules.ZeroR	1.49	3.52	1.73	2.25
meta.Vote	1.49	3.52	1.73	2.25
meta.Stacking	1.49	3.52	1.73	2.25
meta.CVParameterSelection	1.49	3.52	1.73	2.25
functions.LinearRegression	1.01	2.60	8.42	4.01

The analysis of the given data sets was performed in two steps in respect to the pipeline design. First a classifier training for the leaf border detection was performed. Here, randomly 500 pixels were selected and used per class (two classes, class 1 'border pixel', class 2 'inner leaf area pixel') and training image. After completion of the leaf border detection, the second part, the individual leaf segmentation was performed. Table 5 shows the average computation times for each training data set. The calculation time is mostly influenced by the calculation of the GLCM texture features, and the blur- and median-filter calculations. Dataset A3 is of higher resolution (2448 by 2048 px) in comparison to A1 and A2 (500 by 530 and 530 by 565 px), also explaining the longer computation times for the A3 data set. The calculations for the prediction phase take nearly the same time per image, as in this phase the same number of colour- and texture channels needs to be computed and applying the prediction model and performing the labelling is completed in a few seconds. Figure 5 shows an example result

Table 5: Processing time per image for the training data in minutes. Numbers in the heading represent the number of images for each data set.

	A1 (128)	A2 (33)	A3 (27)
per plant	~28	~33	~120
classifier training	~90	~6	~15

compared to the LSC 2014 pipeline [12]. The new approach is an improvement compared to the old approach, as seen in the region of leaf overlaps (top right part of plant). In the illustrated example all leaf overlaps could be resolved. Unfortunately the merging approach is not optimal in case of failed detection of leaf centres. Especially the leaf detection in the centre of the plant is a challenging problem. This leads to remaining artefacts, split leaves will not be merged, if the estimated leaf number is already reached (as can be seen at the small leaf at the right-bottom position in figure 5, right image). Here, meaningful peaks in the plants centre are missing. Detection of wrong regions caused by noise (leaf-veins),

Figure 5: Example results (data set A1, plant 007), left - given ground truth label image, middle - LSC 2014 result, right - LSC 2015 result.

especially in the A3 data set. Here, a better pre-processing could remove noise and resolve the wrongly detected parts.

Table 6 shows the segmentation results for the given training and the independently evaluated testing data sets. Compared to the results of the LSC 2014 challenge [12] our new approach shows a better performance on all training data sets. The testing results showed a more diverse result. The slightly decreased performance in case of the A3 data set could be explained by the different resolution and noise appearance, compared to the A1 and A2 data sets which are quite similar in case of image quality and resolution.

Table 6: Results for the training (first) and testing data sets (second row set).

	BestDice [%]	FGBGDice [%]	AbsDiffFGLabels	DiffFGLabels
A1	82.4 (±5.0)	100.0 (±0.0)	0.1 (±0.3)	0.1 (±0.4)
A2	83.5 (±9.7)	100.0 (±0.0)	0.3 (±0.7)	-0.2 (±0.7)
A3	71.6 (±21.8)	96.3 (±19.2)	0.6 (±1.3)	0.2 (±1.4)
all	81.0 (±10.7)	99.5 (±7.3)	0.2 (±0.6)	0.0 (±0.7)
A1	80.9 (±6.3)	100.0 (±0.0)	0.8 (±1.0)	-0.2 (±1.2)
A2	78.6 (±7.7)	100.0 (±0.0)	1.7 (±1.5)	-0.6 (±2.2)
A3	64.5 (±16.1)	100.0 (±0.0)	1.2 (±1.2)	-0.3 (±1.7)
all	71.3 (±15.1)	100.0 (±0.0)	1.1 (±1.2)	-0.3 (±1.6)

Acknowledgments

This work was supported by the Leibniz Institute of Plant Genetics and Crop Plant Research and by the German Plant Phenotyping Network which is funded by the German Federal Ministry of Education and Research (grant no. 031A053B).

References

[1] Bernhard Biskup, Hanno Scharr, Ulrich Schurr, and Uwe Rascher. A stereo imaging system for measuring structural parameters of plant canopies. *Plant, Cell and Environment*, 30(10):1299–1308, 2007. ISSN 1365-3040. doi: 10.1111/j.1365-3040.2007.01702.x. URL http://dx.doi.org/10.1111/j.1365-3040.2007.01702.x.

[2] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. 2014.

[3] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[4] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998. ISSN 0162-8828. URL http://citeseer.ist.psu.edu/ho98random.html.

[5] Geoffrey Holmes, Mark Hall, and Eibe Frank. Generating rule sets from model trees. In *Twelfth Australian Joint Conference on Artificial Intelligence*, pages 1–12. Springer, 1999.

[6] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256, 1992.

[7] Christian Klukas, Dijun Chen, and Jean-Michel Pape. Integrated analysis platform: an open-source information system for high-throughput plant phenotyping. *Plant physiology*, 165(2):506–518, 2014.

[8] Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *Machine Learning: ECML-94*, pages 171–182. Springer, 1994.

[9] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012*, pages 502–516. Springer, 2012.

[10] Massimo Minervini, Mohammed M Abdelsamea, and Sotirios A Tsaftaris. Image-based plant phenotyping with incremental learning and active contours. *Ecological Informatics*, 23:35–48, 2014.

[11] P Mohanaiah, P Sathyanarayana, and L GuruKumar. Image texture feature extraction using glcm approach. *International Journal of Scientific and Research Publications*, 3 (5):1, 2013.

[12] Jean-Michel Pape and Christian Klukas. 3-d histogram-based segmentation and leaf detection for rosette plants. In Lourdes Agapito, Michael M. Bronstein, and Carsten Rother, editors, *Computer Vision - ECCV 2014 Workshops*, volume 8928 of *Lecture Notes in Computer Science*, pages 61–74. Springer International Publishing, 2015.

[13] Ross J. Quinlan. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore, 1992. World Scientific.

[14] Marko Robnik-Šikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML 97)*, pages 296–304, 1997.

[15] Hanno Scharr, Massimo Minervini, Andreas Fischbach, and Sotirios A. Tsaftaris. Annotated image datasets of rosette plants. Technical report, 2014. URL http://juser.fz-juelich.de/record/154525.

[16] Caroline A Schneider, Wayne S Rasband, and Kevin W Eliceiri. Nih image to imagej: 25 years of image analysis. *Nature methods*, 9(7):671–675, 2012.

[17] Y. Wang and I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.

[18] Xi Yin, Xiaoming Liu, Jin Chen, and David M Kramer. Multi-leaf tracking from fluorescence plant videos. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 408–412. IEEE, 2014.