K-means and Eigen-colors based plant image segmentation (kmSeg) v.0.2 – Quick guide

Michael Henke, Evgeny Gladilin
e-mail: {henke, gladilin}@ipk-gatersleben.de

Image Analysis Group, Leibniz Institute of Plant Genetics and Crop Plant Research (IPK Gatersleben), OT Gatersleben, Corrensstraße 3
06466 Seeland, Germany

February 8, 2021

Example of plant image segmentation using kmSeg. From left to right: pre-segmented image of a maize shoot including plant and non-plant regions, false-color visualization of k-means color clusters, segmented image and k-means clusters of plant regions.

Contents

© 2021, Image Analysis Group, IPK Gatersleben

1 Introduction

The kmSeg tool was developed to speed up generation of ground-truth data for evaluation of multimodal image registration and segmentation algorithms within the scope of our previously published research studies [1, 2, 3].

Efficient tools for accurate image segmentation are highly demanded in diverse biomedical applications dealing with quantitative image analysis including high-throughput plant phenotyping. Generation of accurately annotated ground-truth data is the first indispensable step of any supervised learning approach to automated image analysis. In the absence of appropriate tools, manual segmentation of large, noisy and optically heterogeneous images represents a tedious and time-consuming task, which constitutes the major bottle neck for application of advanced machine and deep learning methods. Here, we present a software solution for semi-automated segmentation of plant images which is based on manual classification of a low number of image regions automatically selected using k-means clustering of image Eigen-colors. Primarily developed for application to plant image analysis this tool is not limited to a particular image content or data modality and can be applied for rapid segmentation of arbitrary image data [4].

1.1 Key Features

Image segmentation in the kmSeg tool is performed in two steps. First, unsupervised k-means clustering of Eigen-colors is applied to automatically identify a predefined number (N) of image regions with similar colors. In the next step, the user manually assigns plant and non-plant labels to N pre-segmented image regions by visual inspection of their average colors and/or other numerical descriptors. Thereby, the user can adjust diverse algorithmic parameters to obtain an optimal segmentation result.

2 Quick Start

2.1 How to install?

After unpacking the zip archive following three folders will be generated:

```
kmSegTool
├── exampleImages
└── quickGuide
```

The *kmSegTool* folder contains the pre-compiled executable of the computer program, a readme and a license file. Please, read both text files carefully before starting the program. The *exampleImages* folder contains a set of example images as described in Sec. 3, and the *quickGuide* folder contains a copy of this file.

2.2 How to run?

The kmSeg tool comes compiled in two versions, one for Linux- and one for Windows-based operation systems, respectively. To run the program the user has to install the MATLAB Runtime Environment. Since the kmSeg tool was developed, tested and compiled under MATLAB 2020a, we recommend to install exactly the same version, i.e. MCR 2020a, which can be downloaded from the official MATLAB side Install and Configure the MATLAB Runtime.

© 2021, Image Analysis Group, IPK Gatersleben

2.2.1 Linux

Under Linux-based operation systems one has to open a terminal and switch to the folder which contains the kmSeg tool . Then type

```
./run_kmSeg.sh /path/to/your/MATLAB/Runtime/v98
```

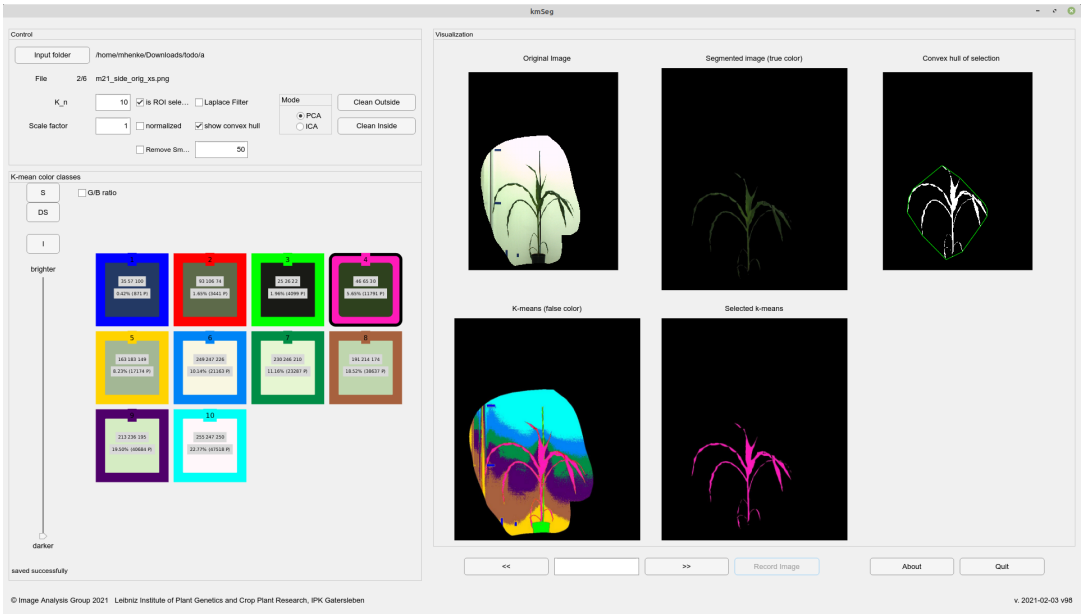
where */path/to/your/MATLAB/Runtime/v98* specifies the path to the locally installed MATLAB Runtime Environment (version 2020a).

2.2.2 Windows

To run the program under Windows double-click on the icon of the provided executable in the Windows file explorer or start it with its name from the command line.

2.3 The Interface Layout

The major elements of the software interface include a **Control-area** at the upper left part, the **k-means class-area** below, and on the right of the GUI a **Visualization-area** as shown below.



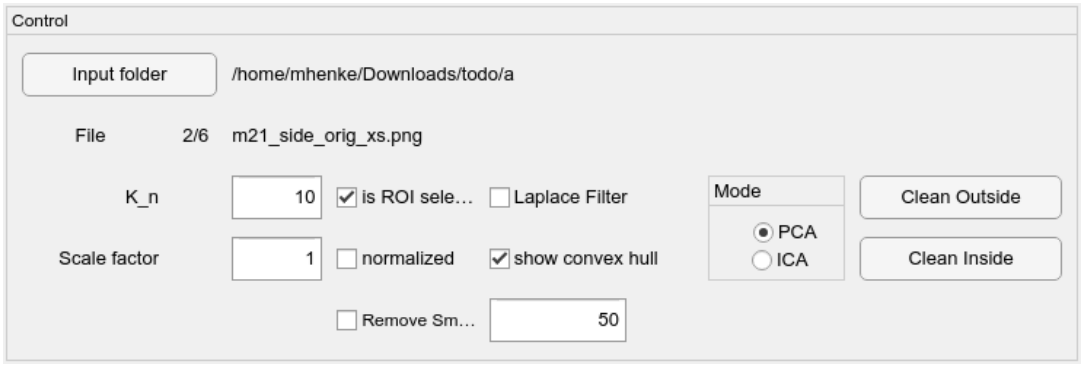
Below the **k-mean class-area**, a **status line** provides information about the current state of the image processing and the latest user interactions. This is particular helpful as a feedback during time consuming calculations.

2.4 First Steps

After running the program, the user has to select a folder with images to be processed.
Note: the current version only supports *png* and *jpg* files.

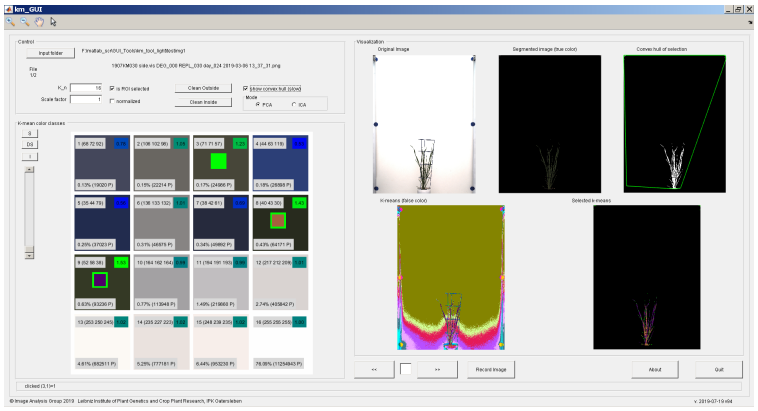
The selected folder, the number of files and the current file name are shown in the upper part of the **Control-area**.

© 2021, Image Analysis Group, IPK Gatersleben

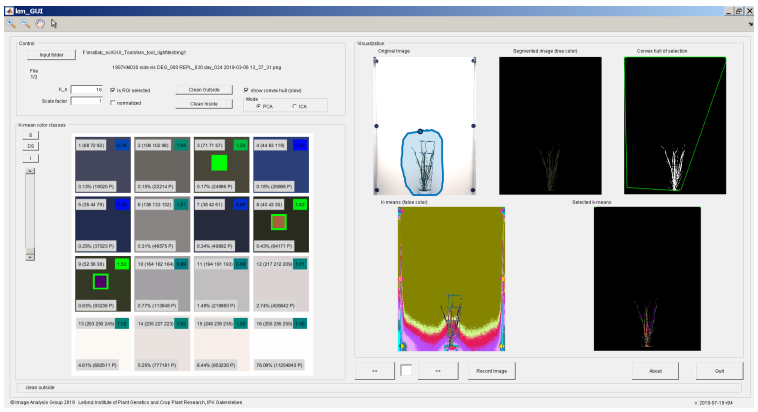


Once images are found and successfully imported into the program, the calculation immediately starts. K-means calculation is always performed using the actual set of algorithmic parameters that can be adjusted by the user.

To remove parts of the input image two functions are provided: *Clean Inside* and *Clean Outside*. Both will allow the user to indicate regions by freehand drawing that has to be cleaned, ether inside or outside.

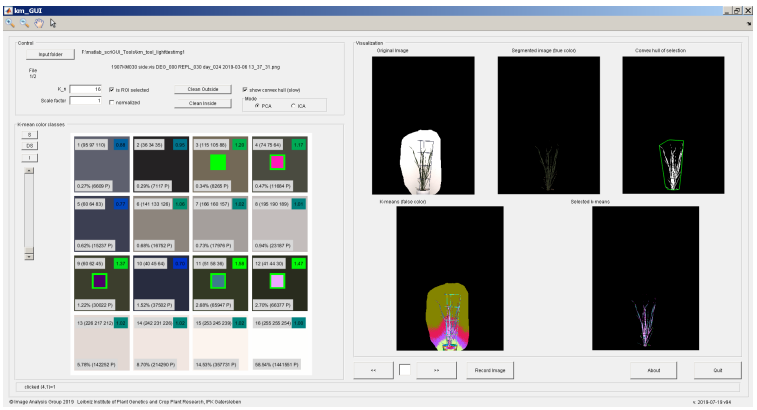


original wheat image



partially cleaned image

© 2021, Image Analysis Group, IPK Gatersleben



final segmentation

K_n Number of k-means regions in color-space (here called: classes).

Scale factor Image scale factor, which can be used to decrease the image size and to speed up the calculation time.

is ROI selected flag, when activated it indicates that the selected k-means classes belong to the region of interest (ROI), otherwise the selected classes represent the background.

normalized is an optional parameter of color pre-processing which is applied prior to k-means clustering. When normalized is selected, colors are transformed to normalized values $(I - \min(I))/(\max(I) - \min(I))$.

Laplace Filter flag activates structure preserving image smoothing prior to k-means clustering. This helps to suppresses statistical noise and to make color distribution more compact. However, the application of the Laplace filter can be quite time consuming.

show convex hull flag activates the visualization of the convex hull of the selected k-means image regions in the **Visualization-area**.

Mode-Box Here one can switch between PCA and ICA as Eigen-color calculation method.

Clean Outside Tool to manually remove unwanted regions from the image. Everything outside the selected area will be replaced by the black color.

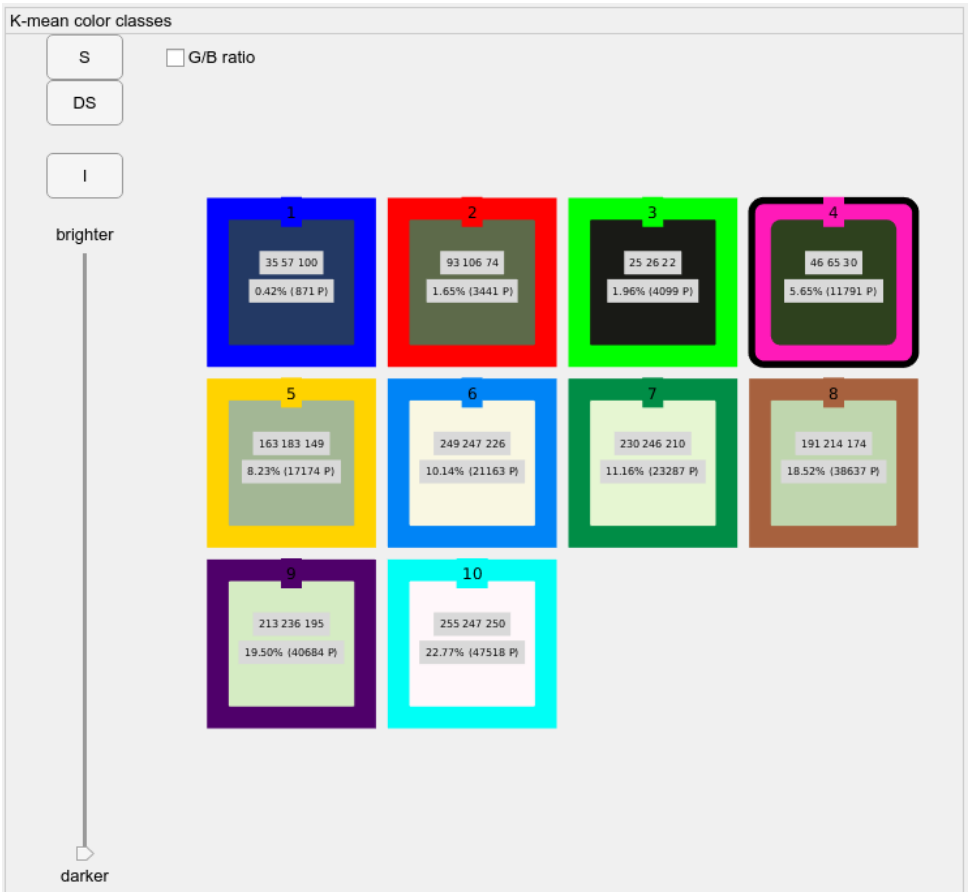
Clean Inside Tool to manually remove unwanted regions from the image. Everything inside the selected area will be replaced by black color.

Remove Small Objects Flag to indicate if small objects should be removed in post-processing. The maximal size of objects to be removed can be set within the edit field right to it.

Most of the described parameter (*K_n, Scale factor, normalized, Laplace Filter, Mode, Clean Outside, Clean inside*) will automatically force a re-calculation of the k-means classes after every change.

Since kmSeg tool performs calculation of non-black colors, cleaning of non-relevant outside or inside image regions using the *Clean* functions will substantially speed up the calculation of k-means classes and the overall segmentation procedure.

© 2021, Image Analysis Group, IPK Gatersleben



The calculated k-means class (K_n) are shown in the **K-means classes-area**, providing the following settings:

S select all k-means classes

DS de-select all k-means classes

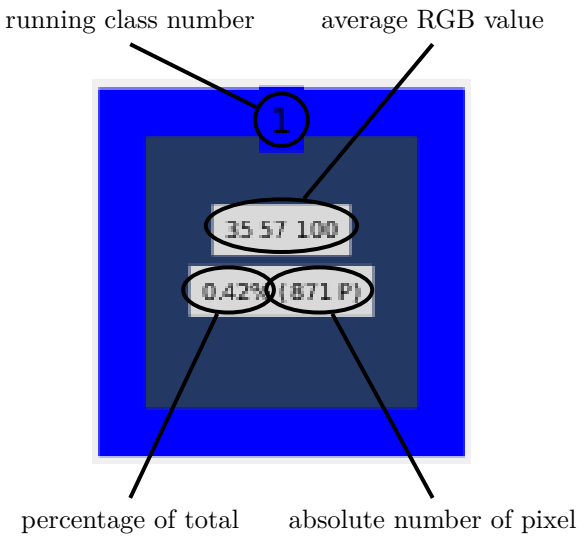
I invert the current selection of k-means classes

Brightness increases the brightness of the visualized image. Note: this affects only the visualization and not the image used for calculation!

G/R ratio flag, when activated the green/red ratio of all k-means classes are shown within each class tiller.

Each k-means class is represented by one tiller containing basic information of each class. The main color of the tiller shows the average color of all pixel within this class.

© 2021, Image Analysis Group, IPK Gatersleben



By clicking on one of the tiller, one can select/de-select the corresponding k-means class. After the selection has changed, the visualization in the **Visualization-area** will be updated automatically. The **Visualization-area** shows the following five images: the original image, segmented k-means classes in true-color, the convex hull of the segmented k-means classes as black and white image, the k-means classes of the original image in false color (each color represents one class), and the selected k-means classes in false-color.

After the user made a selection of k-means classes, the results can be stored by clicking on the *Record image* button in the lower part of the **Visualization-area**. Currently, for each recorded image the following files are stored:

<image file name>.mat a MATLAB workspace file, containing all parameter setting and all selection information, see Appendix for details.

<image file name>.mat a CSV data base file, containing all extracted features for the current image.

<image file name>.txt an ASCII file, containing information and statistics about plant and non-plant color-classes selected by the user, see Appendix for details.

<image file name>_km_features.png a ”feature image” as visualization of some of the extracted features.

<image file name>_km_all.<image suffix> all k-means classes in false-color where each color stands for one particular class.

<image file name>_km_sel.<image suffix> all selected k-means classes in false-color where each color stands for one particular class.

<image file name>_km_sel_TC.<image suffix> all selected k-means classes in true-color.

<image file name>_km_smoothed.<image suffix> the for calculation used input image.

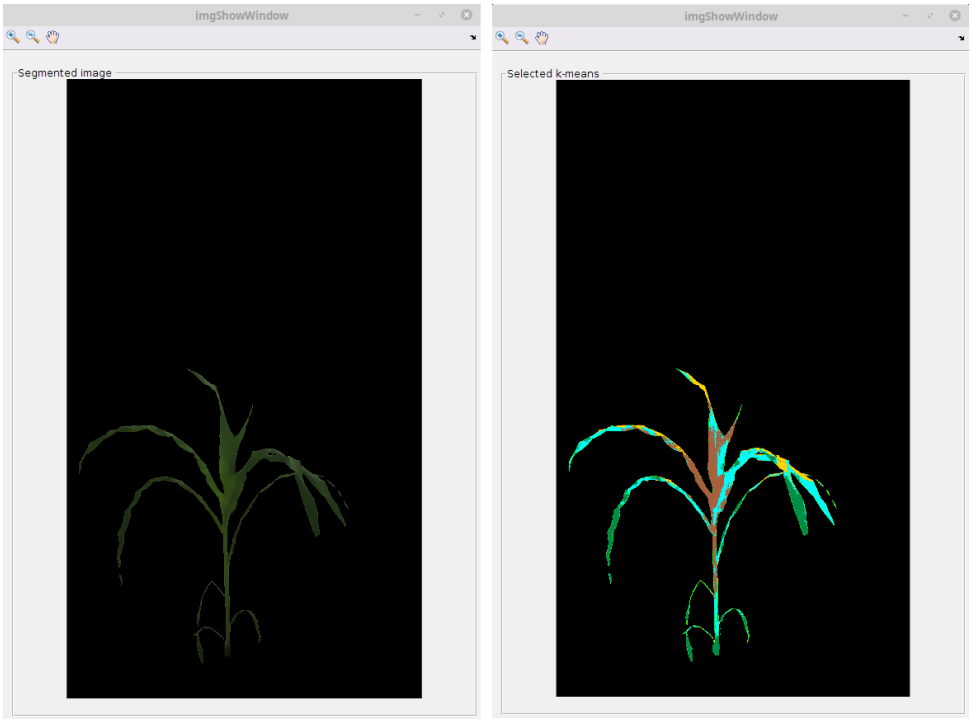
<image file name>_km_mask.png a binary mask of the segmented image.

© 2021, Image Analysis Group, IPK Gatersleben

By using the « or » button, the previous or the next image can be selected. Instead of clicking several times, the user can directly go to wanted image by entering the number of the wanted image in the list of all image in the selected folder. The *About* button provides general information about the authors and copyright statements. With the *Quit* button the kmSeg tool can be terminated.

When exiting the kmSeg tool tool, all CSV files within the output folder are collected and concatenated to a single ”allInOne.csv” file.
For a better visualization of the selection, two additional windows are opened when the kmSeg tool starts. One shows the segmented classes in true colors and the other one the false-color visualization of the selected k-means classes. These windows are particular helpful when small structures are to be selected. For a better user experience, we suggest to use a multi-screen system.

© 2021, Image Analysis Group, IPK Gatersleben



3 Provided example data

The kmSeg tool comes with a small set of example images consisting or young and mid-age arabidopsis, wheat and maize images for visual light in top and side view.

Arabidopsis side image

Barley side image

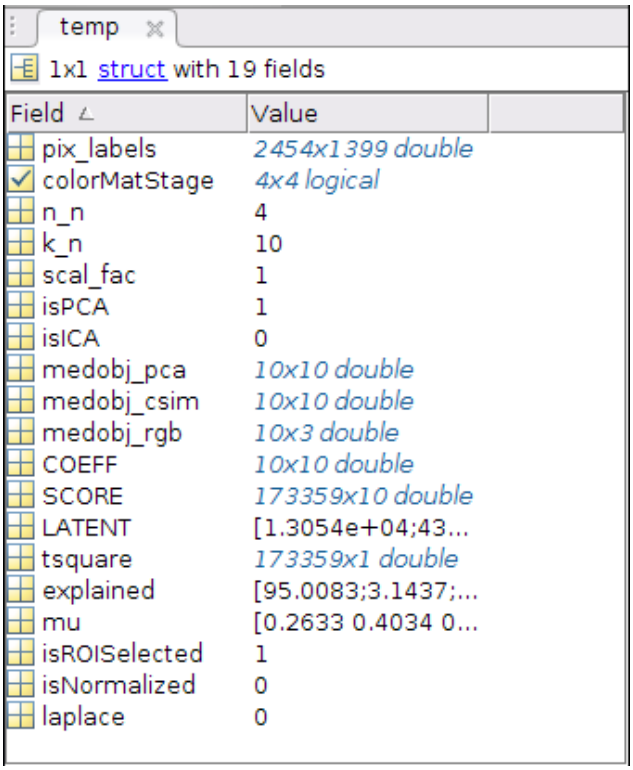
The images can be found at the *exampleImages* folder (see Sec. 2.1.

© 2021, Image Analysis Group, IPK Gatersleben

A Output file description

A.1 MATLAB file

The from the kmSeg tool for each images stored MATLAB file contains one record variable called *temp*. This record contains 19 fields as shown below:



pixel labels
color-class (de)selected
size of visualization matrix
number of k-means classes
image scale factor
flag, is PCA used
flag, is ICA used
medians of class PCA scores
median of class color space values
median of class RGB values
MATLAB PCA coefficients
MATLAB PCA scores
MATLAB PCA latency values
MATLAB PCA tsquare values
MATLAB PCA exaplained values
MATLAB PCA mu values
flag, is ROI selected
flag, is normalized
flag, use Laplace filter

Some of the above workspace variables are help-variables that are essentially stored for error debugging purposes and, thus, not of interest for users.

A.2 ASCII file

In addition to the above described mat-files, a plain ASCII file is stored for each segmented image. This file contains basic information about assigned of k-means classes to plant and non-plant regions (i.e., image segmentation) as well as statistics of segmented color regions.
A typical ASCII output file will include the following entries:

Variable names		Examples of variable values:
km_n:	number of km-classes	`km_n: 10`
plant_classes:	sequence of km-class IDs related to the plant	`plant_classes: 1 2`
nonplant_classes:	sequence of km-class IDs not related to the plant	`nonplant_classes: 3 4 5 6 7 8 9 10`
class_i_color:	mean R G B color values of km-classes i	`class_1_color: 63 104 54`
class_i_color:	number of pixels within km-class i	`class_1_npixels: 3048`
…		…

© 2021, Image Analysis Group, IPK Gatersleben

B Funding

The research was partially funded by the German Plant Phenotyping Network (DPPN) which is funded by the the German Federal Ministry of Education and Research (BMBF) under the project identification number 031A053. All support is gratefully acknowledged.

C Acknowledgments

We would like to thank Kerstin Neumann (wheat) from the Research Group Genome Diversity, Astrid Junker (maize) from the Research Group Acclimation Dynamics and Phenotyping and Mohammad-Reza Hajirezaei (arabidopsis) from the Molecular Plant Nutrition group of the IPK Gatersleben for providing the plant data material.

D References

[1] Henke M, Junker A, Neumann K, Altmann T, Gladilin E (2019) Comparison of Feature Point Detectors for multi-modal Image Registration in Plant Phenotyping; accepted by *PLOS ONE*

[2] Henke M, Junker A, Neumann K, Altmann T, Gladilin E (2019) Comparison and extension of three methods for automated registration of multi-modal plant images; *Plant Methods*, 15:1, 44, doi: 10.1186/s13007-019-0426-8

[3] Henke M, Junker A, Neumann K, Altmann T, Gladilin E (2018) Automated alignment of multi-modal plant images using extended phase correlation approach; *Frontiers in Plant Science-Plant Physiology*, 9, 1-10, doi: 10.3389/fpls.2018.01519

[4] Henke M, Gladilin, E (2019) Efficient semi-automated annotation of plant images using k-means clustering of Eigen-colors, *(unpublished)*

E Terms of use

1. The kmSeg tool and the example image data are distributed for non-commercial usage WITHOUT ANY WARRANTY under the terms described in the EULA license. See the included *EULA.txt* file for details.

2. The user manual is intellectual property of the Image Analysis Group of the IPK Gatersleben. The user may download and use the tool and information available on our web site.

Copyright

© 2021, Image Analysis Group, Leibniz Institute of Plant Genetics and Crop Plant Research (IPK Gatersleben), OT Gatersleben, Corrensstraße 3, 06466 Seeland, Germany