

Article

Semi-Automated Ground Truth Segmentation and Phenotyping of Plant Structures Using k-Means Clustering of Eigen-Colors (kmSeg)

Michael Henke ^{1,2} , Kerstin Neumann ¹ , Thomas Altmann ¹  and Evgeny Gladilin ^{1,*} 

¹ Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), D-06466 Seeland, Germany; mhenke@uni-goettingen.de (M.H.); neumannk@ipk-gatersleben.de (K.N.); altmann@ipk-gatersleben.de (T.A.)

² Plant Sciences Core Facility, CEITEC—Central European Institute of Technology, Masaryk University, 62500 Brno, Czech Republic

* Correspondence: gladilin@ipk-gatersleben.de

Abstract: Background. Efficient analysis of large image data produced in greenhouse phenotyping experiments is often challenged by a large variability of optical plant and background appearance which requires advanced classification model methods and reliable ground truth data for their training. In the absence of appropriate computational tools, generation of ground truth data has to be performed manually, which represents a time-consuming task. **Methods.** Here, we present a efficient GUI-based software solution which reduces the task of plant image segmentation to manual annotation of a small number of image regions automatically pre-segmented using k-means clustering of Eigen-colors (kmSeg). **Results.** Our experimental results show that in contrast to other supervised clustering techniques k-means enables a computationally efficient pre-segmentation of large plant images in their original resolution. Thereby, the binary segmentation of plant images in fore- and background regions is performed within a few minutes with the average accuracy of 96–99% validated by a direct comparison with ground truth data. **Conclusion.** Primarily developed for efficient ground truth segmentation and phenotyping of greenhouse-grown plants, the kmSeg tool can be applied for efficient labeling and quantitative analysis of arbitrary images exhibiting distinctive differences between colors of fore- and background structures.

Keywords: plant image segmentation; plant phenotyping; ground truth data generation; color spaces; principle component analysis; unsupervised data clustering



Citation: Henke, M.; Neumann, K.; Altmann, T.; Gladilin, E. Semi-Automated Ground Truth Segmentation and Phenotyping of Plant Structures Using k-Means Clustering of Eigen-Colors (kmSeg). *Agriculture* **2021**, *11*, 1098. <https://doi.org/10.3390/agriculture11111098>

Academic Editors: Maciej Zaborowicz and Dawid Wojcieszak

Received: 5 September 2021

Accepted: 30 October 2021

Published: 4 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the introduction of high-throughput plant phenotyping facilities, efficient analysis of large multimodal image data turned into focus of quantitative plant research [1]. Typical goals of high-throughput plant image analysis include detection, counting or pixel-wise segmentation of targeted plant structures (e.g., whole shoots, fruits, spikes, etc.) in field or greenhouse environments followed by their quantitative characterization in terms of morphological, developmental and/or functional traits. Especially the pixel-wise segmentation represents a critical step of plant image analysis, since the accuracy and reliability of some phenotypic traits (e.g., linear plant dimensions) are highly prone to smallest errors of image segmentation. Due to a number of natural and technical factors, segmentation of plant structures from background image regions represents a challenging task. Inhomogeneous illumination, shadows, occlusions, reflections and dynamic optical appearance of growing plants complicate definition of invariant criteria for detection of different parts (e.g., leaves, flowers, fruits, spikes) of different plant types (e.g., arabidopsis, maize, wheat) at different developmental stages (e.g., juvenile, adult) in different views (e.g., top or multiple side views) acquired in different image modalities (e.g., visible light, fluorescence, near-infrared) [2]. Next generation approaches to analyzing plant images

rely on pre-trained algorithms and, in particular, deep learning models for classification of plant and non-plant image pixels or image regions [3–7]. The critical bottle neck of all supervised and, in particular, novel deep learning techniques is availability of sufficiently large amount of accurately annotated ‘ground truth’ image data for reliable training of classification-segmentation models. In a number of previous works, exemplary datasets of manually annotated images of different plant species were published [8,9]. However, these exemplary ground truth images cannot be generalized for analysis of images of other plant types and views acquired with other phenotyping platforms. A number of tools for manual annotation and labeling of images have been presented in previous works. The predominant majority of these tools including LabelMe [10], AISO [11], Ratsnake [12], LabelImg [13], ImageTagger [14], VIA [15], FreeLabel [16] are rather tailored to labeling object bounding boxes and rely on conventional methods such as intensity thresholding, region growing and/or propagation, as well as polygon/contour based masking of regions of interest (ROI) that are not suitable for pixel-wise segmentation of geometrically and optically complex plant structures. De Vylder et al. [17] and Minervini et al. [18] presented tangible approaches to supervised segmentation of rosette plants. Early attempts at color-based image segmentation using simple thresholding were done by Granier et al. [19] in the GROWSCREEN tool developed for analysis of rosette plants. A general solution for accurate and efficient segmentation of arbitrary plant species is, however, missing. Meanwhile, a number of commercial AI assisted online platforms for image labeling and segmentation such as for example [20,21] is known. However, usage of these novel third-party solutions is not always feasible either because of missing evidence for their suitability/accuracy by application to a given phenotyping task, concerns with data sharing and/or additional costs associated with the usage of commercial platforms.

A particular difficulty of plant image segmentation consists of variable optical appearance of dynamically developing plant structures. Depending on particular plant phenotype, developmental stage and/or environmental conditions plants can exhibit different colors and intensities that can partially overlap with optical characteristics of non-plant (background) structures. Low contrast between plant and non-plant regions especially in low-intensity image regions (e.g., shadows, occlusions) compromise performance of conventional image segmentation tools based on thresholding, region growing or gradient/edge detection. In a number of previous works, transformation of plant images from original RGB to alternative color spaces (e.g., HSV, CIELAB) was reported to be advantageous for separating chlorophyll containing plant from chlorophyll-free non-plant structures in several previous works [22–24]. However, in view of high variability of optical setups and plant phenotypes, definition of universal criteria (e.g., color/intensity bounds) for accurate plant image segmentation is not feasible.

To overcome limitations of existing approaches to accurate generation of ground truth data for pixel-wise plant segmentation and phenotyping, here we developed a stand-alone GUI-based tool which enables efficient semi-automated labeling and geometrical editing (i.e., masking, cleaning, etc.) of complex optical scenes using unsupervised clustering of image color spaces. In order to enable a ‘nearly real-time’ processing of images of the typical size of several megapixels (i.e., $n > 1 \times 10^6$), unsupervised clustering of image pixels in color spaces was performed using k-means which on one hand is known to be faster than other clustering algorithms such as, for example, spectral or hierarchical clustering [25]. On the other hand k-means turned out to be efficient and sufficiently accurate for annotation of visible light and fluorescence images of greenhouse cultured plants that were in primary focus of this work. Jansen et al. [26] used threshold-based approach to segment fluorescence images of arabidopsis plants. We show that using this approach semi-automated labeling of optically complex plant phenotyping scenes can be performed with just a few mouse clicks by assigning pre-segmented color classes/regions to either plant or non-plant categories. By avoiding manual drawing and pixel-wised region labeling, the k-means assisted image segmentation tool (kmSeg) enables biologists

to rapidly perform segmentation and phenotyping of a large amount of arbitrary plant images with the minimum user-computer interaction.

2. Methods

2.1. Image Data

The kmSeg tool was primarily developed for ground truth segmentation of visible light (VIS) and fluorescence (FLU) images of maize, wheat and arabidopsis shoots acquired from greenhouse phenotyping experiments using LemnaTec-Scanalyzer3D high-throughput phenotyping platforms (LemnaTec GmbH, Aachen, Germany). Figure 1 shows examples of top- and side-view images of maize, wheat and arabidopsis shoots acquired from three different screening platforms for large, mid-size and small plant screening.

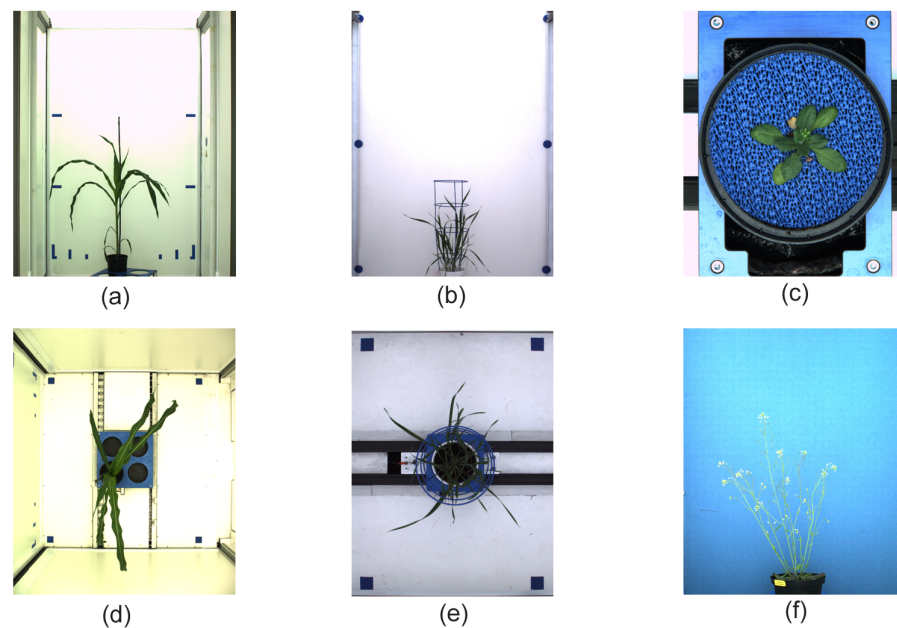


Figure 1. Examples of side-view (upper row) and top-view (bottom row) images of maize (a,d), wheat (b,e) and arabidopsis (c,f) plants.

Furthermore, top-view arabidopsis and tobacco images from the A1, A2 and A3 datasets published in [8] were used in this work for validation of the kmSeg performance, see Figure 2.

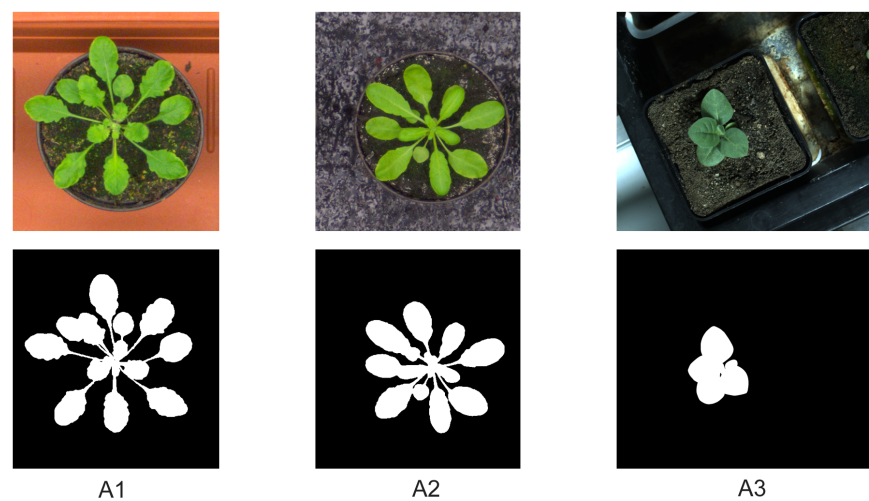


Figure 2. Examples of original (top row) and binary segmented (bottom row) top-view images of arabidopsis (A1,A2), and tobacco (A3) plants from [8].

2.2. Image Pre-Processing and Color-Space Transformations

The goal of image pre-processing is to make representation of fore- and background image structures in color spaces topologically more suitable for subsequent clustering. Straightforward clustering of plant images is often hampered by vicinity of plant and background colors in the original RGB color space. To improve separability of plant and non-plant image regions following pre-processing steps are applied.

2.2.1. Image Smoothing

Structure-preserving Laplacian smoothing (MATLAB built-in function: *locallapfilt*) is optionally used to reduce heterogeneity and noise to make representation of plant and background structures more distinguishable.

2.2.2. Color Space Transformation

The main disadvantage of the RGB color space for automated image segmentation is that luminosity and saturation are lumped together within the color definition what makes it a non-ideal color space especially considering shadowed image regions where the small topological distances are hard to separate. Alternative color spaces such as HSV or CIELAB are typically more suitable for these task. Even when they do not provide more information than original RGB space, the different color representation often allows a better automated segmentation. To decorrelate plant and background colors, transformation of images from RGB to special color spaces is applied. In this work, RGB plant images were transformed to a 10-dimensional color spaces including HSV (3x), CIELAB (3x) and CMYK (4x) color representations that enable topologically more advantageous organization of typical plant and non-plant colors, see Figure 3.

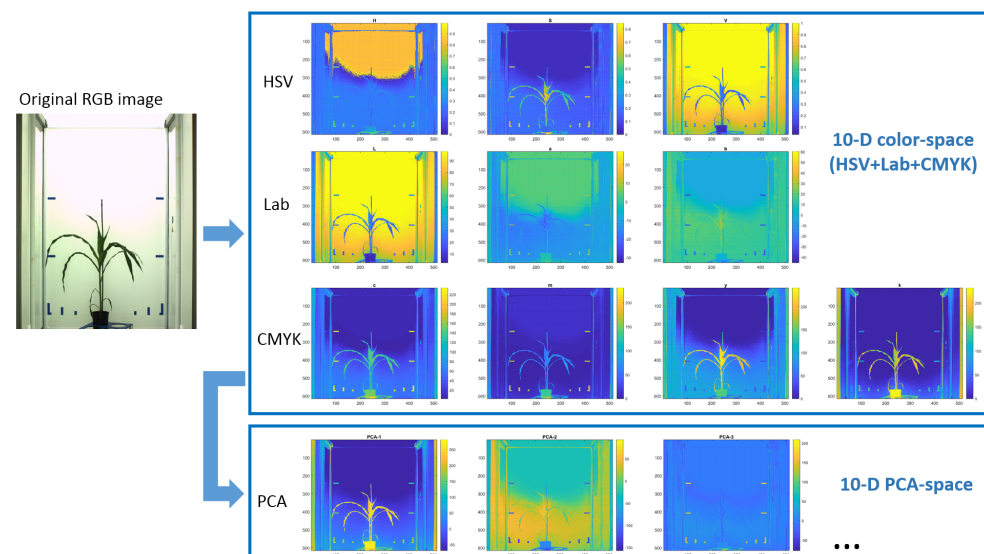


Figure 3. Color-space transformations of plant images in the kmSeg tool. First, original RGB plant images are transformed to a HSV+CIELAB+CMYK 10-dimensional color space. Then, principle components of 10-dimensional image representation are determined. Bottom row shows first three principle components ('Eigen-colors').

All color transformations were performed using MATLAB built-in functions:

- RGB to HSV: $hsv = rgb2hsv(rgb)$,
- RGB to CIELAB: $lab = rgb2lab(rgb)$,
- RGB to CMYK: $cf = makecform('srgb2cmyk');$; $cmyk = applycform(rgb, cf)$.

Thereby, it should be pointed out that MATLAB RGB to CMYK transformation is device-dependent and, therefore, uses a so-called ICC profile that characterizes the color output of a specific target device. By default, MATLAB uses the 'Specifications

for Web Offset Publications' (SWOP) standard ICC profile to transform from from sRGB IEC61966-2.1 to CMYK color space, what results in significantly different results compared with simple RGB to CMYK transformations often found online or in literature, see https://en.wikipedia.org/wiki/Specifications_for_Web_Offset_Publications (accessed on 11 February 2021).

2.2.3. Eigen-Color Transformation

To improve separability of plant and non-plant regions, principal (PCA) or, alternatively, independent components (ICA) of the 10-dimensional (HSV+CIELAB+CMYK) image representation (further termed as Eigen-colors) are determined. In the image pre-processing stage, users can principally select either PCA or ICA for calculation of image Eigen-colors. However, due to a significantly higher algorithmic complexity of ICA, PCA is suggested as a method of choice for rapid processing plant images of the typical size of several megapixels. Figure 3 (bottom) shows the first three principle components of the 10-dimensional image representation of a maize shoot image that correspond to three major color regions: (i) dark green/brown plant/carrier, (ii) white/light green background and (iii) blue marker pixels.

2.3. Unsupervised Image Pre-Segmentation Using *k*-Means Clustering

Unsupervised pre-segmentation of plant images into a user-defined number of sub-regions is performed using MATLAB *kmeans* clustering which was applied to 10-dimensional Eigen-color image representations.

In comparison to other conventional clustering algorithms, *k*-means clustering turned out to be more efficient and this more suitable for nearly real-time image segmentation. Comparison of clustering methods including *k*-means, spectral and hierarchical clustering is presented in the Experimental Results section below.

2.4. Supervised Labeling of Color Classes

Final image segmentation is performed by manual assignment of *k*-means color classes to either plant or non-plant categories. The assignment of plant and non-plant categories is done in a very intuitive and efficient manner, namely, by visual inspection and subsequent clicking on (i.e., selecting) appropriate plant color regions from pre-segmented *k*-means color classes in the GUI.

2.5. ROI Masking

In some plant images, the background regions exhibit so large spectrum of colors that accurate separation of fore- and background regions cannot be achieved just by selecting a relatively large number of *k*-means color classes. In such cases it is advisable to restrict the region of interest to the mask around the plant structures. For this purpose, optional manual masking of the region of interest (ROI) is introduced in the *kmSeg* tool.

3. Experimental Results

The basic idea of our approach to efficient ground truth labeling of plant images consists in automated clustering of image colors followed by selection of plant color classes using the GUI tools. A reliable clustering of images into plant and non-plant classes can, however, be hampered by statistical noise and/or topological vicinity of fore- and background colors in a color space.

To improve separability of plant and non-plant colors, a structure (edge) preserving Laplace smoothing can optionally be applied in the *kmSeg* tool. Figure 4 demonstrates the effect of Laplace smoothing on homogeneity of color distribution in a *Arabidopsis* side-view image. Especially by noisy and low-contrast images, structural enhancement is certainly of advantage for more accurate clustering of basic image color regions.

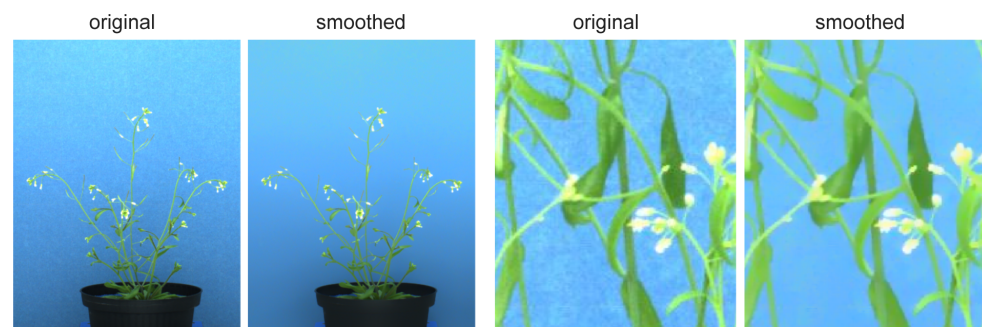


Figure 4. Examples of structure-/edge-preserving denoising of arabidopsis side-view images using Laplace smoothing.

Further important notion is that representation of images in different color spaces can be more or less optimal for separating fore- and background structures.

To quantitatively assess the degree of color decorrelation (D) in a particular color space, the following criterion was introduced:

$$D_j = \sqrt{\sum_{i=1.3} (e_{i,j} - 1/3)^2}, \quad (1)$$

where $e_{i,j} \in [0, 1]$ denotes the percentage of data explained by the i -th PCA component of the image representation in the j -th color space. Here, the percentage of data explained by PCA components was calculated using the MATLAB *pca* function. The criterion in Equation (1) was constructed in a way that $D \in [0, 0.816]$ takes the value $D = 0$ when image colors are distributed equally over all three PCA components, and the value $D = 0.816$ when only one single PCA component explains all data.

To systematically assess the degree of color decorrelation in RGB as well as alternative color spaces including HSV, CIELAB, CMYK, the D criterion in Equation (1) was calculated for a random selection of 100 greenhouse images. The summary of this performance test of k-means vs. spectral vs. hierarchical clustering algorithms is shown in Table 1. As you can see the degree of color decorrelation in alternative color spaces (HSV, CIELAB, CMYK) is higher (i.e., the D value is lower) than in RGB. Otherwise, the D values of alternative color spaces appears to lay in a relatively close range.

Table 1. Summary of color decorrelation (D) measurements (Equation (1)) in RGB, HSV, CIELAB and CMYK color spaces for a test sample of randomly selected 100 greenhouse images.

Color Space	D , Mean	D , Stdev
RGB	0.75	0.04
HSV	0.54	0.06
LAB	0.50	0.09
CMYK	0.45	0.11

A particularly strong decorrelation effect of the RGB to CMYK transformation can be traced back to the particular MATLAB implementation of the target-oriented transformation to a specific ICC color profile. Conventional RGB to CMYK transformations found in literature cannot have such effect, since they are linear transformations where the key value K is inverse to the V value of the HSV color space. Based on the results of this test, a 10-dimensional image representation in the combined HSV+CIELAB+CMYK color space was used here for subsequent clustering of greenhouse images into fore- and background structures.

For binary classification of image colors into fore- and background regions, different unsupervised methods of data clustering including k-means, spectral or hierarchical

clustering can be considered. However, in view of interactive nature of the manual image segmentation an efficient algorithmic performance is required.

To investigate the performance of the above three clustering methods, MATLAB built-in functions *kmeans*, *spectralcluster*, and *clusterdata* were used. In view of a large size of phenotypic images (i.e., $>1 \times 10^6$ pixels), initial performance tests of clustering methods were performed with synthetic data. For this purpose, a series of parametrically identical two-dimensional bi-Gaussian distributions of different size in the range between 400 and 40,000 points were generated. Figure 5a shows an example of such a bi-Gaussian distributed point cloud used in this test. The three above clustering algorithms were applied to a series of these synthetic point distributions with the goal to separate them into two clusters corresponding to the two original bi-Gaussian distributions, and to assess their performance in term of calculation time. The results of these performance tests shown in Figure 5b demonstrate that spectral and hierarchical clustering algorithms are computationally too expensive and, thus, cannot be applied for processing images of typical megapixel size within a reasonable period of time. In contrast, the k-means clustering algorithm has shown an acceptable performance. The MATLAB code of this performance test can be found in Algorithm S1.

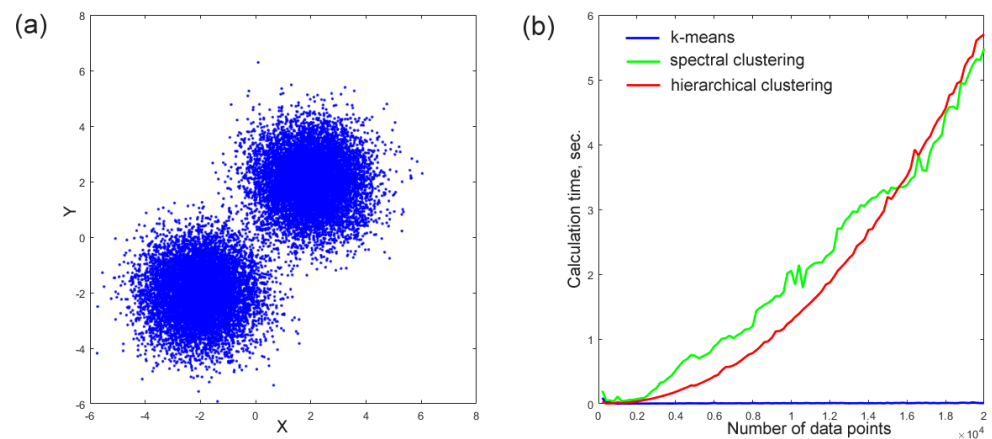


Figure 5. Evaluation of algorithmic performance of conventional clustering techniques including k-means, spectral and hierarchical clustering using synthetic data: (a) visualization of a 2D bi-Gaussian distributed point cloud, (b) plot of the calculation time of three clustering methods as a function of the data size in the range of $n \in [4e + 2, 4e + 4]$ data points. Significantly higher algorithmic complexity of spectral and hierarchical clustering algorithms makes their application to megapixel large images non-feasible.

Consequently, fast k-means clustering was adopted in this work for pre-segmentation of fore- and background image colors in the 10-dimensional Eigen-color space.

The basic approach to plant image segmentation in this work consists in unsupervised k-means clustering of image Eigen-colors followed by manual selection of color classes corresponding to the targeted plant structures, e.g., shoots, leaves, flowers, fruits. To optimize the result of unsupervised k-means clustering, a number of additional image pre-processing steps such as image filtering and/or ROI masking can be applied.

To enable users efficient processing and analysis of image data, the above described algorithmic framework was implemented as a GUI tool. Figure 6 shows a screenshot of the kmSeg tool including three major GUI elements: 'Control', 'k-means color classes' and 'Visualization' areas.

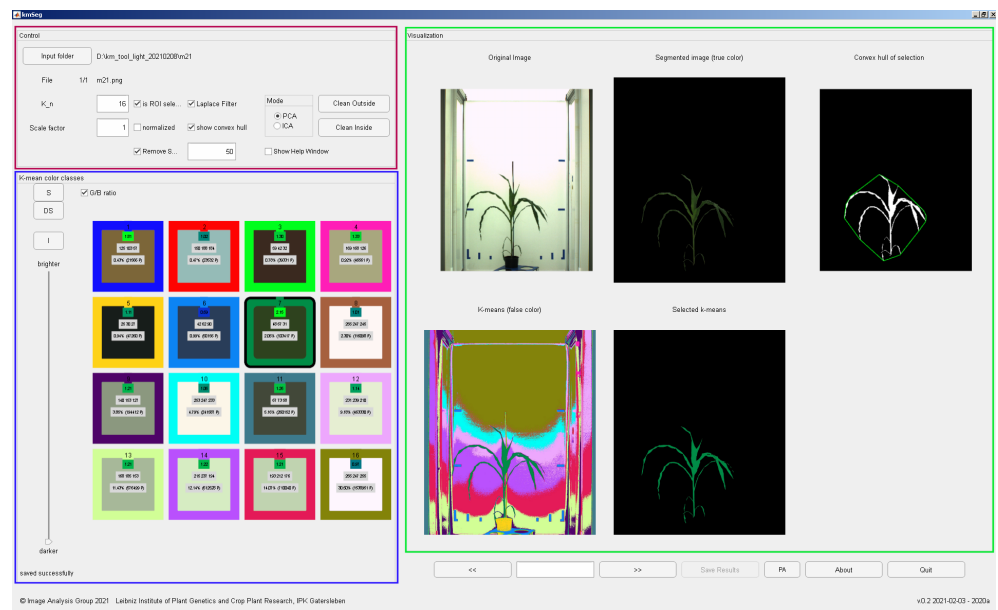


Figure 6. Graphical user interface of the kmSeg tool including three major GUI elements: the ‘Control’ (magenta frame), ‘k-means color classes’ (blue frame) and ‘Visualization’ (green frame) areas. The example shows segmentation of a side-view maize image by selecting only one single k-means color class (icon no. 7) out of totally 16 in the ‘k-means color classes’ area.

The work with the kmSeg GUI begins with selection of a target file directory in the ‘Control’ area containing one or more images stored in *.png or *.jpg formats. Once an image directory is selected and images are successfully imported into the program, the calculation immediately starts using the default set of algorithmic parameters including the number of k-means classes, the image scaling factor and filters that can afterwards be adjusted by the user. As rule of thumb, the higher the number of k-means classes is chosen the more fine and potentially more accurate separation of plant and non-plant structures can be achieved. Figure 7 shows an example of segmentation of a maize image using $k = 5$ and $k = 25$ k-means color classes, respectively.

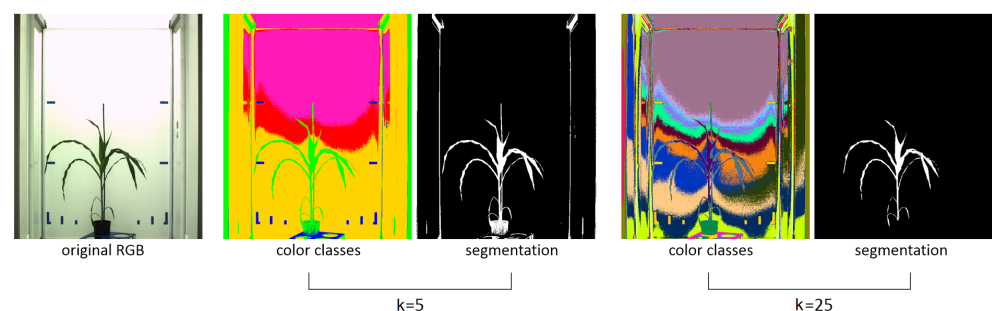


Figure 7. Automated segmentation of a maize shoot image into $k = 5$ and $k = 25$ k-means color classes. With the higher number of k-means classes a more accurate color separation of plant and non-plant structures was achieved. From left to right: original RGB image of a greenhouse-grown maize plant, quasi-color representation of k-means color classes as well the results of plant segmentation for $k = 5$ and $k = 25$, respectively.

A typical number of k-means classes for segmentation of greenhouse plant images ranges between 9 and 36 depending on complexity of color image composition. Fully automated determination of the number of k-means classes is not necessarily advantageous in this application, since users may want to adjust the algorithmic performance for optimal color separation and image segmentation based on their visual inspection. Depending on complexity of image colors, users can explore and select an optimal number of k-means

classes by trying and evaluating the results of image segmentation for a number of guesses, e.g., $k = 9, 16, 25, 36$. Furthermore, in the 'Control' area, user can define the type of color space transformation (PCA or ICA), optional downscaling ratio for faster processing of large images, image smoothing and filtering as well as visualization of the resulting convex hull of segmented ROI can be activated here. Changes in the 'Control' area automatically trigger re-calculation of image segmentation with the actualized set of parameters.

To restrict automatic pre-segmentation to a particular region of interest two functions 'Clean Inside' and 'Clean Outside' are provided in the 'Control' area. They allow the user to clean up the regions outside or inside of a freehand-drawn polygon around a particular ROI (e.g., plant). ROI masking allows the user to avoid artifacts due to faulty segmentation of shadows or reflection in the background region that are, for example, frequently observed on the boundaries of photo chambers. Furthermore, masking of the plant ROI reduces the complexity of color distributions which effectively shortens the calculation time and improves the accuracy of fore- and background color separation for the same number of k -means classes. Figure 8 shows an example of top-view arabidopsis image, where such masking was required to achieve a good segmentation result.

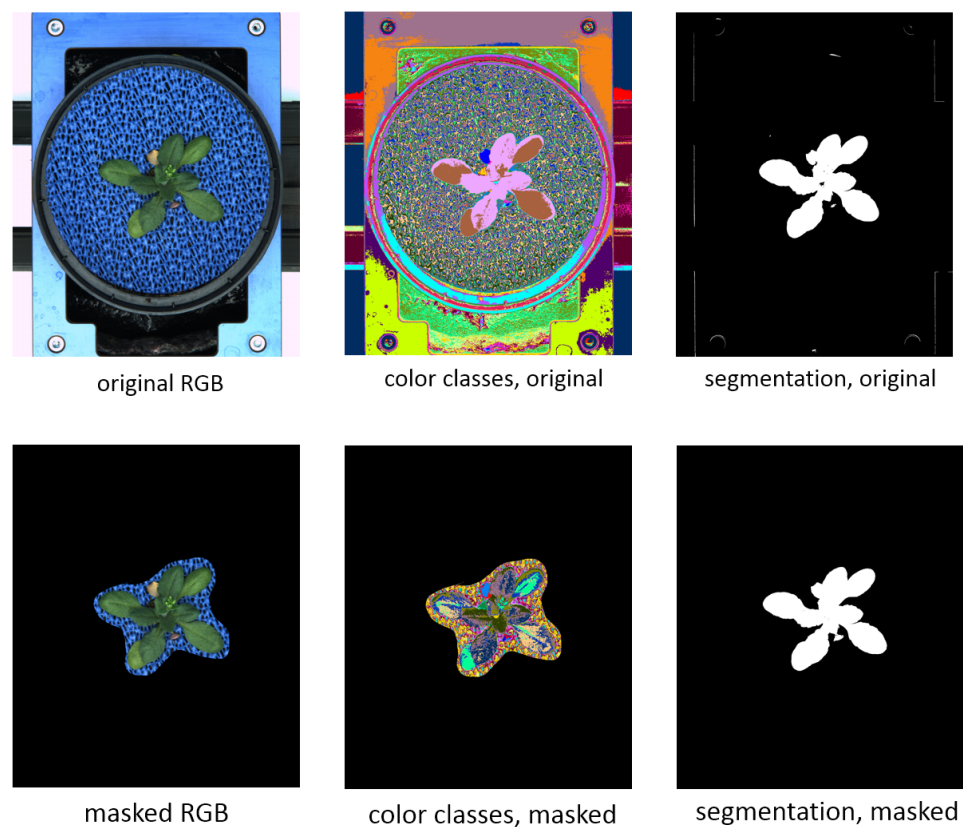


Figure 8. Example of k -means clustering of original full-size and masked images of greenhouse-grown arabidopsis plant. From left to right: original and masked RGB images, k -means color classes ($k = 25$), results of plant segmentation.

The movie https://ag-ba.ipk-gatersleben.de/kmseg_movie.html, (accessed on 11 February 2021) demonstrates all steps of the $kmSeg$ application to segmentation of the arabidopsis image from Figure 8 which took approximately 2 min on a Intel i7-6700HQ powered consumer-notebook with 16 GB RAM. Further examples of ROI masking for efficient segmentation of visible light and fluorescence arabidopsis, barley and maize shoot images in side- and top-views are shown in Supplementary Information (Figures S1–S12). Although, the $kmSeg$ tool was primarily developed for processing of images of greenhouse-grown plants, it can also be applied to other image data that can principally be segmented by means of color clustering. Further examples of the $kmSeg$ application including segmenta-

tion of fruits, flowers, leaf speckles, and multi-stain microscopic images can be found in Supplementary Information (Figures S13–S18).

The 'k-means color classes' area enables visual inspection and manual assignment of pre-calculated k-means color classes to either plant or non-plant categories. Here, the user is supported by a number of numerical indicators including

- running number of the k-means color class,
- the mean RGB values of the color class,
- the green-to-blue (G/B) ratio of the color class which is typically larger than one for plant structures,
- percentage of the total area of the color class,
- absolute number of pixels (area) of the color class.

Furthermore, spatial regions corresponding to all and selected color classes can be inspected in in sub-figures of the 'Visualization' area depicting pseudo-color, original RGB, binary segmentation with an optional convex hull visualization, see Figure 6. Assignment of pre-segmented color classes to plant or non-plant categories is performed by a single click on the icon of the k-means color class which corresponds to the targeted ROI. Renewed clicking on the selected color icon deselects the k-means class and assigns it to another category (e.g., from plant to non-plant category). After the first manual assignment of plant/background categories to colors of k-means regions, plant/background categorization of color regions in subsequent images is automatically extrapolated from the last manual segmentation. It can be, however, changed by the user anytime. By using the 'go back' or 'go forward' buttons, the previous or the next image can be selected. Instead of clicking several times, the user can directly jump to the sought image by entering its running number in the list of all images in the selected folder. By pressing the 'Save results' button the user saves all segmentation results in a subfolder of the source image directory.

In general, segmentation of images using the kmSeg tool depends on the size of images and/or selected ROIs. Figure 9 shows a summary of k-means clustering (i.e., the first automated step toward image segmentation) of up to 5 megapixel large images, which lays in the range between 5–60 s. In comparison, fully manual image segmentation using conventional tools (e.g., thresholding, manual drawing and cleaning in ImageJ) is expected to be several times more time consuming, depending on the user's skills, software choice and image complexity.

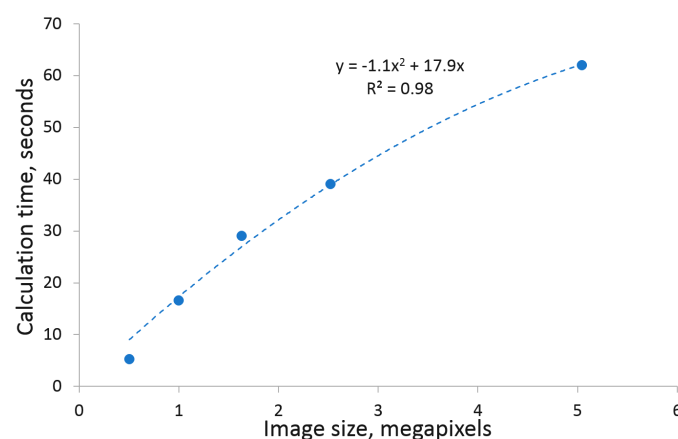


Figure 9. Computational performance of k-means clustering (without any manual editing) by segmentation of greenhouse plant images in dependence on the image size.

To quantitatively assess the accuracy and performance of the kmSeg tool by segmentation of different plant images, original and complementary ground truth images from A1, A2, A3 datasets published in [8] were used. All images were processed as described above using 36 k-means classes for clustering of PCA-transformed 10 dimensional

(HSV+CIELAB+CMYK) image representation, followed by optional ROI masking, selection of plant color classes and image cleaning.

Table 2 gives a summary of the kmSeg performance indicating that typical top-view plant images can be segmented and analyzed using the kmSeg tool within 2–6 min with an average accuracy (i.e., the Dice similarity coefficient) ranging between 0.96–0.99. Thereby, the most time consuming and less accurate segmentation results were observed for A1 images that exhibit a larger variation of colors and background vegetation with a similar color fingerprint as arabidopsis leaves. A2 and A3 images with higher plant-background contrast were segmented more efficiently and accurately.

Table 2. Summary of accuracy and performance of semi-automated kmSeg segmentation on A1, A2, A3 sets of top-view plant images from [8] in terms of the average Dice coefficient of similarity between kmSeg-segmented and ground truth images (\pm standard deviation). Columns ‘Clustering’ and ‘Cleaning’ indicate the maximum time span required for image segmentation using automated k-means clustering followed by optional manual cleaning.

Data Set	# Images	Dice Coeff.	Clustering	Cleaning
A1	128	0.959 \pm 0.011	<1 min	<5 min
A2	31	0.965 \pm 0.021	<1 min	<4 min
A3	27	0.986 \pm 0.010	<1 min	<3 min

As output of image segmentation, the kmSeg tool writes out following files

- segmented images including labeled color classes, RGB and binary images, see the ‘Visualization’ area in Figure 6,
- a *.csv file containing basic traits of segmented plant structures including descriptors of plant area, shape and color fingerprints in RGB, HSV, CIELAB color spaces, see the full list in Supplementary Information (Table S1),
- a plain ASCII file describing assignment of k-means classes to pseudo-colors of plant and non-plant regions, Figure S19a,
- a copy of the entire MATLAB workspace (*.mat file) of the kmSeg tool containing segmentation results and help-variables, Figure S19b.

*.mat files containing the entire internal kmSeg tool variables, that can be used by MATLAB users for a detailed analysis or serve for debugging purposes. Segmented images and complementary ASCII files allow users to retrieve all information necessary for quantitative description of segmented plant and non-plant image regions.

The precompiled executable of the kmSeg tool along with the user guide and examples of greenhouse plant images is provided for download from <https://ag-ba.ipk-gatersleben.de/kmseg.html>, accessed on 11 February 2021.

4. Conclusions

Accurate and efficient segmentation of optically heterogeneous and variable plant images represents a challenging, time-consuming task considerably limiting the throughput of phenotypic data analysis. For training of advanced machine and deep learning models, a large amount of reliable ground truth data is required. Here, we present a software solution for semi-automated binary segmentation of plant images which is based on combination of unsupervised clustering of image Eigen-colors and a straightforward categorization of fore- and background image regions using a intuitive GUI. Consequently, the kmSeg tool simplifies the task of manual segmentation of structurally complex plant images to just a few mouse clicks which can be performed even by users without advanced programming skills.

For the shoot images used as example in this work, the transformation from RGB to alternative color spaces, including HSV, CIELAB and CMYK, turned out to be advantageous for color decorrelation and clustering. Thereby, it should be emphasized that the MATLAB implementation of RGB to CMYK transformation, which is based on the specific SWOP

ICC profile, significantly differs from the conventional CMYK definition in the literature. In general, the choice of appropriate color spaces for image clustering and segmentation is essentially dependent on concrete image data, and can principally be different for other data and/or application. In our previous works on plant image registration and classification [2,27], the kmSeg tool was extensively used for generation of thousands of ground truth images of different plant types, modalities and camera views. Evaluation with ground truth images of different color variability and structural complexity has demonstrated that plant image segmentation and analysis using the kmSeg tool can be performed within a few minutes with an average accuracy of 96–99% in comparison to ground truth data. Despite the fact that this software framework was primarily developed for segmentation of plant shoots in visible light and fluorescence greenhouse images, it can be applied to any other images and image modalities that can principally be segmented using color or grayscale intensity information. The kmSeg tool was designed for binary image segmentation and plant shoot phenotyping. However, it can be also used for multi-class image segmentation when applied in an iterative manner by annotating only one target structure with a distinctive color fingerprint per iteration such as predominantly green-yellow leaves, red fruits, white background, brown speckles, or different color channels of multi-stain microscopic images. In addition to ground truth segmentation, kmSeg can be used as a handy tool for rapid calculation of basic phenotypic traits of segmented plant structures.

Further possible extensions of the present approach include generalization of binary to multi-class image annotation as well as introduction of additional filters and tools for efficient removal of remaining statistical and structural noise which could not be eliminated by rough ROI masking and color separation.

Supplementary Materials: The following are available online at www.mdpi.com/xxx/s1, Supplementary Information accompanies the manuscript.

Author Contributions: M.H., E.G. conceived, designed and performed the computational experiments, analyzed the data, wrote the paper, prepared figures and tables, reviewed drafts of the paper. K.N. executed the laboratory experiments, acquired image data, co-wrote the paper, reviewed drafts of the paper. T.A. co-conceptualized the project, reviewed drafts of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was performed within the German Plant-Phenotyping Network (DPPN) which is funded by the German Federal Ministry of Education and Research (BMBF) (project identification number: 031A053). M.H. was supported from European Regional Development Fund-Project “SINGING PLANT” (No. CZ.02.1.01/0.0/0.0/16_026/0008446). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This work presents examples of application of methods to images from open resources such as <https://search.creativecommons.org>, (accessed on 11 February 2021) as well as previously published works, e.g., [8]. Further examples of image data are available from the corresponding author on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Minervini, M.; Scharf, H.; Tsafaris, S.A. Image Analysis: The New Bottleneck in Plant Phenotyping. *IEEE Signal. Proc. Mag.* **2015**, *32*, 126–131 [[CrossRef](#)]
2. Henke, M.; Junker, A.; Neumann, K.; Altmann, T.; Gladilin, E. Comparison and extension of three methods for automated registration of multimodal plant images. *Plant Methods* **2019**, *15*, 44. [[CrossRef](#)] [[PubMed](#)]
3. Lee, U.; Chang, S.; Putra, G.A.; Kim, H.; Kim, D.H. An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis. *PLoS ONE* **2018**, *13*, e0196615. [[CrossRef](#)] [[PubMed](#)]
4. Zheng, Y.; Kong, J.; Jin, X.; Wang, X.; Zuo, M. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. *Sensors* **2019**, *19*, 1058. [[CrossRef](#)] [[PubMed](#)]

5. Barth, R.; Jsselmuiden, J.; Hemming, J.; Van Henten, E.J. Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Comput. Electron. Agric.* **2019**, *161*, 291–304. [[CrossRef](#)]
6. Zhou, C.; Hu, J.; Xu, Z.; Yue, J.; Ye, H.; Yang, G. A Novel Greenhouse-Based System for the Detection and Plumpness Assessment of Strawberry Using an Improved Deep Learning Technique. *Front. Plant Sci.* **2020**, *11*, 559. [[CrossRef](#)] [[PubMed](#)]
7. Adams, J.; Qiu, Y.; Xu, Y.; Schnable, J. Plant segmentation by supervised machine learning methods. *Plant Phenome J.* **2020**, *3*, e20001. [[CrossRef](#)]
8. Scharr, H.; Minervini, M.; Fischbach, A.; Tsaftaris, S. *Annotated Image Datasets of Rosette Plants*; Technical Report FZJ-2014-03837; Forschungszentrum Julich: Julich, Germany, 2014.
9. Minervini, M.; Fischbach, A.; Scharr, H.; Tsaftaris, S.A. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognit. Lett.* **2016**, *81*, 80–89. [[CrossRef](#)]
10. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173. [[CrossRef](#)]
11. Lingutla, N.T.; Preece, J.; Todorovic, S.; Cooper, L.; Moore, L.; Jaiswal, P. AISO: Annotation of Image Segments with Ontologies. *J. Biomed. Semant.* **2014**, *5*, 50. [[CrossRef](#)] [[PubMed](#)]
12. Iakovidis, D.K.; Goudas, T.; Smailis, C.; Maglogiannis, I. Ratsnake: A Versatile Image Annotation Tool with Application to Computer-Aided Diagnosis. *Sci. World J.* **2014**, *2014*, 286856. [[CrossRef](#)] [[PubMed](#)]
13. Tzutalin, D. *LabelImg*: Git Code. 2015. Available online: <https://github.com/tzutalin/labelImg> (accessed on 11 February 2021).
14. Fiedler, N.; Bestmann, M.; Hendrich, N. *ImageTagger: An Open Source Online Platform for Collaborative Image Labeling*; RoboCup 2018: Robot World Cup XXII; Springer: Berlin/Heidelberg, Germany, 2018.
15. Dutta, A.; Zisserman, A. The VIA Annotation Software for Images, Audio and Video. In Proceedings of the 27th ACM International Conference on Multimedia, New York, NY, USA, 21–25 October 2019.
16. Dias, P.; Shen, Z.; Tabb, A.; Medeiros, H. FreeLabel: A Publicly Available Annotation Tool Based on Freehand Traces. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 21–30.
17. De Vylder, J.; Vandebussche, F.; Hu, Y.; Philips, W.; Van Der Straeten, D. Rosette tracker: An open source image analysis tool for automatic quantification of genotype effects. *Plant Physiol.* **2012**, *160*, 1149–1159. [[CrossRef](#)] [[PubMed](#)]
18. Minervini, M.; Giuffrida, M.V.; Tsaftaris, S.A. An interactive tool for semi-automated leaf annotation. In *Proceedings of the Computer Vision Problems in Plant Phenotyping Workshop, CVPPP 2015*; BMVA Press: Swansea, UK, 2015.
19. Granier, C.; Aguirrezabal, L.; Chenu, K.; Cookson, S.; Dauzat, M.; Hamard, P.; Thioux, J.; Rolland, G.; Bouchier-Combaud, S.; Lebaudy, A.; et al. PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in *Arabidopsis thaliana* permitted the identification of an accession with low sensitivity to soil water deficit. *New Phytol.* **2006**, *169*, 623–635. [[CrossRef](#)] [[PubMed](#)]
20. TOLOKA. Take Control of your Data Labeling. Available online: <https://toloka.ai> (accessed on 11 February 2021).
21. DARWIN. A Scalable, AI Powered Image Annotation Platform. Available online: <https://www.v7labs.com/darwin> (accessed on 11 February 2021).
22. Philipp, I.; Rath, T. Improving plant discrimination in image processing by use of different colour space transformations. *Comput. Electron. Agric.* **2002**, *35*, 1–15. [[CrossRef](#)]
23. Yang, W.; Zhao, X.; Sile, W.; Zhang, J.; Feng, J. Greenness identification based on HSV decision tree. *Inf. Process. Agric.* **2015**, *2*, 149–160. [[CrossRef](#)]
24. Fahlgren, N.; Feldman, M.; Gehan, M.; Wilson, M.; Shyu, C.; Bryant, D.; Hill, S.; McEntee, C.; Warnasooriya, S.; Kumar, I.; et al. A versatile phenotyping system and analytics platform reveals diverse temporal responses to water availability in *Setaria*. *Mol. Plant* **2015**, *8*, 1520–1535. [[CrossRef](#)] [[PubMed](#)]
25. Kowsari, K. Comparison Three Methods of Clustering: k-Means, SPECTRAL clustering and Hierarchical Clustering. 2013. Available online: <http://xxx.lanl.gov/abs/https://arxiv.org/pdf/1312.6117v1.pdf> (accessed on 11 February 2021).
26. Jansen, M.; Gilmer, F.; Biskup, B.; Nagel, K.; Rascher, U.; Fischbach, A.; Briem, S.; Dreissen, G.; Tittmann, S.; Braun, S.; et al. Simultaneous phenotyping of leaf growth and chlorophyll fluorescence via GROWSCREEN FLUORO allows detection of stress tolerance in *Arabidopsis thaliana* and other rosette plants. *Funct. Plant Biol.* **2009**, *36*, 902–914. [[CrossRef](#)] [[PubMed](#)]
27. Henke, M.; Junker, A.; Neumann, K.; Altmann, T.; Gladilin, E. A two-step registration-classification approach to automated segmentation of multimodal images for high-throughput greenhouse plant phenotyping. *Plant Methods* **2020**, *16*, e20001. [[CrossRef](#)] [[PubMed](#)]