Plant Segmentation for Growth Analysis in Temporal Datasets

Author: Shisheng Wang

Supervisor: Dr. Hannah Dee
Prof. John Doonan

21st February, 2018
Mandatory Layout of Declaration/Statements

<table>
<thead>
<tr>
<th>Word Count of thesis: DECLARATION</th>
<th>27842</th>
</tr>
</thead>
</table>

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Candidate name: Shisheng Wang

Signature: [Signature]

Date: 01.03.2018

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. Where *correction services* have been used, the extent and nature of the correction is clearly marked in a footnote(s).

Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signature: [Signature]

Date: 01.03.2018

[*this refers to the extent to which the text has been corrected by others]*

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signature: [Signature]

Date: 01.03.2018

NB: Candidates on whose behalf a bar on access (hard copy) has been approved by the University should use the following version of Statement 2:

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loans after expiry of a bar on access approved by Aberystwyth University.

Signature: [Signature]

Date: 01.03.2018
Abstract

High-throughput phenotyping is an important means to meet the agricultural needs for future food and energy production. This entails an increasing amount of work in Image-based, non-destructive phenotyping systems. This thesis describes a low-cost phenotype collection system for growth chambers, and methods to segment plants from time series images using temporal information.

The system uses a webcam to record plant growth in a top-down view with a fixed time interval to create time-lapse images of multiple plants. It has successfully recorded the growth of Arabidopsis thaliana over three months from seedling to flower. The development of plant segmentation methods involves experiments to compare and select the optimal colour space for plant segmentation, and the development of an unsupervised plant segmentation method that is capable of segmenting multiple plant species (e.g. Arabidopsis thaliana, Oats, Oilseed Rape) without relying on knowledge of plant colour. The method is also modified to provide colour-based, superpixel-based and supervoxel-based approaches to the segmentation of plants from time series images.
Acknowledgements

I would like to express my deep gratitude to Dr Hannah Dee and Professor John Doonan, my supervisors, for their patient guidance throughout my degree.

I am grateful to everyone in the Computer Science Department and Institute of Biological, Environmental and Rural Sciences at Aberystwyth University, especially to Dr Bernie Tiddeman, for their help and support.

I also owe thanks to all my office mates and my friends, especially to Marek, Pete and Alex who have provided help and advice.

Last but not the least, I wish to thank my parents for their support and encouragement throughout writing this thesis and my life in general.
6 Superpixel and Supervoxel Segmentation

6.1 Introduction

6.2 Superpixel

6.2.1 Superpixel Method Comparison

6.2.2 Superpixel Algorithms Comparison Results

6.2.3 Choice of Superpixel Method

6.2.4 Combining SLIC with the Developed Method

6.2.5 Define the Number of Superpixels

6.2.6 SLIC Segmentation Performance on Plant Datasets

6.2.7 Superpixel Based Method Results

6.2.8 Superpixel with Plant Tracking

6.3 Supervoxel

6.3.1 Supervoxel-Based Method Implementation

6.3.2 Defining the Number of Supervoxels

6.3.3 Supervoxel Based Method Results

6.4 Conclusion

7 Conclusion and Further Work

7.1 Introduction

7.2 Discussion

7.2.1 The Experimental Setup
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.2</td>
<td>Exploring Colour Spaces</td>
<td>97</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Colour-Based Segmentation Method</td>
<td>97</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Superpixel- and Supervoxel-segmentation Methods</td>
<td>98</td>
</tr>
<tr>
<td>7.3</td>
<td>Conclusion</td>
<td>98</td>
</tr>
<tr>
<td>7.4</td>
<td>Key Contribution</td>
<td>99</td>
</tr>
<tr>
<td>7.5</td>
<td>Limitations</td>
<td>100</td>
</tr>
<tr>
<td>7.6</td>
<td>Future Work</td>
<td>100</td>
</tr>
</tbody>
</table>

Bibliography | 102
Chapter 1

Introduction

The increasing demand for food and energy due to global human population growth requires high production capabilities and robustness in agriculture in order to meet global needs. To solve this problem, governments and organizations have been working to develop better seeds, plants, and systems to improve resource production. This has entailed growth in the study of plant phenotypes and the development of high-throughput phenotyping platforms.

Plant phenotyping investigates the physical appearance and biochemical characteristics of a plant as a result of the interaction between its genotype and the environment. However, there is a gap between scientists understanding of the genetics, and their ability to link this with the way in which these genes are expressed in terms of an organisms physical characteristics. Understanding how the genotype results in the phenotype that scientists observe is one of the biggest challenges in biology, and is commonly termed the “genotype-phenotype gap” [1].

Traditional phenotyping approaches measure plant leaf area directly which often requires harvesting the plant. Leaves are cut from the stem and pressed against a monochrome background to take top-down photographs for measurements. Therefore, only one measurement per plant can be obtained, and differences in rates of growth are hard to measure. Moreover, the process is time-consuming, and can be subjective and error-prone.

With the help of computing, image-based, non-destructive approaches are becoming the mainstream for plant phenotyping. Methods have been developed to segment images taking plant measurements from images of living plants. Semi-
and fully-automatic systems are now able to take measurements of plants with
different types of cameras. This approach does not require harvesting the plants,
which means that fewer plants are required for high quantity data recording.

1.1 Hypothesis and Research Question

The chief hypothesis of this thesis is: “The use of temporal information can
improve plant image segmentation quality.”

The key, underlined terms of the hypothesis are elaborated upon here to clarify
four research questions:

- **Temporal information**: Time series images are taken by using a static
camera positioned over a set of static or moving plants. Is temporal infor-
mation derived from the time series datasets reliable for plant segmentation?
  Finding ways to deal with problems such as temporal variability, noise and
time-lapse frequency are the key to the question.

- **Plant**: Different plants have different characteristics. Sample plants used
  in this work include Oats, Oilseed rape and Arabidopsis. Can temporal
  information be applied to different types of plants?

- **Image**: Plant images are taken with different cameras and lighting settings,
  and other noise may also occur. Do these factors affect the segmentation
  performance and can these problems be solved?

- **Segmentation quality**: Colour-based and superpixel-based segmentation
  methods are used to segment plants. Can they be applied to temporal
datasets and improve segmentation quality?

Answering the questions would provide an alternative approach to image seg-
mentation with temporal information.

1.2 Objective

The aim of this work is to develop a segmentation method for a phenotype system
which makes use of temporal information to segment plant pixels from time series
images, with minimum interaction. Firstly, an image capture system is built to collect time-lapse plant images. Secondly, the acquired dataset is used to analyse plant colour to find the optimal colour-space for plant segmentation. Lastly, the method is developed using the colour-space, with minimal assumptions based on plant growth and colour information, in order to segment plants. Superpixel- and supervoxel-based methods are also investigated.

1.3 Contributions

A lot of research has provided methods for plant segmentation that are either supervised or require specific settings. We aim to develop an unsupervised plant segmentation method that can work on multiple plant species, e.g. Arabidopsis thaliana, Oats, Oilseed Rape. We have achieved this via a number of experiments. The key contributions developed over the course of the project are as follows:

- A supervised comparison between all popular colour-spaces to determine an optimal colour-space for plant image processing in visible light range; see Chapter 4.
- An unsupervised plant segmentation method based on plant growth to segment plants from time-lapse images without relying on knowing the plant colour; see Chapter 5.
- Investigation of a combination of superpixel methods with unsupervised segmentation to further improve the segmentation results and track plants over a set of time series images; see Chapter 6.
- A new set of Arabidopsis time-lapse image data from seed to flower; see Chapter 3.
- A number of manually-segmented ground truth images derived from various plant datasets used in this project. These can be used for methods comparison and evaluation in other plant research; see Chapter 3.
1.4 Thesis Structure

The remainder of this thesis is structured as follows:

- Chapter 2 reviews existing work in plant phenotyping, image segmentation and other related computer vision techniques used in these areas.

- Chapter 3 describes the plant image capture system setup, looking into plant image datasets and possible ways to evaluate the results.

- Chapter 4 provides an analysis of different colour-spaces, and tests their performance on plant images.

- Chapter 5 presents the development of unsupervised segmentation methods using temporal information.

- Chapter 6 tests and improves the segmentation results by combining the method with superpixel and supervoxel algorithms.

- Chapter 7 provides a discussion of the developed system described in chapter 5, and moves on to critically review methods, offer concluding remarks, and point to future work that might be undertaken on the basis of this research.
Chapter 2

Background

2.1 Introduction

This chapter provides an overview of plant phenotyping and computer vision techniques that are related to this work. First, we describe the current status and importance of plant phenotyping. Then we look into the hardware and software that are used and developed in this area. Finally, we describe a list of existing methods which are used or compared in later chapters.

2.2 Arabidopsis Thaliana

The first literature record of Arabidopsis dates from 1873 when German botanist Alexander Braun described a wild-type plant found in a field near Berlin [2]. As a model plant (or model organism which is defined as “a non-human species that is extensively studied to understand particular biological phenomena, with the expectation that discoveries made in the organism model will provide insight into the workings of other organisms [3]”), Arabidopsis has been widely adopted in the area of plant genetics, physiology, and molecular genetics since the 1980s. In the year 2000, the DNA sequencing of the Arabidopsis genome was completed [4], a breakthrough that would help the further analysis of plant life and evolutionary history.

Arabidopsis can be found all over the world, mainly growing in Europe, North America and Asia. Nine species and eight subspecies are currently recognised,
most of which are indigenous to Europe. Arabidopsis has a short life cycle, from sowing to seed harvest only requires approximately 6 to 8 weeks. This fast growth allows for rapid experimentation, potentially speeding up the progress of scientific research.

Early analysis of Arabidopsis using computer vision can be found in [5] where experiments were setup to measure leaf movement and size in pixels every 20 minutes with a video camera from vertical and lateral positions. Since this early vision work, Arabidopsis thaliana has been widely studied using image processing and analysis techniques.

2.3 Phenotyping

Phenomics is “a study concerned with the measurement of phenomes, the biochemical and physical traits of organisms with respect to their change in response to environmental influences and genetic mutation [6]”. The phenotype is the result of the interaction between the genotype and environment. However, the rate of progress in phenotyping lags behind that in genotyping, and this creates a bottleneck [7].

Traditional approaches destroy the plant when measurements are taken; this requires a large quantity of the plants if the measurements are taken frequently. However, destructive methods take measurements at the end of the experiment and incur the loss of data concerning spatial and temporal features during growth [8]. Therefore, biologists are interested in non-destructive methods to acquire plant data [9, 10]. High capacity data recording and automated environmental data collection have become key for high-throughput phenotyping [11].

Taking measurements from images allows us to collect the phenotypic characteristics of plants through the application of computer vision. Therefore, a large quantity of data can be obtained without destroying the plants [12]. The type of measurements that can be taken from images includes colour, shape, size, growth rate, etc. The manual collection of these plant data from a large number of samples is time-consuming, subjective and prone to errors. For these reasons, biologists have sought to automatically measure all aspects of plants via computer vision techniques, and this in turn has led to a growth in the development of non-invasive, imaging-based methods for plant phenotyping [13, 14].
Amongst plant characteristics, colour is one of the most important observable elements. Plant colour and texture can provide useful information such as whether or not the plant is under stress, and whether the plant is thirsty or overwatered, and these characteristics can be perceived by humans [15]. We are interested in the colour difference between wild-type and mutant plants, and colour variation over the course of a plant's growth process. Since this early vision work does not destroy plants, the process has been adapted to study the growth of many different kinds of plants.

2.3.1 Plant Phenotyping Systems

Plant phenotyping evaluates the viability of a given genotype through its organisms observable characteristics or traits. This work can be broken down into two steps: data acquisition and data analysis.

Plant data acquisition can be performed for multiple plants either indoors or in the field with the data captured in image or video formats. There are also automated phenotyping systems built to collect plant image and weight data periodically [16, 17]. The cost of a setup could range from several hundred pounds to hundreds of thousands of pounds. Plant phenotyping can be done with a single low cost digital camera [18], multiple cameras [19], or expensive devices sourced from commercial entities (LemnaTec, CropDesign).

Different types of recording device can be used in many approaches, for example popular devices for plant data recording include visible light cameras (VIS), near infrared cameras (NIR), infrared cameras (IR) and fluorescence cameras. LemnaTec uses multiple types of camera for data collection. IR and NIR cameras are commonly used to photograph plants both during daylight hours and at night [8]. Good quality data can also be collected using low-cost hardware; for example, a Microsoft Kinect Device can be used to build 3D model of the plant [20].

Multispectral cameras allow different wavelengths to be captured separately; this also includes light from frequencies beyond the visible light range. Many applications have been developed with multispectral fluorescence and reflectance imaging for characterizing plants and their health status [21, 22]. Biologists frequently use integrating sphere spectrometers to record spectrum measurements for specular/diffuse components in order to study leaf optical properties [23, 24].
A laser scanner is used to record the diurnal pattern of leaf growth [25]. Laser rangefinder and depth cameras have been used to acquire 3D depth information about plants [26, 27]. Chlorophyll fluorescence imaging is used to observe photosynthetic activity [28, 29, 30].

For regular RGB cameras (cameras that acquire and deliver red, green and blue colour signals), different camera configurations provide a variety of ways to obtain plant data. Top-down views can record leaf surface growth effectively, and side views are better for recording the plant height growth. Multiple images with different viewpoints of a single plant could provide better image-based plant modelling [31, 32, 33]. Multiple images can also be used to calculate 3D plant models with space carving [34].

2.3.2 Data analysis for Plant Phenotyping

Some systems and algorithms are designed for leaf recognition with a view to the classification and management of plants. One solution is to build a leaf database from different plants and classify leaves based on digital morphological features [35]. The morphological features are more robust than contour-based features since it is difficult to find significant curvature points for leaf.

For leaf area measurement, a simple solution is to count the total number of pixels in a leaf area region with a reference object (e.g. a coin), then calculate the area based on the reference object [36]. This method can work with low cost digital camera; however, it requires that the leaf be separated from its stalk and placed on a clean background in order to provide accurate results.

There are also hardware systems designed to utilise temporal information from time series images or videos. Images are taken at a fixed time interval, which can vary from a few minutes [37] to once a day [38] depending on the plant growth rate, and the amount of data obtained may also vary. The temporal information can be used to monitor the plant growth under different light conditions for different plant species [39] or for the purposes of leaf tracking via fluorescence video [40]. All these systems acquire plant images or videos under a controlled stable environment of a known species.

For software systems, a number of ImageJ plugins have been developed for plant phenotyping using a Gaussian mixture model for background extraction,
and plant segmentation [19, 41]. These systems require either a controlled setup or some reference data as input parameters which are not efficient when processing a large quantity data with a large variety of plant species.

Application such as Rosette Tracker [41] use a mixture of Gaussians with Expectation Maximization (EM) on the hue channel (from the HSV colour space). The method segments plants by selecting the Gaussian with a mean closest to a manually selected hue value (if the hue value is not manually selected, a hue value that is close to green is used as the default value). For model order, the method starts with 2 clusters and increases the number of clusters until any cluster reaches a mean that is close to the specified hue value within a defined range. There are two main stages in Rosette Tracker: Segmentation and Rosette Detection. The segmentation method is described above. Detection is performed by applying a connected-components algorithm to the segmented results, then using a nearest-neighbour approach with a manually defined number of plants to classify which rosette a pixel belongs to.

2.4 Colour

The results of colour-based algorithms heavily depend on the choice of colour space [42, 43]. Especially, different colour spaces work differently to segmentation results. However, there is no optimal colour space to work on every single image, so the choice of a good colour space increases the precision of the segmentation.

Plants of the same species under the same environment often share similar colours. However, under different light sources, leaf colour varies due to the light reflection. From a physical perspective, light reflection occurs whenever light strikes the surface of an object. The interaction reflects two types of light: specular (or interface) reflection and diffuse (or body) reflection [44]. Specular reflection consists of light reflected from the surface of the object, and diffuse reflection returns the colour of the object. Therefore, when a coloured object has both specular and diffuse reflection, usually only the diffuse component is coloured.

The light reflection may result in small changes in plant colour and makes it difficult to compare plant colour differences. By leveraging plant reflectance and mechanisms to generate diffuse images, it could be easier to detect and observe plants.
2.4.1 Colour Calibration

Colour calibrations are generally used to modify the colour behaviour of a device. The process is mainly achieved by changing the device settings and controls, or by applying curves to the colour channels. One of the simplest ways to perform colour calibration for digital cameras is by using a colour calibration target, e.g. ColorChecker (X-Rite ColorChecker Targets) to generate camera calibration profiles which can be applied to all photos taken in the same environment.

The aim of colour calibration is to adjust the camera device and output image to a known state. This can be done by calibrating the device with a calibration target in order to produce consistent images. A ColorChecker target is used as a calibration target, in this way we can produce colour-stable images over time. There are many existing colour calibration methods developed for cameras or images which could be viable to solve our problems. We have experimented with some algorithms below, which aim to retrieve a stable plant colour via different cameras.

2.4.2 The ColourChecker

The existing ColorChecker camera calibration software developed by X-Rite can produce a Digital Negative (DNG) profile for a given image which can be applied to other images taken from the same environment.

The standard raw image format is Tag Image File Format/Electronic Photography (TIFF/EP), which was published in 2001 [45]. It provides a basis for the raw image formats of a number of cameras with added proprietary data. As an extension of TIFF/EP, Adobe launched the DNG format as another lossless raw image format for digital photography. Most raw image data are capable of conversion into DNG format which is supported by most Adobe products, including the ColorChecker applications. The DNG profile contains colour matrices (which each handle a linear component of colour correction) and a set of lookup tables; users are free to modify DNG settings with Adobes DNG Profile Editor.

The ColorChecker is included in all plant images in order to inspect the correctness of image colours. The software cannot process image data from our initial experiment since we only captured JPEG rather than RAW images images. However, we can still estimate a transformation based on colour balance.
2.4.3 Colour Balance

Colour balance is used to adjust the global colour intensities. This allows us to correct images which are either taken in artificial light or underexposed. Most digital cameras nowadays come with automatic colour balancing (also referred to as AWB) function [46].

White balance is one of the basic methods of this type. The method generally assumes that there is a white pixel in the image and attempts to find it or the closest one. Then it applies the difference between the white colour and the found pixel to the image in order to correct colours. One simple white balance algorithm involves the calculation of the difference between a white pixel in the image with an expected value and its application to all pixels [47]. In Figure 2.1, the image on the left side is the original image taken by a Canon camera. Such images are characterized by yellowing due to fluorescent light. To correct this, white balance has been applied, and the mean of the white value from the ColorChecker has been used. The results are shown on the right side: every pixel has been multiplied by the ratio of the white value in the original image to the correct white value in RGB (245, 245, 243) [48] which is the value of the ColorChecker while in the image.

There are many other colour constancy algorithms, and we can apply colour balance techniques by stretching values in each channel. This is done by: saturating a percentage of the minimum and maximum RGB (or other colour space) value; setting them to 0 and 255; and dividing each pixel across this range [49].

We can also perform colour correction globally or locally. To correct colour
globally, a “gray world” algorithm can be used which takes the average RGB value from the pixel and then tries to correct white colour by fitting R-G and -B-G into the same value. More colour constancy algorithms are discussed in [50, 51].

2.4.4 Reflective Model

If we could separate the specular reflection from images, we would be able to obtain images that contain only diffuse reflection which will provide correct colour information for the purposes of our test objects.

The dichromatic reflection model is one of the most used reflection models based on the physical properties of reflection [52]. The model represents light reflection with a mathematical formula (2.1) which states that the total radiance ($L$) is the sum of the radiance reflected at the interface ($L_i$) and the surface body ($L_b$). These light components can also be decomposed into two parts: composition ($c_i$ or $c_b$) is the relative spectral power distribution depending on wavelength and magnitude ($m_i$ or $m_b$) is the geometric scale factor depending on geometry (illumination, view directions and surface orientation).

$$L(\lambda) = L_i(\lambda) + L_b(\lambda)$$
$$= m_i c_i(\lambda) + m_b c_b(\lambda) \quad (2.1)$$

Based on the reflection model and difference above, many methods have been developed for separating reflection components. Among them, many methods require the use of a polarizing filter [53, 54, 55], since for most incident angles, diffuse reflection is less polarized than specular reflection. However, the filter is not utilized in our experiment. Tan [56] produced a method using a single image to separate reflection components.

There are three main optical differences between specular and diffuse reflections.

- The reflection has different degrees of polarization (specular reflection is generally more polarized than diffuse reflection) [57].
- The intensity distribution of specular reflection follows the Torrance-Sparrow reflection model (from the basis of geometrical optics) [58] and the Beckmann
Spizzichino reflection model (from the basis of physical optics) \cite{59}. The intensity distribution of diffuse reflection follows Lambert’s Law.

- For most inhomogeneous objects, the spectral power distributions of specular and diffuse reflections are different (Dichromatic reflection model) \cite{52}.

### 2.4.5 Polarizing Filter

A camera with a polarizing filter prevents highlights on images caused by direct reflection from a light source. It is often used to see through or detect windows and other transparent objects \cite{60}. Plants often reflect a large amount of light to prevent dehydration, and for this reason using a polarizing filter can improve the appearance of vegetation by causing the plants to appear greener (Figure 2.2). Many algorithms have been developed based on use of a polarizing filter to separate reflection components (as described in the Reflective Model section). Polarizing filters do cut out some of the light, making images as a whole darker. We have taken our plant image set without a polarizing filter due to the system setup. However, all images are taken indoors under a controlled environment in order to prevent some highlights on plants.
2.5 Segmentation

Image segmentation is the process of partitioning an image into multiple meaningful segments in order to reduce complexity. Segmentation processes are commonly used to separate objects from the background. Methods have been developed to divide pixels based on colour, texture, intensity or by detecting contours such as lines and edges.

In the field of plant phenotyping, segmentation is required to provide better plant measurement and analysis. Many studies avail themselves of a threshold over the green channel or multiple colour channels to segment leaves or plants [61, 62]. The aim of these methods is mainly to segment the green parts of a specific plant. Another popular method for segmentation is the use of a K-means clustering algorithm [63, 64], which can use texture and colour features to form the feature vectors in order to classify objects into K classes of pixels.

Clustering is a popular approach to segmenting objects. Clustering algorithms divide data into groups that share similar attributes or structures. Image segmentation using clustering often involves splitting an image into segments based on colour or intensity.

There are different types of clustering approaches that have been developed for different objectives. Most clustering methods can be categorised into connectivity-based, centroid-based or distribution-based types. Which one to use is determined by the data type, data size and research goal (or usefulness).

Connectivity-based clustering methods build modes based on the distance between data points. The hierarchical clustering algorithm [65] is one of the connectivity-based methods which iterates and merges the closest pair of clusters into a single cluster until all data are clustered into a single cluster. These types of clustering methods do not require knowledge of the number of clusters to start and can be easy to implement. But they do have high time complexity and are sensitive to noise.

Centroid-based clustering methods create mean vectors to represent each cluster. Both K-means [66] and Fuzzy C-means (FCM) [67, 68] belong to this category. K-means defines the number of clusters (k) based on user input and are represented by their centre. The centre with the smallest distance to a data point is considered to be the data's cluster. Studies have been devoted to finding the optimal number
of $k$ \cite{69}. Additionally, a compactness value can be used to discover how well the data has been modelled by calculating the total sum of the distance between each data point and their closest cluster centre. K-means is very fast and robust, but it requires the number of clusters as input and is sensitive to noise. Fuzzy C-means works similarly to K-means by assigning each data point to corresponding clusters based on the distance between the cluster centre and the data point, but, unlike K-means, it allows the data on the border of a cluster to be assigned to multiple clusters. A membership value is used to represent the likelihood of a data point belonging to each cluster. The total sum of the membership for a single data is 1. FCM provides better results for overlapped data, but it requires the number of clusters as input.

Distribution-based clustering methods build models using statistical distributions. Gaussian Mixture Models (GMM) \cite{70} is a distribution-based clustering method which represents a fixed, user-defined number of clusters ($k$) by using multiple Gaussian distributions (which can be represented using mean and co-variance). The algorithm fits data into $k$ number of Gaussians by maximizing the maximum likelihood of Gaussian centres. Although the algorithm is more complex than other clustering methods, it produces useful results for real world datasets.

### 2.6 Classification and Model Order

It is popular to segment objects using clustering; however, predefining the number of clusters is difficult since a fixed value would not satisfy all kinds of images/data unless the setups are limited or assumptions are made to prevent changes in data. With too few clusters, each object can barely be represented, and with too many clusters, it may over-fit the data. To determine the optimal number of clusters for our segmentation method, we have investigated a number of model order algorithms shown below.

We decided to use a classification algorithm to separate different components from our plant image data in order to segment plants. The classifier we use in this work is a Bayesian classifier using a Gaussian Mixture Model (GMM) trained with the Expectation Maximization (EM) algorithm \cite{70}. The advantage of this is that the trained model is capable of labelling test data with a set of likelihood values from each mixture component instead of trying to identify the exact component the
data belongs to. Although in this case the component with the highest likelihood of belonging to a given data is selected for the label. The GMM is a collection (K) of n-dimensional Gaussians each defined by three parameters: weight $\pi_k$, mean $\mu_k$ and covariance $\Sigma_k$. Its Probability Density Function (PDF) with sample data $x$ can be defined as ($T =$ Matrix Transposition)

$$p(x) = \sum_{k=1}^{K} \frac{\pi_k}{(2\pi)^{\frac{n}{2}}|\Sigma_k|^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right)$$  \hspace{1cm} (2.2)

Then the EM algorithm is used to define the parameters for each Gaussian model. This is an iterative process which contains an E step to find the probability $p_k$ of each Gaussian given each data $x$ defined as

$$p_k = \frac{\pi_k p(x | \mu_k, \Sigma_k)}{\sum_{i=1}^{K} \pi_i p(x | \mu_i, \Sigma_i)}$$  \hspace{1cm} (2.3)

Once the new $p_k$ is obtained, the M step then updates each parameter using the new value ($N =$ size of the data)

Weights:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} p_{ik}$$  \hspace{1cm} (2.4)

Means:

$$\mu_k = \frac{\sum_{i=1}^{N} p_{ik} x_i}{\sum_{i=1}^{N} p_{ik}}$$  \hspace{1cm} (2.5)

Covariances:

$$\Sigma_k = \frac{\sum_{i=1}^{N} p_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^{N} p_{ik}}$$  \hspace{1cm} (2.6)

For initialization, the EM algorithm requires some parameters to start with. These are supplied by:

- random data points from the dataset;
- data points from the boundary of the dataset;
- estimation using a K-means algorithm.

The random points and boundary points may result in a bad distribution and cause results to vary each time. A simple cluster algorithm such as K-means can
provide good initial parameters and is often close to the final model compared to other approaches.

When using a clustering method, one of the key things which needs to be decided is the number of clusters, or in other words, the model order. However, predefining the number of clusters is difficult since a fixed value would not satisfy all types of images/data unless the setups are limited (using fixed equipment, environment and layout to produce data so that the number of clusters is fixed).

Selecting the number of clusters becomes less important when the number of components in a scene are known or the method is supervised \[71\]. For unsupervised models, it is crucial to estimate the optimal number of clusters. \[72\] classifies model order methods into two categories: deterministic and stochastic.

Deterministic methods apply different model selection criteria (e.g. Bayesian criteria, Likelihood) based on the output of a range of \( k \) (number of clusters). This can either start by increasing from a small value or reducing from a large value, in any case, the optimal \( k \) is assumed to be contained. Most model order methods are deterministic, such as Akaike’s Information Criterion (AIC), Bayesian Information Criterion (BIC) and Minimum Description Length (MDL). Laplace-empirical criterion (LEC) \[73\] and integrated classification likelihood criterion (ICL) \[74\] are found to have the best performance among all methods.

Stochastic methods use random variables to find the optimal \( k \). Compared to deterministic methods, stochastic methods are rarely used due to their high computational cost. The Markov chain Monte Carlo (MCMC) \[75\] method is the most widely used in this case.

Many methods and applications have been developed for selecting the number of clusters:

**Rule of Thumb** One of the simplest methods based on the number of data points, calculated using the formula:

\[
k = \sqrt{\frac{n}{2}}
\]

This method is used to calculate an approximate value and often produces too many clusters for large datasets with minor differences \[76\], especially for an overlapped dataset.
Elbow Method  Uses a cost function with increased $k$ and finds a bend (based on the gradient) to be optimal $k$. The sum of square error (SSE) between each data point and their cluster centre are used in the cost function. The gap statistic \cite{77} is similar to the elbow method. It finds the largest gap (difference) between two $\log$ functions calculated based on the sum of the squared Euclidean distance between a range of $k$.

Either method encounters a problem when the gap/gradient between $k$ is equal. Figure 2.3 shows plots of $k$ and SSE from two datasets. In dataset A, at $k = 4$ the "elbow" is clearly visible, and selected by the elbow method as the optimal number of clusters since values after elbow usually start to have diminishing returns by increasing $k$. However, in dataset B, the gradient between $k$ is equal, hence the optimal number of clusters cannot be estimated by the elbow method.

Bayesian Approaches  For all approaches, data can be obtained using an EM algorithm between a specified range of $k$. However, EM is very sensitive to initialization.

The Bayesian Information Criterion (BIC) \cite{78} is defined as

\[
BIC = k \ln(N) - 2 \ln(L)
\] (2.8)

Where $k$ is the number of free parameters, $L$ is the likelihood (and $\ln(L)$ is often written as LL). BIC takes the number of sample data into consideration. A model
with a minimum BIC value is preferred. The BIC tends to select a simple model when a small sample size is applied.

However, as stated in [74], the BIC does not have the ability to provide evidence for a $k$ value in a mixture content. This is because the estimation will hit the parameter space boundary when a large number of components are used.

**Information-Theoretic Approaches**  
Akaike’s Information Criterion (AIC) [79] is defined as

$$AIC = 2k - 2 \ln(L)$$  \hspace{1cm} (2.9)

The criterion aims to find a balance between the complexity of a model (number of parameters) and the goodness of fit of the model (maximum log-likelihood). A model with a minimum AIC value is preferred. The AIC tends to select complex models when a large sample size is applied.

Minimum Description Length (MDL) [80] is defined as

$$MDL = \frac{1}{2}k \ln(N) - \ln(L)$$  \hspace{1cm} (2.10)

Figueiredo and Jain [72] purposed an algorithm using a Minimum Message Length (MML) criterion to find the optimal model directly during the training process. The algorithm uses a component-wise EM algorithm (CEM$^2$) [81] instead of a regular EM with a modified weight function allowing a kill component by setting its weight to zero.

**Likelihood Approaches**  
Integrated Classification Likelihood (ICL) [74] is defined as

$$ICL = \log(L) - k \log(N) + \log(\Gamma(\frac{K}{2})) + \sum_{k=1}^{K} \log(\Gamma(n_k + \frac{1}{2})) - K \log(\Gamma(\frac{1}{2})) - \log(\Gamma(n + \frac{K}{2}))$$  \hspace{1cm} (2.11)

The Gamma function ($\Gamma$) can be calculated using a Stirling formula to approximate the factorials especially when $n$ is a large number.

All the above criteria provide a statistic value for each $k$ to show how well the $k$ fits the given data. The optimal $k$ can be found by comparing the lowest/highest value and the degree of stability of the values between each cluster.
2.7 Superpixels

An object from an image is usually formed by more than one pixel, and it would be easier to process (segment) pixel groups with similar attributes than a large quantity of pixels. Doing so could improve segmentation results and reduce computational complexity.

A superpixel is an area of pixels that share a similar colour or texture. They are used to group pixels and features together in a graph in order to segment and analyse the region as a whole. It can reduce the complexity of an image and hence entails less training time for any subsequent processing. It can also provide oversegmentation results. The use of superpixels with colour segmentations can also improve results for the object boundary. Most superpixel methods are based on pixel colour, and therefore they are not efficient when applied to images containing textures with high differences in colour intensity.

Although there has been a lot of research on superpixels since the term was established [82], the benchmark has been proposed only recently [83] by the introduction of two error metrics: Boundary Recall (TPR rate for a given boundary against the ground truth edges that fall within a certain distance) and Undersegmentation Error (correctly segmented pixels under total segmented superpixels).

Superpixel algorithms can be categorized as either graph-based or gradient-ascent-based methods. A number of state-of-the-art algorithms are reviewed below with tests using their open source implementations (from OpenCV and scikit-image).

2.7.1 Graph-based

These methods treat each pixel as a node, and weight between two nodes as an edge. Superpixels are generated by minimizing a cost function defined over the graph.

Normalized cuts [84] Most superpixel algorithms are graph-based. Normalized cuts recursively cycle through all pixels from the image, calculating the optimal partition using generalized eigenvalue systems (to find the splitting point). The method is controlled by the number of defined superpixels. The algorithm is relatively slow (the base method is considered the slowest method) compared to other superpixel methods, and it has poor boundary adherence.
Efficient Graph-Based Image Segmentation (FH) [85] This algorithm uses pairwise region comparisons to generate superpixels. Resulting superpixels have irregular sizes and shape, and the user does not have control over the number of superpixels. While testing FH (using a scikit-image) with an Arabidopsis web-cam image (1600*1200pixels), the size and shape of superpixels are very unstable, and the number of segments varies from one hundred to a few thousand. The results sometimes provide poor segmentation with one large superpixel covering over half of the image, and this could be caused by the input parameter (scale) that influences the segment size.

Superpixel Lattices [86] This algorithm generates a boundary map using a non-returned vertical or horizontal path (finding the path seam with the lowest cost in one direction only) to form grid shaped superpixels. The method adds the path with minimal cost and uses graph cuts to divide the image. The method allows control of the size, number and compactness of the superpixels.

Superpixels and Supervoxels in an Energy Optimization Framework [87] Work of this kind involves stitching overlapping image patches, categorizing each pixel to one of the overlap regions with an energy function, and dividing the image with graph cuts. The method is also applicable to 3D supervoxel segmentation.

2.7.2 Gradient-ascent-based

These methods initialise by roughly defining some clusters, then iterate through pixels and reassign them to clusters to generate superpixels.

Quick Shift [88] The method is based on a (mode seeking) non-parametric clustering algorithm: Mean Shift [89]. It first calculates the Kernel Density Estimation (KDE) and converges each pixel towards the same mode from a cluster. Quick shift speeds up the approach by grouping each pixel to the nearest neighbour (or which there is an increment of the cluster density KDE) instead of using the gradient. The algorithm has good segmentation accuracy with a slow speed and does not require the user to control the size or number of superpixels.

While testing quick shift (using a scikit-image), it provides good boundary segmentation, the size of superpixels is generally small and regular with the number of segments usually around 10-20k. The input parameters also provide control of
the scale of the local density approximation and allow the user to select a level in
the hierarchical segmentation that is produced by the maximum distance between
nodes in the quick-shift tree.

**Watersheds** [90] A superpixel method that is based on the watershed method.
Watershed is a transformation defined on a grayscale image which is often used
for segmentation purposes. This method converts pixel to height based on bright-
ness and generates catchment-basin superpixels using the local minimum. Similar
approaches are developed in [91, 92]. The method is relatively fast, but does not
provide control of the number of superpixels.

**TurboPixels** [93] Uses a level-set based geometric flow to generate superpixels
by dilating the given number of superpixels. The method relies on local gradients to
generate evenly-sized superpixels. The method provides uniform sized superpixels
and offers control over the superpixel shape and density.

**SLIC** [94] This algorithm uses a distance function to assign each pixel to
a cluster; the distance measure is based on the 5D space l, a, b (colour), x, y
(position). Instead of using Euclidean distance, SLIC calculates the colour and
position separately and applies a weight on each side. Once all pixels have been
assigned to each cluster, the program recalculates each cluster centre by using the
mean of pixels in 5D. This loop carries on until the L1 distance between previous
cluster centres and recomputed cluster centres are below a certain threshold (in the
code provided by the author 10 iterations are run as prescribed by the algorithm).

SLIC improves process speed by initialising the cluster centre in a divided
rectangle based on the number of superpixels setting. This also restricts the pixel
size and shape because some of the superpixels have straight edges which affects
the segmentation results.

### 2.8 Using Superpixels with Temporal Information

Utilising temporal information is another problem when segmenting objects from
time series images. For this reason we have looked into video segmentation algo-
rithms. However, there are several differences between time series image and video
data.
One of the biggest differences is the variation in time between each frame. In time series images, the time between each frame can vary between a few seconds to a few days or years, and during this period the environment may change dramatically and make it more difficult to track the target object. Under such environmental conditions, the optical flow might not perform properly due to its two key assumptions: colour constancy (or brightness constancy) and small motion (points do not move very far).

As described in the phenotyping section, temporal information has been used to track leaves via fluorescence video \cite{40} and for observing plant growth under different light conditions \cite{39}. Additionally, a few algorithms use superpixels to provide segmentation without using optical flow.

Temporal Superpixels (TSP) \cite{95} uses a modified SLIC algorithm to generate superpixels in each frame and a likelihood function to perform split and merge actions between pixels. The time cost to process each frame is constant since it only considers the change between two frames. Multiple Hypothesis Video Segmentation (MHVS) \cite{96} uses a mean shift to generate superpixels and a conditional random field over all frames to perform split and merge actions.

However, the three major problems to solve are: creation, continuation and termination of labels; spatial and temporal long-range relationships between pixels; and the optimal number of clusters.

\section{2.9 Supervoxel}

A voxel is a single point in three-dimensional space often visualised as a small cube in the same way that pixels are visualised as small squares. Similar to a superpixel, a supervoxel represents a group of voxels that share similar attributes in a three-dimensional space. Video or images with depth or temporal information can also be treated as a three-dimensional space by stacking all images and using time as the third dimension. Supervoxel algorithms have been developed to segment 3D space or point clouds into regions \cite{97,98}. For time series images and videos, we can treat the dataset as a 3D space by using time as the third dimension. Similar work has been done processing video with supervoxels \cite{99}. 

23
2.10 Segmentation Evaluation

To measure the performance of segmentation methods, there are many evaluation techniques, both supervised and unsupervised, developed for image segmentation. All these evaluation methods can be divided into many categories based on subjective and objective evaluation [100, 101]. A set of criteria for plant segmentation can also be found in [102].

For supervised evaluation, the segmented image is compared to the reference image, also known as the ground truth. Ground truth images are manually segmented images, based on the subjective view of the individual performing the segmentation. The process of manually creating such ground truth image is tedious, subjective and time-consuming.

The unsupervised evaluation avoids human interaction, and instead performs results comparison between two different segmentation methods or different parameter sets of a single method. It may allow for self-tuning of parameters to perform better segmentation based on the evaluation results.

Most evaluation methods return a score based on how well one segmentation performs compared to another. The evaluation can be based on intra-region uniformity, inter-region disparity or semantic cues (summarised based on the definition of good segmentation [103]).

For image segmentation, there are several types of measures for evaluation metrics. One of the commonly used metrics is Receiver Operating Characteristic (ROC) sensitivity [104]. Each positive and negative segment prediction can be calculated to produce a plot of sensitivity against specificity. A good segmentation method aims to achieve high sensitivity and low specificity.

More in-depth consideration of segmentation evaluation metrics is presented in Chapter 3 below.
Chapter 3

Experimental Setup & Datasets

3.1 Introduction

This chapter describes the experimental datasets in detail. The datasets used in this project are from the National Plant Phenomics Centre (NPPC) and an experimental system. Each dataset has a different plant type and setup which is described in the following sections. Manually made ground truth and evaluation methods used to evaluate segmentation results are also described in this chapter.

3.2 The Datasets

The datasets used in this work represent a set of images featuring the growth of one or more plants within the same environmental conditions. The datasets come from two sources: (a) an experimental framework made for this thesis and (b) the National Plant Phenomics Centre (NPPC). We sought to obtain a wide range of samples by selecting datasets containing different plants, viewpoints and time intervals between frames. Since none of the datasets contain any segmentation masks for evaluation, a subset of 70 images were manually segmented as the ground truth to evaluate the results.
Figure 3.1: Top-down view of two one-month-old Arabidopsis: (left) wild-type (col0) and (right) mutant (ede1-1).

3.2.1 Choice of Plants

For our experimental setup, we chose Arabidopsis as our model plant, specifically, the widely used species called “Arabidopsis thaliana”. Arabidopsis belongs to the family of brassicaceae and their leaves are alternate (i.e., are in a spiral arrangement). The type of Arabidopsis used in this project are col0 (wild-type) and ede1-1 (mutant). Figure 3.1 shows both samples after a one-month growth period taken by a webcam. Viewed with the naked eye, col0 has a larger leaf size and more leaves than ede1-1: this might be good information to be able to distinguish them via computer vision.

Additionally, we also chose oilseed rape and oats in addition to Arabidopsis to add variety to the test. Figure 3.2 shows both samples of oilseed rape and oats. Both plants are much larger and taller than Arabidopsis, and, for this reason, top-views provide much less information on the growth of the plant than side-views. Frames were also required to hold the plant and prevent leaves from falling.
3.2.2 Experimental Framework for the Webcam Arabidopsis Dataset

We established a simple framework to take time-lapse images of the plants. The setup was inspired by [18] which uses a low-cost camera to take top-down view images. In our case, we used a standard consumer webcam (by Logitech) with programmed code to automatically take time-lapse images. Figure 3.3 shows the setup of our experimental framework, a clamp stand was used to support the webcam above the Arabidopsis.

Usually, top-down views of the plant are less useful than side views, due to the difficulty of recording plant height from a top-down view. But in this project, we mainly focussed on early-stage leaf growth (that is to say, growth before flowering). Hence the height of the plant was less important.

This experiment was conducted in a growth room to make sure all plants inhabited a controlled stable environment. The plants were illuminated by fluorescent lights for 8 hours per day and all photos were taken during this period. Since all lights were turned off during the night, night images were totally black and not
Figure 3.3: Framework setup for taking top-down view Arabidopsis images in a growth room.

used in our work.

The plant tray contained 48 (6*8) Arabidopsis plants which were divided into two genotypes (wild-type and mutant) and placed in a chess board pattern with one genotype on the white grid and the other genotype on the black grid. This layout allowed plants to share light evenly and minimised leaf overlap during the growth process. Cameras started taking images from week 3 and ceased doing so at week 12 (flowering). The plants were manually watered every day, which caused some noise in the form of a few water drops around plants and trays, and occasional tray motion. We collected multiple datasets with the same experimental setup, and selected datasets with better lighting and with all plants surviving for the duration of the data collection for this project.

The webcam took one image for each time-lapse of 15 minutes in JPEG format using the *RGB* colour space and a resolution of 1600*1200 pixels. A total of 3000 images were taken from seed to flower for each dataset. Figure 3.4 shows sample images at four growth stages. We chose to focus on half of the images collected between week 3 and 8 since plants before week 3 were too small to identify and beyond week 8 they started to overlap and become over-grown. From this point onwards, this dataset is referred to as the “webcam Arabidopsis dataset”.

There were many issues raised by webcam Arabidopsis dataset which increased
segmentation difficulty, and these are discussed in the next section:

- Small tray movement after watering.
- Images taken after watering contained water drops.
- Plant pots and part of the tray shared a similar colour with the plants.
- Parts of plants were not being captured by the camera.
- Use of the uncalibrated camera resulted in blurring at the edge of images.
- Some of the plants died at an early stage in the data capture process.

### 3.2.3 Datasets from NPPC

The source of the other datasets is the National Plant Phenomics Centre (NPPC). Plant images from the NPPC are taken by LemnaTec (LT) system with an RGB camera. All plants are controlled by the LemnaTec system in a stable environment and are automatically watered every day.

Plant images are automatically taken by the LemnaTec system daily. The plant is first moved in to a small chamber with a white background and a single high resolution colour camera (visible light) is used to take images of the plant from fixed angles. The system is capable of taking plant images from multiple angles including top-down view and side-view. Based on the plant type, we choose to take the top-down view images for small and short plants such as Arabidopsis, and take side-view images for larger plants such as Oats.

The LemnaTec datasets contain both top-down and side view images of different plants. We have focused on three datasets including top-down view Arabidopsis (LT Arabidopsis), side view oats (LT oats) and side view oilseed rape (LT oilseed rape). Each selected dataset contains different visual properties in addition to the plants. The LT Arabidopsis contains multiple trays above a belt, and a few reflective objects such as screws, water and a reflective label. The LT oats and LT oilseed rape have a blue frame and reference marker, the light source focused at the centre of the image and dimming around the edge. Figure 3.5 shows sample images of the three datasets used in this work.
Figure 3.4: Webcam Arabidopsis images in four growth stages. (a) 3-week-old plants, the system starts to record the plant images, each plant has less than 500 pixels; (b) 6-week-old plants; (c) 8-week-old plants, occlusion starts to appear; (d) 11-week-old plants, plants start to flower and it becomes very hard to identify individual plants.
Figure 3.5: Sample images of three selected datasets from the LemnaTec System. (a,b) Day 1 and Day 12 Arabidopsis; (c,d) Day 10 and Day 50 Oats; (e,f) Day 1 and Day 12 Oilseed rape. The day count used here represents the time when the system started to record plant images and is not the actual plant age.
Table 3.1: Plant datasets summary.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>View Point</th>
<th>Plants per Image</th>
<th>No. of Images</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam Arabidopsis</td>
<td>Top-down</td>
<td>48</td>
<td>850</td>
<td>1600*1200</td>
</tr>
<tr>
<td>LemnaTec Arabidopsis</td>
<td>Top-down</td>
<td>5</td>
<td>12</td>
<td>2454*2056</td>
</tr>
<tr>
<td>LemnaTec Oats</td>
<td>Side</td>
<td>1</td>
<td>56</td>
<td>2056*2454</td>
</tr>
<tr>
<td>LemnaTec Oilseed Rape</td>
<td>Side</td>
<td>1</td>
<td>12</td>
<td>2056*2454</td>
</tr>
</tbody>
</table>

3.2.4 Time-Lapse and Time Series

Based on the way of the images were captured, the datasets are divided into two categories, time series images and time-lapse images. Time-lapse images imply the scene does not change which applies to webcam Arabidopsis datasets. Time series images mean sequence of images graphed in time order which is represented by the LemnaTec datasets.

One difference between time-lapse and time series images can be referred to the sudden rotation of plants between images from LemnaTec dataset, which is caused by the vibration of the trays when they were transported on the belt. This may affect the segmentation results for methods relying on the position or movement of the plants.

3.3 Data

Table 3.1 shows the datasets that our work focused on. The webcam arabidopsis image dataset was originally stored in an RGB colour space in the JPEG format. This caused blurring of the plant edge because of the lossy compression. To prevent further repeated compression (where images maybe modified and saved multiple times during the segmentation process), all images were first saved in the lossless PNG format, and then converted into other colour spaces. We captured more images and datasets during the data acquisition stage; however, only datasets with one or more ground truths were selected in order to provide detailed evaluation. All LemnaTec datasets were originally stored in PNG format.
3.4 Evaluation

With the ground truth, one way to evaluate the segmentation is to calculate the total number of true and false positives identified at a pixel level, and precision of the image, which is also known as the positive predictive value. We also use Receiver Operating Characteristics (ROC) with ground truth to evaluate our segmentation results.

3.4.1 Ground Truth

The ground truth was used in the evaluation plan as the “standard results” in order to verify that the pixels detected by the system were correct. The ground truth generally refers to information that is collected “on location”. In image processing, this is represented as a series of interesting or correct pixels that can be used to relate or calibrate with the image data. The ground truth can also be used to mark the objects in a training set for supervised learning.

For our dataset, the ground truth is a binary image data set manually generated from the colour image data set (Figure 3.6). The binary images have the same dimensions as their original images, all plant pixels are marked as black pixels and non-plant pixels are marked as white pixels corresponding to the colour images. These ground truth images can only be used to verify the results with their corresponding (original) colour images. Therefore, every single testing image requires a ground truth in order to evaluate all results.

Each ground truth image set was created manually by the same individuals, using an image editing tool (GIMP). We did not manually create the ground truth for every image in the dataset since this is very time consuming; instead we created a few ground truth images for each growth stage in the dataset. For the Webcam Arabidopsis set, we divided the whole dataset into seven growth stage sets and created 10 ground truths of each set to form a total of 70 ground truth images. For the LemnaTec dataset, we created three ground truths from each dataset (the first, the middle and the last image). The ground truth does not cover every image; however it greatly reduces the amount of manual work but still retains the testing data coverage. Compared to a dataset with full ground truth, datasets with partial ground truth provide a less accurate evaluation of plants in three different growth stages, but we can still estimate the results of the missing ground truth since the
plant grows constantly.

### 3.4.2 Receiver Operating Characteristics (ROC)

The ROC curve is applied to examine the performance of the system. The ROC is also known as a Relative Operating Characteristic curve: the curve is plotted using a true positive rate (TPR) against a false positive rate (FPR) to show the performance between selectivity and sensitivity \[^{[105]}\].

To evaluate a segmentation method, the segmentation results are compared with the ground truth, then the outcomes are labelled either positive (P) or negative (N). Specifically, the system produces result images in binary form that predict and mark all plant pixels, and these results are compared with the ground truth to produce the outcomes. This problem can be considered as a binary classification, and there will be four possible results:

- **True Positive (TP) (hit)** is the total number of pixels that are correctly predicted as plant.

- **True Negative (TN) (correct rejection)** is the total number of pixels that are correctly predicted as non-plant.
- False Positive (FP) (false alarm) is the total number of pixels that are incorrectly predicted as plant.
- False Negative (FN) (miss) is the total number of pixels that are incorrectly predicted as non-plant.

The TPR and FPR can be calculated after the classification is done by using the formulae (3.1) and (3.2). TPR shows the sensitivity of the system: it defines how many correct positive results (TP) occur among all positive (P) samples. On the other hand, FPR defines how many incorrect positive results (FP) occur among all negative (N) samples.

\[
TPR = \frac{TP}{P} = \frac{TP}{(TP + FN)} \quad (3.1)
\]

\[
FPR = \frac{FP}{N} = \frac{FP}{(FP + TN)} \quad (3.2)
\]

All obtained data can be plotted to form the ROC curve in an ROC space which is defined by FPR and TPR as x and y respectively. The ROC graph is also called the sensitivity vs (1 - specificity) plot. The ROC curve always starts from (0, 0) and ends at (1, 1). The results should always be better than a random classifier or otherwise the polarity of the classifier output can be swapped in order to improve the system performance.

Figure 3.7 shows a sample of the ROC curve. The green line is the ROC curve which has an accuracy rate of 0.80 where it interacts with the black line. The red line represents the random guessing line with an accuracy rate of 0.50, any data below the red line is considered to be worse than random and the polarity of the classifier output is required to be swapped in order to improve the system performance.

Other statistical measures can also be calculated to check the performance of the classification. Precision works best for evaluation of images with a large background and small objects. Jaccard measures dissimilarity between sample which also provide good measurement to the results.

\[
Accuracy = \frac{(TP + TN)}{(P + N)} \quad (3.3)
\]

\[
Precision = \frac{TP}{(TP + FP)} \quad (3.4)
\]
Figure 3.7: Sample of an ROC curve. The green line represents the ROC curve; the red line represents the random guessing line; the black line indicates the accuracy level.
Recall = \frac{TP}{TP + FN} \quad (3.5)

Jaccard = \frac{TP}{TP + FP + FN} \quad (3.6)

Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (3.7)

The evaluation of the system is checked by the ROC curve and all the values with all datasets. The system is considered as usable (pass) if the overall precision exceeds 0.80.

3.4.3 Conclusion

This chapter presents the setup of a plant image acquisition system and describes the datasets and the method to evaluate the segmentation results. The system is capable of providing the full life cycle of plants in stable time-lapse images from a fixed view point. The system provides lower resolution data in comparison to the LemnaTec system, but it is low cost and easy to deploy. Four datasets were selected to be used in the development of segmentation method, including one dataset from the developed system and three datasets from the LemnaTec system. A number of ground truth images were also created to evaluate the segmentation methods. The precision, recall and Jaccard values are focused on evaluating the performance of segmentation results.
Chapter 4

Exploring Colour Spaces for Plant Segmentation

4.1 Introduction

This Chapter examines the use of colour in plant imaging, by analysing pixels in plant image datasets and real plants using a spectrometer to study the colour differences between plants. Then we experiment with different colour spaces to segment plants from background in images to determine which colour space is best for further processing.

4.2 Analysing the Plant Colour

Before segmenting the plant pixels from the images, we first need to understand plant colour, which contains valuable information which can be used to analyse the plants status if a correct colour space is selected. A right choice of colour space should improve the performance of image segmentation.

However, there does not exist a single colour space that works for all image segmentation problems. To find a suitable colour space for our plant dataset, we would first like to know whether the plants colour changed over time and whether this would affect colour-based methods (segmentation).

As mentioned in Chapter 3 all plant datasets are collected under a controlled environment, which is to say that all images are taken with the same light source.
which does not affect colour change of the plant itself.

Plant colour changes often appear during plant stress (and/or) over a long observation period. Taking the webcam Arabidopsis dataset as an example, there is a clear division in the greenness of plant pixels between two Arabidopsis genotypes and their colour changes over time through visual inspection. This is mostly due to specular light reflection and shadows from other leaves making it harder to define the plant colour even under a controlled environment. However, the actual values from the $RGB$ colour space do not show enough difference to separate the two Arabidopsis genotypes (Figure 4.1).

### 4.3 Choice of Colour Space

This section examines different colour spaces and compares them to discover which one is the best choice for plant colour analysis. We decided to compare the performance between seven classical colour spaces against a selection of plant images from our datasets with a view to determining which colour space is best for separating plants and background. The tested colour models are shown below.

- $RGB$ - red, green, blue.
- $HSL$ - hue, saturation, lightness.
- $HSV$ - hue, saturation, value.
- $Lab$ - luminance, ranges from green to red, ranges from blue to yellow.
- $Luv$ - Provides similar results to the $Lab$ colour space, they are both transformations of the $XYZ$ colorspace.
- $XYZ$ - $Y$ is luminance and $XZ$ represent all possible chromaticities.
- $YCbCr$ - luma, blue difference, red-difference.

There are approximations since the colour models are converted from RGB colour space, and some of the colour channels are scaled to fit the 0 to 255 range for easier comparison.
Figure 4.1: Plant colour values in RGB colour space with an error bars from the webcam Arabidopsis dataset in seven different time periods to show that the plants colour changes over time. (a) Plot of R, G and B value of all plants from the webcam Arabidopsis dataset; (b) Plot of R, G and B value of wild-type and mutant Arabidopsis from the webcam Arabidopsis dataset.
4.3.1 Separation of Plant from Background

To test which colour space works best with our datasets, we need to determine whether the colour space is capable of separating the plant pixels from all other objects based on colour alone. Thresholding is used here to test the performance of each colour space on a test dataset which consists of three images in different growth stages from each plant dataset. This is a very basic classifier but we assume a good performance in this simple test will lead to better performance with a more sophisticated classifier.

The thresholding method separates all image pixels into two groups ($G_1$ and $G_2$) using the threshold value ($T$) based on the given channel of the colour space. $G_1$ consists of all pixels with channel value $> T$ and $G_2$ consists of pixels with value $\leq T$. The test repeats from $T = 0$ to $T = 255$ with an increment by 1 for each iteration.

The segmented results from each loop are compared with the ground truth to calculate its True Positive Rate (TPR) and False Positive Rate (FPR). All data are then used to form the Receiver Operating Characteristic (ROC) curve which determines whether the colour channel can be used to provide a good predictor or not.

Figure 4.3 shows the ROC curve of each colour space. Points close to (0, 1) are considered good classifiers, and an ROC curve which contains most points below the random guess line (the diagonal line which divides the ROC space) is considered poor.

For a lot of plant images, especially taken in the early growth stage, the number of background pixels is significantly greater than the number of plant pixels, and this makes it harder to judge the results by accuracy, precision and recall because there are insufficient foreground plant pixels to compare with, and the results would always be very high despite the number of any incorrectly predicted plant pixel. In this case, the Jaccard and Dice prove valuable in the evaluation of the results as these measures take sample size into account.

Although many channels produce good results, we did not consider using a hybrid colour space since it would contain different colour components and be difficult to represent. We chose a Lab colour space for our work because it provides the best model to segment plant from background in comparison with other colour
Figure 4.3: ROC curve of segmentation results on three channels from minimum value to maximum value for seven colour spaces. (a,b,c) RGB colour space; (d,e,f) HLS colour space; (g,h,i) HSV colour space; (j,k,l) Lab colour space; (m,n,o) Luv colour space; (p,q,r) XYZ colour space; (s,t,u) YCbCr colour space.
4.4 Spectrometer Data

Changes in camera viewpoint may result in small plant colour changes because of light reflectance variation on the plant leaf. This section attempts to separate specular and diffuse reflection from plant leaves, and to observe the colour difference with and without specular reflection to determine whether the specular reflection would affect plant segmentation.

We recorded hyperspectral measurements from the Arabidopsis before they started flowering (Figure 4.4) by using an Ocean Optics Dual Channel Spectrometer (S2000 Miniature Fiber Optic Spectrometer). The device contains a 'Y' design fibre optic (ref QR 400 7 VIS BX) which connects to a light source and a leaf holder. The leaf holder is a clip-shaped adapter with two observation slots, including at 45° and 90° angles from the focus point, to obtain a different reflection from the test plants (Figure 4.5). During data collection, the leaf holder was always set at the 45° slot in order to capture only diffuse spectrum data from the plant leaf.
Each spectrum reading contains reflectance value data taken against wavelengths between 330nm and 1030nm. Before taking plant measurements, a dark and a reference spectrum were recorded for data calibration by taking a reading with the light off and with a spectral standard white colour block (as standard black and white records for calibration). While taking sample readings, five measurements were collected from different leaves for each Arabidopsis plant. We tried to take readings without damaging any leaves so that the original experiment could continue.

To analyse the data, the following formula were applied to each sample spectrum data (4.1) in order to calculate the percentage reflectance in each wavelength.

\[
\text{Reflectance(\%)} = \frac{\text{Sample} - \text{Dark}}{\text{Reference} - \text{Dark}} \times \text{ReflectanceOfSpectralon} \tag{4.1}
\]

The mean and standard deviation of all samples were plotted and demonstrated that the spectral reflectance between wild-type and mutant plants were almost identical (Figure 4.6a). Although mutant Arabidopsis exhibited slightly higher near-infrared (NIR) reflectance than wild-type, the difference in the visible light range between 400nm and 700nm was negligible. However, the mutant Arabidopsis plants appeared to have higher reflectance distribution in the visible light range.
and lower in NIR, the opposite to the wild-type Arabidopsis (Figure 4.6b). The similarity in diffuse spectrometer readings confirms our suspicion that the visible colour variation between two Arabidopsis plants appears to be in their specular reflection ability.

### 4.4.1 Diffuse Images

The spectrometer data shows that both Arabidopsis genotypes grown in the same environment share the same colour when there is no specular reflection on leaves, therefore the two genotypes have a different level of light reflectance which may also vary from different samples. To be able to prove this from our experiment, we first need to generate diffuse images from original images. However, the best way to obtain a diffuse image is via a polarizing filter and illumination, techniques not considered for use in our experimentation prior to camera setup.

An alternative solution is to use the method pioneered by Tan which allows separating reflection components from images without a polarizing filter [56]. The specular-to-diffuse mechanism first generates specular-free images by setting a fixed diffuse maximum chromaticity value and applying it to all pixels, and then assigns a property value to each pixel to indicate whether it is a diffuse pixel and iterates over them in order to decrease the intensity of specular pixels.

As a result, the generated images have a significantly different appearance when compared to the original image (Figure 4.7). Most of the highlighted specular pixels have been removed and the plants appear more intense. One problem is that the method cannot deal with achromatic pixels, and therefore the grey grid disappeared in the diffuse image. However, this does not affect the analysis of plant pixels.

To further analyse the data, the mean and standard deviation are each calculated from the RGB colour space (Figure 4.8). The overall decrease of mean values in the diffuse image demonstrates that the specular reflection strongly affects the colour of plant pixels. However, it is difficult to find the colour variation between wild-type and mutant arabidopsis in both the original image and the diffuse image,
(a) Mean percentage reflectance with errorbars for all Arabidopsis.

(b) Mean percentage reflectance plotted for each Arabidopsis plant.

Figure 4.6: Spectrometer data for both wild-type and mutant arabidopsis.
Figure 4.7: Separating reflection components in the arabidopsis image using specular-to-diffuse mechanism. (a) The original webcam Arabidopsis image; (b) The specular-to-diffuse method on first application; (c) The diffuse image as the output; (d) The specular image upon subtraction of the diffuse image from the original image.
especially in the later growth phase. This result is consistent with our expectations based on the spectrometer data.

Some issues arose during the process which had a minor impact on the diffuse image:

- The specular reflection can also be changed by the cameras viewpoint and the intensity of the light source.
- The specular-to-diffuse mechanism relies on achromatic pixels to find specular reflection, and this might not work on plant datasets with gray background objects.
- The data from the spectrometer and camera image are taken from different experiments at different times, and moreover one has been recorded at a single point in time while the other has been recorded over time.

### 4.4.2 Other Images

Apart from regular cameras, we also collected the image set with Kinect (Figure 4.9) which additionally records depth information (RGB-D images). Regardless of the performance of its RGB camera, the resolution and accuracy of the depth camera do not satisfy the requirements of our experiment. Over time, the growth of Arabidopsis increases the complexity of the model shape. The size of the Arabidopsis plant is also too small to be accurately detected. One advantage of the depth camera is that it functions normally during the night, but the disadvantages of resolution and accuracy mean we did not pursue this capture set up.

### 4.5 Conclusion

We performed a number of experiments to demonstrate the colour of the plants changing over time; however, the colour variation of green appeared to be small. Additionally, we investigated the performance of different colour spaces by using a supervised segmentation method. We chose to use the Lab colour space for plant segmentation since it has overall the highest sensitivity and lowest specificity in
Figure 4.8: Mean and standard deviation of plant pixels in RGB colour space using Tan’s algorithm.
Figure 4.9: Arabidopsis plant image taken by Kinect.
plant image segmentation. Study of the process of separating specular reflection from the plant and the creation of diffuse images also proved that plant colour change was influenced by specular reflections.
Chapter 5

Segmentation Using Colour and Dynamics

5.1 Introduction

Building on the results of the colour space investigation from the previous chapter, this chapter presents a number of experiments carried out using different clustering methods. By finding ways of identifying plant clusters, we developed an unsupervised colour-based clustering method to segment plants from time series images based on temporal information and the assumption that plants grow over time. The overall algorithm is shown first, followed by a comparison and evaluation of various choices and design decisions.

5.2 The Method

Based on the work on plant colour and plant growth, we were able to develop a segmentation method that relies on the colour and size of the foreground object, with an assumption of size change based on the growth of the plants.

Figure 5.1 shows a walkthrough of the algorithm. A sequence of time series images with at least one plant in each picture is the only requirement for the segmentation. Once the images are selected as an input dataset, the dataset is used to train a set number of clusters by a clustering algorithm, and then each cluster is labelled as either a plant or non-plant cluster based on the cluster size
Figure 5.1: A walkthrough of the developed algorithm. The algorithm contains three steps: 1. train a Gaussian Mixture Model using a set of time series images in a given colour space ($\text{Lab}$); 2. label each cluster as foreground or background based on the change in cluster size across the time series; 3. separate foreground and background based on the growth of clusters over time.

across the time series. Finally, the method merges all plant clusters to segment them from the background.

5.2.1 Clustering in Colour Space

First, all images were converted into Lab colour space, and each value across all channels was normalised by scaling them between 0 and 255. Then all images were downsampled to produce a smaller training set. A clustering model was then trained based on these pixel data using a clustering algorithm with 3 clusters. We chose to start with 3 clusters because it was the lowest possible number of clusters to represent a basic plant image: we assumed all images contained plants grown under a controlled environment. Therefore, these clusters had the capacity to classify plant, soil and other backgrounds (such as pot, or tray). However, images usually contained more complex objects, so the initial setup had low accuracy in
separating plants and background.

To generate a better classifier, the second step was to identify the near optimal number of clusters by repeating the above process with an increased number of clusters until the following conditions were met: all clusters could provide modelling of the data to represent each object; and adding more clusters would not improve the representation.

The selected model was then generated by retraining all pixels with the near-optimal number of clusters, and all objects were separated into each cluster. However, it was difficult to determine which cluster contained a plant using only colour, since not all the plants were green. The method required additional information to complete the segmentation.

5.2.2 Choice of Plant Cluster/Foreground

The main plant characteristic which changes over time is size (surface area). Figure 5.2 shows a number of plant pixels from a set of time series images over 12 days. The figure shows that the number of plant pixels grows over time at an almost linear rate. This is the assumption we adopted to determine which clusters were plant clusters. At least one cluster must contain plant pixels. When there was more than one plant cluster, the total sum of the plant clusters over time would still be close to a linear growth.

With the trained clustering model, it was difficult to identify plant clusters amongst all clusters, and the plant growth assumption solved this problem. An unsupervised segmentation method could be created to segment plants by monitoring the size of clusters in a set of time series images using a correlation coefficient (such as Pearson's $r$). A cluster with a strong positive correlation indicated that the clusters total number of pixels was growing over time, and could be defined as a foreground plant cluster. Clusters with a small or negative correlation could be defined as background.

To select which clusters corresponded to plants, the correlation coefficient ($r$) function was used to calculate the correlation between time and size in pixels for each cluster. Figure 5.3 shows one of the datasets: cluster 8 has a very strong positive correlation with an $r$ value close to 1 (which represents an almost perfect positive fit). This is defined as a plant cluster.
Figure 5.2: Daily growth of the number of plant pixels grow over a 12-day period before flowering.

The entire method was based on a single assumption that the plant would grow over time. The surface area of the plant increased monotonically, and all images were sorted in time order. This brought one key advantage: the unsupervised segmentation would function regardless of the colour of the plant.

5.2.3 The Assumptions

Before the segmentation, we made some assumptions based on the plant datasets:

1. Each image in the dataset contains at least one plant that needs to be segmented.
2. The dataset has been stored in a sortable chronological order. e.g. by filename.
3. There is visible growth in plant size over time.
4. The time interval between images is constant.
Figure 5.3: The correlation coefficient calculated from the Arabidopsis dataset trained by EM. Cluster 8 is the plant cluster.

5. There are no overlapping plants in the image.

The first two assumptions are reasonable since the method is designed to segment plants using temporal information. The third and fourth assumptions indicate that the method relies on near constant plant growth. And the rationale for the fifth assumption is to prevent plant overgrowth which may affect the plant detection function. Although these assumptions are perhaps somewhat strict, it was possible to obtain datasets using the developed plant image acquisition system and the LemnaTec system. Furthermore, many plant segmentation methods do not function when multiple plants overlap.

Based on the plant datasets, our hypothesis was that each plant and background object share similar colour attributes, so we could group colour pixels using an unsupervised clustering method. We accommodated colour pixels into multiple clusters so that each cluster could be used to represent an object from the dataset. The dataset also contained time information, which meant that, as the plants were growing, so we could find the plant pixels by identifying the clusters
that evinced growth over time.

5.3 Clustering

The first thing to decide for our method was to choose a clustering method that boasted the best performance for plant segmentation. We compared three popular clustering algorithms in this section and selected the algorithm with the best performance on the plant datasets.

5.3.1 Clustering Algorithms

Three clustering algorithms, K-means, Fuzzy C-means (FCM) and Gaussian Mixture Model (GMM), were compared in this study. K-means is a hard clustering algorithm which clusters data points that share the same centroid based on Euclidean distance. FCM is a fuzzy clustering algorithm that is very similar to the K-means algorithm and which groups data points based on a maximum membership value in relation to the centroid. GMM is a probabilistic clustering algorithm that models data points as a Gaussian.

All three clustering algorithms were implemented on the basis of their original test models. The pixel value in Lab colour space was used to train and generate clusters for evaluation.

5.3.2 Clustering Results

To test each clustering algorithm, all cluster models were trained using a fixed number of clusters from the plant datasets. A cluster size of \( k = 7 \) was manually selected here to represent each cluster as either plant, soil, pot, grid, moss or background-supporting frame from the dataset. We then manually selected the plant clusters and evaluated the results.

Table 5.1 shows the segmentation results for the three clustering algorithms on the webcam Arabidopsis dataset with the same number of clusters. Both FCM and GMM demonstrated good results on plant segmentation, whilst K-means showed good precision but was adversely influenced by the noise. GMM showed the best
Table 5.1: WebCam Arabidopsis segmentation results with different clustering algorithms.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>Jaccard</th>
<th>Dice</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>0.90</td>
<td>0.59</td>
<td>0.56</td>
<td>0.72</td>
</tr>
<tr>
<td>FCM</td>
<td>0.88</td>
<td>0.71</td>
<td>0.65</td>
<td>0.79</td>
</tr>
<tr>
<td>GMM</td>
<td>0.91</td>
<td>0.71</td>
<td>0.66</td>
<td>0.80</td>
</tr>
</tbody>
</table>

overall performance across all clustering algorithm. For this reason, we selected GMM as the clustering algorithm for our method.

5.4 Subsampling

Training the colour models is computationally expensive and infeasible on the entire datasets (millions of pixels). Thus we need to select a smaller training dataset. Efficient plant segmentation requires a choice of a good training subsample and model selection criteria. This section tests the performance of different types of subsamples generated from the same dataset in model training and compares their efficiency in clustering.

5.4.1 The Subsamples

We compared different subsampling methods on our data to further investigate the stability of the algorithm and reduce processing time by using as few data points as possible. A single sample data, in this case, refers to a pixel which is composed of three parameters in Lab colour space. The subsamples were selected as shown below:

1. Gaussian Pyramid - Downsampled (Pyrdown) image. This was performed by using the Gaussian pyramid function to downsample images reducing the image size by 1:256 (e.g. from 1600*1200 to 100*75).

2. Evenly Distributed Points - Every 200th pixel from all images. To simplify comparison with other subsamples, we reduced images by 1:256 by repeatedly removing pixels from every other row and column.
3. Every tenth frame - Every 10th frame starting from the first frame of a dataset.

4. Random - A random number of samples were selected from each image. This subsample required multiple runs for an accurate result.

5. Tan’s [56] - A generated subsample based on Tans method to produce a diffuse-only image dataset.

5.4.2 Subsampling Results

Five subsamples were generated from the same dataset. The GMM and the EM algorithms with a fixed number of clusters \( k = 7 \) were used here to compare their performance. Clusters containing most plant pixels were labelled as the plant clusters and were manually selected to create segmentation images and evaluated against the ground truth.

Table 5.2 shows the segmentation results with different subsamples. Both the full dataset and Tans subsample trained clusters with all pixels sharing the same results; however, training the model with all pixels is time-consuming, and data overfitting also reduces its performance. Tans subsample converted all images to diffuse-only images but did not provide an improved result. The Gaussian Pyramid subsample showed the worst performance: it trained on downscaled pixels values that did not exist in the dataset, and this resulted in a bad model. The performance of the random subsample was unstable and difficult to replicate: it produced a bad result when the plant pixels were not fully covered. Both the evenly distributed subsample and every-10th-frame subsample demonstrated the best performance with good data sample coverage. We selected the evenly distributed points method since it provided small but evenly covered samples and had the overall best performance.

5.5 Model Order

All previous experiments were conducted with a manually defined number of clusters \( (k) \). However, the optimum size of \( k \) varies between different datasets. To automate this process, an algorithm to estimate \( k \) was necessary. This section
Table 5.2: LemnaTec Arabidopsis segmentation results with standard deviation with different subsamples.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>Jaccard</th>
<th>Dice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Dataset</td>
<td>0.99 (0.00)</td>
<td>0.87 (0.02)</td>
<td>0.87 (0.02)</td>
<td>0.93 (0.01)</td>
</tr>
<tr>
<td>Gaussian Pyramid</td>
<td>0.29 (0.15)</td>
<td>0.98 (0.00)</td>
<td>0.28 (0.14)</td>
<td>0.42 (0.17)</td>
</tr>
<tr>
<td>Evenly Distributed</td>
<td>0.98 (0.00)</td>
<td>0.95 (0.01)</td>
<td>0.94 (0.00)</td>
<td>0.97 (0.00)</td>
</tr>
<tr>
<td>Every 10th Frame</td>
<td>0.98 (0.00)</td>
<td>0.95 (0.01)</td>
<td>0.94 (0.00)</td>
<td>0.97 (0.00)</td>
</tr>
<tr>
<td>Random</td>
<td>0.30 (0.15)</td>
<td>0.95 (0.01)</td>
<td>0.29 (0.14)</td>
<td>0.43 (0.17)</td>
</tr>
<tr>
<td>Tan’s Subsample</td>
<td>0.99 (0.00)</td>
<td>0.87 (0.02)</td>
<td>0.87 (0.02)</td>
<td>0.93 (0.01)</td>
</tr>
</tbody>
</table>

examines different model order selection criteria and compares their performance on the plant dataset, and on synthetic data.

5.5.1 Model Order Selection Criteria

Thus far the number of clusters \(k\) has been defined manually. Therefore, to segment the plant, we need to train the GMM model using EM with a small sample size selected from the data (using one of the subsamples) with a range of \(k\). We manually scrutinised each result to decide the optimum \(k\) with current settings and dataset. To automate this step, we used the model order selection criteria to evaluate the trained model, and the optimal \(k\) can be selected by comparing the criteria value. We used a standard EM function to simplify our task, and the function used the K-means algorithm for initialization.

The method creates a fixed number of clusters based on a predefined parameter. However, the size of \(k\) is not constant for different data. We used a model order selection method here to find the optimal \(k\), and a cluster split and merge method to further improve the cluster quality. The method merged clusters when the mean and covariance were very close to each other based on some model selection criteria. Here we compare four commonly used model order selection criteria: Akaikes Information Criterion (AIC), Bayesian Information Criterion (BIC), Minimum Description Length (MDL) and Integrated Completed Likelihood (ICL) criteria.
Using Synthetic Data to Test Model Order Criteria

We tested the performance of each criterion here using synthetic test data generated from four Gaussians with known mean $\mu_k$ and covariance $\Sigma_k$ (a diagonal matrix is used here). A total of 120 samples were generated (with 30 from each Gaussian), using function \texttt{munrnd($\mu_k$, $\Sigma_k$, n)} from Octave\(^1\) which draws \(n\) random d-dimensional vectors from a multivariate Gaussian distribution with the input mean and covariance matrix. Therefore, all samples were randomly generated as d-dimensional vectors from a multivariate Gaussian distribution. We used \(d = 3\) here since the real plant data samples were inputted as 3-dimensional data (Lab colour space). The covariance was set to $\Sigma = 0.5$ for all four Gaussians and the means $\mu_k$ were given by:

- $\mu_1 = (0, 0, 0)$
- $\mu_2 = (8, 8, 8)$
- $\mu_3 = (2, 3, 4)$
- $\mu_4 = (-2, -3, -4)$

We then ran GMM with the test data from $k = 1$ to $k = 30$ and found each criteria value as shown in Figure 5.4. However, in the case of large cluster numbers, some models would have zero weight. In our test, we determined that $k = 10$ was the threshold for this phenomenon. All criteria show that $k = 4$ was the optimal number of clusters for the test sample (which was expected).

There was an issue when two sample Gaussians had very close means, or the covariance was too large: the criteria would merge some of the Gaussians into one. To illustrate by way of an example, if we replaced samples from $\mu_2 = (8, 8, 8)$ with $\mu_2 = (4, 4, 4)$ which rendered it very close to $\mu_3 = (2, 3, 4)$. The criteria value would then select $k = 3$ as the optimal value (except Akaike Information Criterion (AIC), since it does not take sample size into account, and, as a result, is mostly affected by the likelihood and the complexity of the model) as shown in Figure 5.5. Each criterion was also tested with five and six different Gaussians and all show correct results.

\(^1\)Octave version 3.8.1 [Online] https://www.gnu.org/software/octave/
Figure 5.4: Model selection results from four criteria for test samples generated using four Gaussian distributions.

Figure 5.5: Model selection results from four criteria for test samples generated using four Gaussian distributions with close means.
Test Data with Different Covariance

To test the model with different criteria, we added two new Gaussians to each with 30 samples. The mean and covariance are now given by:

- $\Sigma = 0.5, \mu_1 = (0, 0, 0)$
- $\Sigma = 0.5, \mu_2 = (2, 3, 4)$
- $\Sigma = 0.5, \mu_3 = (-2, -3, -4)$
- $\Sigma = 1.0, \mu_4 = (4, 4, 4)$
- $\Sigma = 1.2, \mu_5 = (-1, -1, -1)$

A total of 150 samples from five Gaussians were trained in GMM using diagonal covariance (COV_MAT_DIAGONAL). The result is shown in Figure 5.6. Apart from AIC (selected $k = 5$), all other criteria selected $k = 3$ as the optimal number of clusters.
5.5.2 Likelihood

There remain a few things that need to be tested to find the best training model: we discovered that the colour space, data size and covariance had the greatest impact on model selection. The colour space would affect the distribution of the data and, therefore, some colour spaces are much easier to model than others. Here we tested seven different colour spaces introduced in Section 4.3 with an evenly distributed point subsample to test their performance in GMM.

The data format was converted and scaled in OpenCV. We used both diagonal covariance and generic covariance to train the GMM in order to compare and select the better model. The results are shown in Figure 5.7. A Log Likelihood with high value indicates the better data model, and generic covariance performs better than diagonal covariance. However, for all colour spaces, the likelihood variation between \( k \) is huge, which means each increased \( k \) heavily influences the model and causes the model to fit the data better. As a result, the model selection criteria is heavily driven by the Log Likelihood since the penalty (from using too many parameters) is too small in comparison, which causes the models with large \( k \) invariably to outperform those with smaller \( k \) (in the first 30 clusters). For large datasets, the model favours a large size for \( k \). However, we do not require a large \( k \) value to model each plant.

When there is insufficient information contained in a large dataset, it became difficult to estimate the model. This is an issue known as rank deficiency. When training plant data using GMM-EM at a certain number of the clusters \( (k) \), the log likelihood appeared positive. In our case, this manifested itself when there was insufficient information carried by the data to estimate the model (an overlapped dataset). As a consequence, one of the clusters was occupied by a large amount of duplicated data and the covariance was 0 (or very small).

This tends to occur when the data contains too many identical values. In the LemmaTec Arabidopsis data for example, the white background occupied a large proportion of the image, and all pixels shared the same colour value. Such replication does serve to reduce noise but adds no information content. As another example, training black and white images using GMM-EM with \( k = 2 \) would result in two clusters each with a covariance equal to 0 and a positive log likelihood value.
Figure 5.7: Log Likelihood in seven colour spaces trained with different covariance using LemnaTec Arabidopsis dataset.

(a) GMM trained using diagonal covariance.

(b) GMM trained using generic covariance.
5.5.3 Segmentation Results Based on Model Selection Methods

Each generated subsample was tested with ICL criteria for segmentation. Figure 5.8 shows the differences when models are trained using different subsamples; the outer grid is segmented in the same cluster group in the evenly distributed point subsample. The segmentation result derived from using the correlation coefficient method with a threshold of $r > 0.85$ was used to segment the plant cluster (as shown in black pixels), and with this setting, all plant clusters were identified and segmented. The segmentation results were not processed with any noise removal method. We observed that evenly distributed point subsamples suffer less noise than Gaussian Pyramid subsamples.

Overall the model selection criteria worked well with the trained model using the subsample. The overall performance of selecting the number of clusters was close to that of manually selecting $k$. Figure 5.9 shows the criteria value using a model trained with a sample size of 300. All criteria showed the same performance when selecting 11 as the number of clusters and ICL appeared to be more stable than other criteria.

Among all tested criteria, ICL exhibited the best results with most plant datasets along with the previous design decision. The results show good segmentation while keeping the number of cluster low. Although increasing the number of clusters would further improve segmentation accuracy, the improvement would be very small.

5.6 Plant Cluster Selection

To separate plant clusters from other clusters, we used cluster size in combination with time to identify the plant pixels. The correlation coefficient was used here to detect which cluster was the plant cluster. We also examined some parameters in order to define appropriate values for them.
Figure 5.8: Segmentation based on an ICL criterion using different subsamples with cluster mean colour. (a, b) Gaussian pyramid subsample and segmentation result with $k = 4$; (c, d) Evenly distributed points subsample and segmentation result with $k = 4$; (e, f) Every tenth frame subsample and segmentation result with $k = 14$; (g, h) Random subsample and segmentation result with $k = 12$. 

68
Figure 5.9: Model selection criteria on GMM trained with a small number of plant samples using LemnaTec Arabidopsis dataset. The criteria value indicates the performance of the cluster model in each number of clusters: the lower the criteria value, the better the trained model fits the data.
5.6.1 Correlation Coefficient

The experiment took the downsampled (evenly distributed points) Arabidopsis dataset and trained it with a GMM-EM clustering method with a range of $k$ (number of clusters) defined by the ICL criteria. The process produced a list of weights, mean and covariance for each of the clusters. The trained model was then used to predict pixels for the full dataset. The total sum of pixels belonging to the cluster centres from each image was collected to calculate their correlation.

We used Pearson and Spearman's correlation to find the relation between time and each cluster. If the data had a strong positive correlation the results would be close to 1, and a negative correlation would result in a -1, and if there were no relation between the data, the result would be close to 0. If the plant growth were monotonically increasing but not linear, then Spearman's correlation would perform better than Pearson's correlation since Pearson's $r$ is parametric and Spearman's $\rho$ is non-parametric.

Table 5.3 shows an example of Pearson and Spearman's correlation calculated from 7 clusters (the number of pixels belonging to the clusters) against time (where time is the number of frames starting from 1). Cluster 7 with the highest correlation coefficient value is most likely to be the plant cluster (which also has a cluster centre value closest to green). Spearman's correlation appears to have a higher value than Pearson's correlation since the plant data relationship with growth over time is not linear, and this shows that Spearman's correlation performs better as a simulation of plant growth.

5.6.2 Pearson’s $R$ versus Spearman’s $\rho$

Two different correlation functions were applied to the GMM-EM cluster and their results differed since both functions scale differently. Both functions may be affected by noise since the size of the cluster may change (see Figure 5.10). The data shows the same results since they use the same clustering method. Therefore, the total pixel count and cluster centre are shared.

More detailed experiments were performed by manually selecting the plant clusters from a set of trained GMM-EM models with the webcam Arabidopsis dataset. Table 5.4 shows the Pearson and Spearman's correlation for the best manually selected cluster. A graph of manual segmentation results (Figure 5.11)
Figure 5.10: The growth of pixel size in different numbers of clusters over time: (a,b,c) Webcam Arabidopsis, (d,e,f) LemnaTec Arabidopsis, (g,h,i) LemnaTec Oats, (j,k,l) LemnaTec Oilseed Rape.
Table 5.3: Correlation coefficient on an individual cluster with a LemnaTec Arabidopsis dataset.

<table>
<thead>
<tr>
<th>Cluster No.</th>
<th>Cluster Center (RGB)</th>
<th>Pearson’s r</th>
<th>Spearman’s rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>255, 254, 253</td>
<td>0.50</td>
<td>0.56</td>
</tr>
<tr>
<td>2</td>
<td>167, 174, 207</td>
<td>-0.65</td>
<td>-0.51</td>
</tr>
<tr>
<td>3</td>
<td>255, 179, 125</td>
<td>-0.57</td>
<td>-0.51</td>
</tr>
<tr>
<td>4</td>
<td>237, 143, 100</td>
<td>0.55</td>
<td>0.46</td>
</tr>
<tr>
<td>5</td>
<td>62, 64, 63</td>
<td>0.27</td>
<td>0.31</td>
</tr>
<tr>
<td>6</td>
<td>25, 30, 34</td>
<td>-0.85</td>
<td>-0.75</td>
</tr>
<tr>
<td>7</td>
<td>48, 143, 97</td>
<td>0.96</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.4: Segmentation results by manual selected clusters from Arabidopsis webcam dataset.

<table>
<thead>
<tr>
<th>k</th>
<th>Precision</th>
<th>Recall</th>
<th>Jaccard</th>
<th>Dice</th>
<th>Pearson</th>
<th>Spearman</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.246</td>
<td>0.902</td>
<td>0.239</td>
<td>0.386</td>
<td>0.880</td>
<td>0.920</td>
</tr>
<tr>
<td>4</td>
<td>0.461</td>
<td>0.917</td>
<td>0.442</td>
<td>0.613</td>
<td>0.895</td>
<td>0.956</td>
</tr>
<tr>
<td>5</td>
<td>0.539</td>
<td>0.891</td>
<td>0.505</td>
<td>0.671</td>
<td>0.898</td>
<td>0.959</td>
</tr>
<tr>
<td>6</td>
<td>0.602</td>
<td>0.855</td>
<td>0.547</td>
<td>0.707</td>
<td>0.895</td>
<td>0.960</td>
</tr>
<tr>
<td>7</td>
<td>0.822</td>
<td>0.779</td>
<td>0.667</td>
<td>0.800</td>
<td>0.892</td>
<td>0.963</td>
</tr>
<tr>
<td>8</td>
<td>0.800</td>
<td>0.760</td>
<td>0.639</td>
<td>0.780</td>
<td>0.890</td>
<td>0.961</td>
</tr>
<tr>
<td>9</td>
<td>0.940</td>
<td>0.763</td>
<td>0.728</td>
<td>0.842</td>
<td>0.893</td>
<td>0.959</td>
</tr>
</tbody>
</table>

show the evaluation metric values on the LemnaTec dataset between 2 and 29 clusters. The results show good precision for more than 8 clusters, with a precision of 0.999. However, the recall, Jaccard and Dice values show lower precision values.

We chose to use Spearmans correlation to estimate the plant clusters since it is non-parametric and provides better results than Pearsons correlation in the presence of steady, but non-linear growth.

### 5.6.3 Partition

To find the partition between these clusters (as background and foreground), the method generates a list of permutations based on k and combines all clusters in
the list to calculate its new correlation. The combination with the maximum correlation becomes the plant clusters.

The method also assumes that the individual cluster with the highest correlation is part of the plant cluster since only plants grow over time in the entire dataset. As the input dataset always contains temporal information, this assumption always works on plant datasets. We added the additional rule that the (maximised correlation) plant clusters set must contain the assumed cluster. The opposite rule can be true where the individual cluster with the lowest correlation is the non-plant cluster. However, there is a possibility that a negative correlation is caused by a plant changing colour.

Table 5.6 shows all partition combinations with correlation results from the permutation list in descending order (the cluster number in partitions 1 and 2 refers to the Cluster No. from Table 5.3). In the case of four clusters from the data, the cluster set that maximises the correlation is cluster 2, and clusters 0, 1 and 3 minimise the correlation; this partition is considered to be the best foreground/background segmentation solution (when \( k = 4 \)). Figure 5.12 shows one of the segmented images using the solution from the original image (Figure 5.12).

The segmentation result at \( k = 4 \) is not perfect, and can also be improved using a higher size of \( k \). Table 5.6 compares the maximised correlation between a range of \( k \) from 3 to 9. The cluster No. here refers to different clusters in each \( k \).
Table 5.5: Spearmans correlation Partition using four clusters sorted in descending order with the first row being the best partition. Higher Spearmans rho values show a strong correlation of the cluster growth over time and is treated as a plant cluster.

<table>
<thead>
<tr>
<th>Partition 1</th>
<th>Spearman’s rho 1</th>
<th>Partition 2</th>
<th>Spearman’s rho 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.956</td>
<td>0&amp;1&amp;3</td>
<td>-0.956</td>
</tr>
<tr>
<td>2&amp;3</td>
<td>0.935</td>
<td>0&amp;1</td>
<td>-0.935</td>
</tr>
<tr>
<td>1&amp;2</td>
<td>0.888</td>
<td>0&amp;3</td>
<td>-0.888</td>
</tr>
<tr>
<td>1&amp;2&amp;3</td>
<td>0.883</td>
<td>0</td>
<td>-0.883</td>
</tr>
<tr>
<td>0&amp;1&amp;2</td>
<td>0.881</td>
<td>3</td>
<td>-0.881</td>
</tr>
<tr>
<td>1</td>
<td>0.684</td>
<td>0&amp;2&amp;3</td>
<td>-0.684</td>
</tr>
<tr>
<td>1&amp;3</td>
<td>0.390</td>
<td>0&amp;2</td>
<td>-0.390</td>
</tr>
</tbody>
</table>

Figure 5.12: Webcam Arabidopsis original image.
Figure 5.13: Webcam Arabidopsis segmentation at $k = 4$ using the partition and Spearman’s correlation method.

Table 5.6: Spearman’s correlation in different numbers of clusters.

<table>
<thead>
<tr>
<th>Number of Clusters (k)</th>
<th>Number of Clusters classified as plant</th>
<th>Spearman’s rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0.920</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.956</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0.963</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0.980</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.967</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.962</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>0.965</td>
</tr>
</tbody>
</table>
which are not stated, this just to show it is possible that a set of more than one cluster can have the maximum correlation. For this data, a maximum Spearman’s $\rho$ is found when $k = 6$. Figure 5.14 shows one of the segmented images using the maximum correlation. The combination of the plant clusters and background cluster may result in a higher Spearman’s correlation.

It is possible that there is a very small difference between two maximised Spearman’s $\rho$ in different (or the same) $k$ which may result in a different segmentation. We have estimated the mean of the distribution using Students t-distribution and the highest probability is close to 0.00004 which can be neglected.

5.7 The Proposed Segmentation

Following experimentation and comparison between different algorithms and parameters, we developed an automated plant segmentation method that relies temporal information. Table 5.7 shows the segmentation result of four plant datasets. The colour-based method had overall good performance on all dataset. Figure 5.15 shows the visualised segmentation result on four plant datasets. The developed method works well for all types of plants. The segmentation steps are shown below:

1. Generate a subsample based on input time series images, and take evenly
Table 5.7: Colour-based segmentation method results (with standard deviation) on plant datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>precision</th>
<th>Recall</th>
<th>Jaccard</th>
<th>Dice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam Arabidopsis</td>
<td>0.94 (0.11)</td>
<td>0.83 (0.21)</td>
<td>0.79 (0.21)</td>
<td>0.88 (0.18)</td>
</tr>
<tr>
<td>LemnaTec Arabidopsis</td>
<td>1.00 (0.00)</td>
<td>0.91 (0.01)</td>
<td>0.91 (0.01)</td>
<td>0.95 (0.00)</td>
</tr>
<tr>
<td>LemnaTec Oats</td>
<td>0.91 (0.05)</td>
<td>0.68 (0.33)</td>
<td>0.64 (0.28)</td>
<td>0.78 (0.27)</td>
</tr>
<tr>
<td>LemnaTec Oilseed rape</td>
<td>0.98 (0.00)</td>
<td>0.91 (0.08)</td>
<td>0.90 (0.07)</td>
<td>0.94 (0.04)</td>
</tr>
</tbody>
</table>

2. Perform GMM with EM on the subsample, with a number of clusters \( k \) calculated using ICL model ordering criteria.

3. Calculate Spearman's correlation based on the cluster size change across each frame to predict the plant clusters.

4. Segment the plant clusters and generate binary images.

5. Run Morphological Transformation operations on the images: closing to remove noise and opening to group objects (so objects have no edge discontinuities).

6. Combine multiple groups to form the segmentation using only the region from the over-segmented images.

For a morphological transformation, the first image of the dataset is used to find the max value of closing in order that the value not grow so large as to remove any of the plants. Then the last image of the dataset is used to find the max value of erosion to make sure none of the plants touch each other.

5.8 Conclusion

The experiment described in this chapter uses temporal information combined with colour to develop an unsupervised method to segment plants from time series images based on the primary assumption of plant growth. Tests were performed to find the best subsample for cluster training, and the Evenly Distributed Point
Figure 5.15: The segmented plant images using the developed method. (a,b,c) Webcam Arabidopsis, (d,e,f) LemnaTec Arabidopsis, (g,h,i) LemnaTec Oats, (j,k,l) LemnaTec Oilseed Rape. (Left) The original images, (Middle) The clustered images coloured with the cluster mean colour, (Right) The segmented images.
subsample was selected due its good coverage of all samples. Model order selection criteria were compared to estimate the optimal number of clusters for plant datasets. All tested criteria produced similar results, and we selected ICL due to ease of implementation. Furthermore, Pearsons correlation and Spearman's correlation were implemented to predict the plant clusters, and Spearman's correlation was selected as it is non-parametric and is better suited to modelling the plant growth. The method worked well on all plant datasets, especially on the LemnaTec datasets since plants have much clearer growth over time and exhibit less noise. The method was heavily affected by the green tray in the webcam Arabidopsis dataset. This was difficult to avoid since the method is colour based, and a simple solution is to avoid objects of similar colours to the plants in future plant image acquisition.
Chapter 6

Superpixel and Supervoxel Segmentation

6.1 Introduction

This chapter shows an alternative method, which further improves the segmentation results by using a superpixel approach combined with the work from previous chapters. A split and merge method is also developed for superpixels to group different objects in order to segment plant superpixels. We also extend this to a supervoxel-like approach to segment plants with temporal information.

6.2 Superpixel

In computer vision, a superpixel is a group of pixels that share similar attributes. Superpixels can be used to replace regular pixels to reduce the complexity of an image. Based on different superpixel methods, the resulting superpixels are often irregular with varied shapes and sizes. We aim to improve our segmentation method with superpixels, in order to improve boundary accuracy and reduce computational complexity. Therefore, we require a superpixel method with good boundary segmentation accuracy, and capable of producing superpixels with a stable shape and size that can be used for further object tracking.
6.2.1 Superpixel Method Comparison

In this section we select and compare three state-of-the-art superpixel algorithms in detail: Felzenszwalb [85], quick-shift [106] and SLIC (Simple Linear Iterative Clustering) [94]. We use public source code implementations on Arabidopsis images.

Felzenszwalb is a graph-based algorithm using a spanning tree clustering method to generate superpixels. The size of the superpixels generated is based on an input parameter, but the method is unable to control the number of superpixels. Felzenszwalb has a complexity of $O(N\log N)$. Quick-shift is a gradient-based algorithm using a local mode-seeking method to generate superpixels. In common with Felzenszwalb, the method requires parameter tuning to achieve the desired number of superpixels. Quick-shift has a complexity of $O(dN^2)$. SLIC is also a gradient-based algorithm using K-means in a 5D space of pixel colour and location to generate superpixels. This method is designed to output a desired number of superpixels. SLIC has a complexity of $O(N)$ which is faster than most of the superpixel algorithms, especially for images with more than one million pixels.

Due to the differences in parameter setting between each superpixel method, we attempted to manually tune each parameter to achieve the best performance and control the number of superpixels at around the same size. We evaluated each superpixel algorithm with boundary recall and superpixel shapes and sizes.

6.2.2 Superpixel Algorithms Comparison Results

Each superpixel algorithm was tested using our webcam Arabidopsis dataset. We compared each result in both visual comparison and numerical data comparison. Figure 6.1 shows a visual comparison of each superpixel method using one of the webcam Arabidopsis images. All results have a controlled number of superpixels of around 4000 superpixels to prevent differences in accuracy arising from a different number of superpixels. Based on the tested dataset and input parameters, the size of each superpixel is estimated at around 480 pixels.

The size and shapes of superpixel from Felzenszwalb are highly irregular, and the noise from the soil attracted most of the superpixels and for this reason the method missed parts of the plants and the tray. The method requires a much greater number of superpixels in order to improve boundary accuracy. The Felzen-
szwalb algorithm also has a very unstable number of superpixels between each image. For our webcam Arabidopsis dataset, it varies between 800 and 30000.

The quick shift algorithm has good boundary coverage and superpixel size. The shape and location of each superpixel varies between frames. Since the algorithm does not let the user control the number of superpixels, there is a difference in the number of superpixels between each image. For the webcam Arabidopsis dataset, it varies between 4000 and 5000.

In comparison, SLIC provides good superpixel shape and size, allowing easier superpixel tracking between frames. The SLIC algorithm also has a stable number of superpixels across the image dataset.

Table 6.1 shows the performance of a boundary recall measure of each superpixel method on the webcam Arabidopsis dataset. The calculation of boundary
Table 6.1: Superpixel boundary recall measure (with standard deviation) on webcam Arabidopsis dataset.

<table>
<thead>
<tr>
<th></th>
<th>Felzenszwalb</th>
<th>Quick-shift</th>
<th>SLIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary Recall</td>
<td>0.77 (0.16)</td>
<td>0.78 (0.16)</td>
<td>0.89 (0.10)</td>
</tr>
</tbody>
</table>

recall is based on the superpixel segmentation benchmark [83] which computes the fraction of ground truth edges which fall within one pixel ($d = 1$) of at least one superpixel boundary. Superpixel segmentation algorithms with higher boundary recall are considered better algorithms.

The results show SLIC has the highest boundary recall with lower standard deviation than the other two algorithms with a similar number of superpixels on Arabidopsis images. Felzenszwalb shows similar results to quick-shift, but Felzenszwalb produces a more irregular superpixel shape, and the size and the number of superpixels is often larger than those produced by the other algorithms. All algorithms could achieve a better boundary recall by increasing the number of superpixels ( oversegmenting) until a certain limit is reached.

### 6.2.3 Choice of Superpixel Method

A superpixel algorithm is required to produce a stable superpixel shape and size, a controllable number of superpixels, and high boundary recall in order to improve on our developed segmentation method. Based on both visual and numerical comparison of superpixel algorithms, we found SLIC is a better overall superpixel algorithm for plant images, with higher boundary recall. The roughly equal-sized superpixels, combined with a controllable number of superpixels, also supports superpixel tracking between frames. Additionally, the SLIC algorithm requires fewer input parameters and low computational complexity (of $O(N)$). Although SLIC does not generate the same number of superpixels across the dataset, it is very close to the input parameter and is more stable than other superpixel algorithms in maintaining the same number of superpixels.
6.2.4 Combining SLIC with the Developed Method

Based on the results from the previous section, we decided to use the SLIC superpixel algorithm to improve our developed segmentation method. The goal is to replace all pixels in the developed method with superpixels in order to improve boundary accuracy and reduce computational complexity.

We implemented the SLIC superpixel algorithm based on [94] in C++ with OpenCV. The SLIC algorithm divided the image into sections corresponding to the number of superpixels required as input parameter, and set each cluster centre based on the colour and location of pixels. We used all three channels from the Lab colour space and the X, Y location of each pixel from the images, and a K-means was then run for each cluster in a limited region (2D) to define each superpixel.

The base developed segmentation method remained the same, except that it treated the superpixels cluster mean colours and locations as pixels. In this way, when a superpixel was predicted as the plant (foreground), all pixels belonging to that superpixel were treated as plant pixels. This change reduced the complexity of images from millions of pixels to a few thousand superpixels, which lowered computational and memory costs.

6.2.5 Define the Number of Superpixels

The SLIC algorithm only requires the number of desired superpixels as an input parameter, and optionally weight-factors for the data (pixel colour and location). Defining the number of superpixels is important: error segmentation may appear when the number of superpixels is too small, and segments may contain multiple objects and lose boundary accuracy; with too many superpixels, complexity and processing time may increase.

To achieve the best performance of SLIC on plant images, we needed to identify the optimum number of superpixels for its input parameter. Comparison of image resolution and plant sizes from our datasets revealed that the size of plants varied between a few hundred pixels (in early stages) to one hundred thousand pixels (in later stages). The dataset with the biggest plant size difference between early and the late stages is the webcam Arabidopsis dataset, which also has the smallest plant size in pixels.

Three images from three growth stages (weeks 1, 3 and 5) were selected from the
webcam Arabidopsis datasets to test the effect of a different number of superpixels. Figure 6.2 shows the boundary recall and Jaccard of three Arabidopsis images over a number of superpixel settings. The plots show that both boundary recall and Jaccard increase with the number of superpixels. However, the increase is minor when the number of superpixels exceeds 2000, which is most noticeable in the week one image when plants are at their smallest.

Based on the tested image resolution (1600 * 1200), the average superpixel size should be maintained at above 960 pixels. To legislate for the case of a superpixel size larger than the smallest plant size in pixels (which is approximately 500 pixels) in our dataset, the average size of superpixel was set at 500 to prevent segmentation errors. As a result, the input parameter for SLIC was set at: \( \text{Number of Superpixels} = \frac{\text{Image Width} \times \text{Image Height}}{500} \). This yields 3840 which is well over our minimum of 2000.

6.2.6 SLIC Segmentation Performance on Plant Datasets

To obtain a preview of the best possible superpixel-based segmentation results, we performed SLIC on all of the plant datasets, and manually labelled the plant superpixels that contained more than half of the plant pixels. This gave us a baseline “optimal” performance for superpixel-based methods.
Table 6.2: Manually labelled SLIC superpixel segmentation results (with standard deviation) on plant datasets.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Precision</th>
<th>Recall</th>
<th>Jaccard</th>
<th>Boundary Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam Arabidopsis</td>
<td>0.98 (0.01)</td>
<td>0.91 (0.03)</td>
<td>0.89 (0.04)</td>
<td>0.97 (0.04)</td>
</tr>
<tr>
<td>LemnaTec Arabidopsis</td>
<td>0.99 (0.00)</td>
<td>0.93 (0.02)</td>
<td>0.92 (0.02)</td>
<td>0.95 (0.03)</td>
</tr>
<tr>
<td>LemnaTec Oats</td>
<td>0.78 (0.14)</td>
<td>0.86 (0.14)</td>
<td>0.71 (0.19)</td>
<td>0.75 (0.08)</td>
</tr>
<tr>
<td>LemnaTec Oilseed Rape</td>
<td>0.98 (0.00)</td>
<td>0.92 (0.02)</td>
<td>0.90 (0.02)</td>
<td>0.62 (0.07)</td>
</tr>
</tbody>
</table>

Table 6.2 shows the human labelled plant superpixel segmentation result of four plant datasets compared to the ground truth. The results are overall better than the developed colour-based segmentation results, and this demonstrates the potential value of the superpixel-based method. The segmentation performance is less valuable in the case of the side view plant datasets: the boundary shifts to more than one pixel away in comparison to the ground truth, and some superpixels are affected by the frame (background). This particularly affects the Oats dataset.

Figure 6.3 shows the image processed using SLIC and coloured with superpixel means. There is a clear division of means between plant and background, all objects can be recognised, and plant superpixels share close means while maintaining connectivity. Parts of the plant pixels close to the base supporting frame are clustered with frame pixels resulting in a drop in boundary recall. This only appears in images from side-view plant datasets, in which the supporting plant frame/stand is a major visible feature.

6.2.7 Superpixel Based Method Results

We applied the colour-based segmentation method developed in the last chapter with SLIC to create a superpixel-based segmentation method. The method used the means of the superpixels to reduce complexity and segment the superpixels instead of pixels. We tested the method on all plant datasets to check the performance on both top-down and side view plant images.

Table 6.3 shows the superpixel based segmentation results. The superpixel based method shows similar results to the “optimal” results in Table 6.2 and shows a significant improvement in segmentation results on all datasets compared
Figure 6.3: Part of the oats image processed using the SLIC superpixel algorithm. (6.2a) The original image from the LemnaTec oats dataset; (6.2b) The superpixel image with red boundary and coloured with superpixel means.
Table 6.3: Superpixel-based segmentation results (with standard deviation) on plant datasets.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Precision</th>
<th>Recall</th>
<th>Jaccard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam Arabidopsis</td>
<td>0.97 (0.04)</td>
<td>0.74 (0.19)</td>
<td>0.73 (0.19)</td>
</tr>
<tr>
<td>LemnaTec Arabidopsis</td>
<td>0.99 (0.00)</td>
<td>0.92 (0.02)</td>
<td>0.91 (0.02)</td>
</tr>
<tr>
<td>LemnaTec Oats</td>
<td>0.71 (0.17)</td>
<td>0.87 (0.04)</td>
<td>0.63 (0.13)</td>
</tr>
<tr>
<td>LemnaTec Oilseed Rape</td>
<td>0.95 (0.00)</td>
<td>0.93 (0.05)</td>
<td>0.89 (0.05)</td>
</tr>
</tbody>
</table>

to our colour-based method. The superpixel-based method is less vulnerable to noise.

For the webcam Arabidopsis dataset, Figure 6.4 provides a visual comparison between the colour-based and superpixel-based segmentation results. The superpixel-based method is less influenced by the green tray and pots and has better boundary segmentation due to the way in which the method segments each superpixel region as a whole. The segmentation results can be further improved by applying morphological transformations to remove noise.

6.2.8 Superpixel with Plant Tracking

Plant tracking is used to segment each plant and observe their growth between images that contain more than one plant in the dataset. After image segmentation, an object tracking function can be used to track the same objects between frames.

The segmentation method presented in Chapter 5 produced binary images for the segmentation results, which do not track each individual object or determine which pixel belongs to which object. However, the SLIC superpixel algorithm produced a stable number of superpixels of regular size and shape, which made it possible to track most superpixels between frames using superpixels mean colour and location. Although it was not necessary to track every single superpixel, tracking foreground superpixels allowed us to distinguish and track each object between frames in a dataset.

Once all images in a dataset have been segmented, superpixels within one pixel distance of any boundaries are connected as a group in each image. Each superpixel group is treated as a separate plant object, and the mean colour and
Figure 6.4: Segmentation results comparison between the colour-based and superpixel-based methods. (6.4a) The original image from Webcam Arabidopsis dataset; (6.4b) The superpixel image with red boundary and coloured with superpixel means; (6.4c) The colour-based segmentation result; (6.4d) The superpixel-based segmentation result.
central location are used to track the object between frames. A tracking list containing all objects is created using the first frame at the beginning of the tracking process. The likelihood of each object appearing in the last and current frame can be calculated using Euclidean distance. The distance between the object in the current frame and all objects in the next frame are calculated, and two objects with the shortest distance separating them are treated as the same object, and this process is then repeated until all objects between two frames are matched. When there are more objects in the current frame, extra objects are treated as new objects and are tracked in the next frame. When there are fewer objects in the current frame, missing objects are removed from the tracking list.

Figure 6.5 shows the first and last frames from the LemnaTec Arabidopsis dataset. The segmented superpixels are grouped and tracked in each frame and coloured differently. All plants are tracked and matched between frames despite plant movement and rotation.

6.3 Supervoxel

A voxel represents a single point in a three-dimensional space. Analogously to the superpixel, a supervoxel contains multiple voxels that share similar features. Supervoxels are used in 3D-space to segment sets of data points (point clouds) into regions. Alternatively, two-dimensional video or images with temporal information can be treated as a three-dimensional space by stacking frames in chronological order and using time as the third-dimension. Although space might not be evenly distributed, since it is hard to convert the axis that represents time with the pixel position on the image, a weighting factor is applied to manipulate the importance of the time axis.

6.3.1 Supervoxel-Based Method Implementation

Our superpixel-based method can be modified to segment supervoxels instead of superpixels by adding one more piece of data into position as the new axis. Specifically, the SLIC superpixel algorithm used in the method is modified to perform clustering of voxels in 6D-space instead of pixels in the 5D space defined by L, a, b values of the Lab colour space, the x, y pixel coordinates and the time.
Figure 6.5: Plant tracking between two frames from the LemnaTec Arabidopsis dataset. Superpixels are grouped and tracked in different colours. (6.5a and 6.5b) Plant superpixels are segmented and grouped; (6.5c and 6.5d) Plants are tracked and coloured in a different colour.
Instead of using time in seconds, the temporal information is added as a new dimension and is represented as an integer which increments on each consecutive frame. Therefore, the time value is related to the position of the frame in a dataset starting at one, and the time intervals between frames are treated as the same.

The supervoxel-based method segments plants based on temporal information and plant growth. Therefore, the data used for training in the Gaussian mixture model and the predict function remain the same, but the mean colour of the supervoxel is used instead of the pixel to predict the probability of which cluster it belongs to.

6.3.2 Defining the Number of Supervoxels

As with SLIC, the modified supervoxel algorithm requires the number of supervoxels as an input parameter. Similar approaches have been used to test the optimal number of supervoxels, and boundary recall is checked between each frame against the ground truth.

Based on a test of both Arabidopsis datasets, the average size of supervoxel should be maintained above $500 \times \text{depth}$, where $\text{depth}$ is the total number of frames in the dataset or the time value of the last frame. Therefore, the number of supervoxels is calculated in the same way as the number of superpixels: $\text{NumberOfSupervoxels} = \text{ImageWidth} \times \text{ImageHeight} \times \text{ImageDepth}/500 \times \text{ImageDepth}$. Since for top-down view images, objects have small movements, and the plant growth extends in all directions, the result does not change a great deal in comparison to the results of the superpixel algorithm.

However, for the side-view plant dataset, error segmentation appears where objects exhibit large movements between frames. A larger number of supervoxels is required to improve the boundary accuracy. A supervoxel size of 1500 appears to produce good segmentation results when setting the depth of supervoxels to be less than or equal to 3. This means that the max depth of the supervoxel is restricted and the width and height of supervoxel are not limited. As a result, the number of supervoxel is set at: $\text{NumberOfSupervoxels} = \text{ImageWidth} \times \text{ImageHeight} \times \text{ImageDepth}/1500$ for side view images in which plants can move considerably.
Table 6.4: Supervoxel-based segmentation results (with standard deviation) on plant datasets.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Precision</th>
<th>Recall</th>
<th>Jaccard</th>
<th>Boundary Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Webcam Arabidopsis</td>
<td>0.65 (0.28)</td>
<td>0.83 (0.20)</td>
<td>0.61 (0.27)</td>
<td>0.98 (0.03)</td>
</tr>
<tr>
<td>LemnaTec Arabidopsis</td>
<td>0.99 (0.01)</td>
<td>0.92 (0.02)</td>
<td>0.91 (0.02)</td>
<td>0.95 (0.03)</td>
</tr>
<tr>
<td>LemnaTec Oats</td>
<td>0.46 (0.40)</td>
<td>0.72 (0.28)</td>
<td>0.42 (0.36)</td>
<td>0.75 (0.08)</td>
</tr>
<tr>
<td>LemnaTec Oilseed Rape</td>
<td>0.87 (0.05)</td>
<td>0.96 (0.02)</td>
<td>0.84 (0.06)</td>
<td>0.59 (0.07)</td>
</tr>
</tbody>
</table>

6.3.3 Supervoxel Based Method Results

The supervoxel-based method was modified using the superpixel-based method as base model. The supervoxel method uses the means of supervoxels to reduce complexity and segment the entire dataset together instead of one image at a time. The method is tested using all plant datasets to check the performance of time series images from a different camera view.

Table 6.4 shows the supervoxel-based segmentation results. Apart from the boundary recall, the supervoxel-based method shows lower results overall than the superpixel-based method. Although both methods have similar results on the LemnaTec Arabidopsis and Oilseed Rape datasets, the performance of the supervoxel-based method on the Webcam Arabidopsis and LemnaTec Oats datasets is poorer than the colour-based method.

Figure 6.6 shows one of the initial frames of the Webcam Arabidopsis dataset and its segmentation result using the supervoxel-based method. The segmentation result is affected by the edges of plant pots and segments them as the foreground. The plot Figure 6.6c shows the segmentation result (precision, recall and Jaccard) of each frame from the Webcam Arabidopsis dataset: noise appears to impact heavily in the earlier rather than later frames. The supervoxel-based method also failed to segment some small seedlings in early frames.

6.4 Conclusion

This chapter presents the implementation and results of the superpixel-based and supervoxel-based segmentation methods. Both methods used a modified SLIC
Figure 6.6: Images and plot of segmentation results using supervoxel-based algorithms on the Webcam Arabidopsis dataset. (6.6a) One of the original image from the Webcam Arabidopsis dataset; (6.6b) The supervoxel-based segmentation result; (6.6c) Plot of precision, recall and jaccard of all segmentation results from Webcam Arabidopsis dataset.
superpixel algorithm combined with the colour-based segmentation method from Chapter 5 to improve the plant segmentation results with lower computational and memory costs.

For the superpixel-based method, we started by comparing a few state-of-the-art superpixel algorithms and selected SLIC based on the performance of boundary recall and Jaccard on plant datasets. The segmentation results show a significant improvement compared to the colour-based segmentation method. The superpixel-based method is less affected by noise and has a good segmentation precision across frames in all datasets (with low standard deviation). A plant tracking function was also developed to segment each plant separately in order to provide better observation of individual plants over time.

For the supervoxel-based method, we modified the data to add temporal information as the new axis to create a three-dimensional space. The superpixel algorithm was modified to perform clustering in 6D space to segment the entire dataset into regions and segment supervoxels instead of pixels. The performance of the supervoxel-based method is less impressive on time series images with a complex background and large object movements. This shows the potential for a new approach to segment plant images with temporal information. However, the supervoxel-based method requires higher computational and memory costs especially for large datasets with over a thousand time series images. Dividing the large dataset into multiple small datasets may solve the problem. Therefore, the best results have been achieved with the superpixel method, which outperforms all the other methods presented here.
Chapter 7

Conclusion and Further Work

7.1 Introduction

This chapter reviews the work from previous chapters and provides a discussion based on each experiment. Issues and ways to improve the work are also discussed, and suggestions for further work are provided at the end.

7.2 Discussion

The project aimed to develop a plant image acquisition system and plant segmentation methods to segment plants using temporal information whilst making minimal assumptions about the appearance of the plants and the capture setup. The project mainly involved the development of software for plant phenotyping, data acquisition, and the testing and comparison of different algorithms. We discuss each of these in following sections.

7.2.1 The Experimental Setup

The experimental hardware setup involved the development of a low-cost experimental framework to collect time-lapse image from a single viewpoint. The system is low-cost, easy to deploy and is built to work in a controlled environment. It is capable of providing time-lapse images of multiple plants from seedlings to flowers. The image datasets taken by the system are of reasonable quality and resolution.
The webcam Arabidopsis dataset is one of the products of the experimental system and is used as one of the test dataset in this project.

The experimental setup requires a webcam and a PC and, as such, it might not be capable of working outdoors for long periods. One of the issues with the capture setup is the system's use of the JPEG format for image storage, a format that uses compression and incurs data loss. Another issue is the green pots used in the plant images, which increase the difficulty of plant segmentation for our test dataset although this is easy to mitigate against in future capture runs.

7.2.2 Exploring Colour Spaces

The first experiment which examines plant colour in order to find the optimal colour space for segmenting plants. A number of different colour spaces were compared using a supervised segmentation method and evaluated with ROC curves. As a result, the Lab colour space is found to be the best separation between plant and background. However, other colour spaces such as YCbCr also demonstrate good segmentation results and could be considered for use in further experiments.

We also found that plant colour variation is often due to specular reflection, proven phenomenon revealed by using spectrometers and computer vision methods to remove specular reflections.

7.2.3 Colour-Based Segmentation Method

The second experiment involved the development of an unsupervised method to segment plant pixels from time series images. The method relies on the assumption of plant growth and uses temporal information to find plant clusters from a Gaussian mixture model.

Many experiments were done to find the best parameter setting for the developed algorithm. Different subsamples are tested to provide a broad coverage of samples yet a small sample size is retained. Model order selection criteria are compared to estimate the number of clusters before the training process. However, a change of a cluster number does not impact negatively on the performance of plant segmentation as long as the number of the clusters is large enough to represent each object in the scene.
A different correlation coefficient is also used to find the plant cluster based on the plant growth in pixels over time. The Spearmans rho is found to be the best method since it is non-parametric and performs better as a simulation of plant growth.

The developed colour-based segmentation method works well on plant datasets, but it can be influenced by noise and cause segmentation errors especially with a low number of clusters. The method heavily relies on plant growth, and might struggle to rise to the challenges posed by images of stressed plants which may not grow reliably, or plants that never grow in size, or plants which decrease in size due to processes such as senescence.

7.2.4 Superpixel- and Supervoxel-segmentation Methods

The final experiment modified our colour-based segmentation algorithm to superpixel- and supervoxel-based algorithms. The modified methods improve boundary segmentation accuracy at lower computational and memory cost.

Different superpixel algorithms are compared to select the best algorithm for plant images, and a tracking function is made to separate segmented plants to provide segmentation and observation on a single plant. The supervoxel-based method provides better results than the colour-based method but its performance falls short of the standard achieved by the superpixel-based method. The results are lower in terms of boundary recall, and the computational and memory costs are directly related to the size of the dataset.

For training the clusters, use of the mean of superpixel and supervoxel as subsamples would produce bad segmentation results, since there are too many overlapping data samples. Use of an evenly distributed point as training subsample and prediction based on superpixel would still produce better results.

7.3 Conclusion

Through multiple experiments, this project developed a plant segmentation method using temporal information. By analysing plant colour and size, the method is capable of segmenting plants from images unsupervised. The method is capable of providing satisfactory results regardless of different camera viewpoints, plant
Table 7.1: Summary result of our methods on all four plant datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>precision</th>
<th>Recall</th>
<th>Jaccard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour-based</td>
<td>Webcam Arabidopsis</td>
<td>0.94 (0.11)</td>
<td>0.83 (0.21)</td>
<td>0.79 (0.21)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Arabidopsis</td>
<td>1.00 (0.00)</td>
<td>0.91 (0.01)</td>
<td>0.91 (0.01)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Oats</td>
<td>0.91 (0.05)</td>
<td>0.68 (0.33)</td>
<td>0.64 (0.28)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Oilseed rape</td>
<td>0.98 (0.00)</td>
<td>0.91 (0.08)</td>
<td>0.90 (0.07)</td>
</tr>
<tr>
<td>Superpixel-based</td>
<td>Webcam Arabidopsis</td>
<td>0.97 (0.04)</td>
<td>0.74 (0.19)</td>
<td>0.73 (0.19)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Arabidopsis</td>
<td>0.99 (0.00)</td>
<td>0.92 (0.02)</td>
<td>0.91 (0.02)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Oats</td>
<td>0.71 (0.17)</td>
<td>0.87 (0.04)</td>
<td>0.63 (0.13)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Oilseed rape</td>
<td>0.95 (0.00)</td>
<td>0.93 (0.05)</td>
<td>0.89 (0.05)</td>
</tr>
<tr>
<td>Supervoxel-based</td>
<td>Webcam Arabidopsis</td>
<td>0.65 (0.28)</td>
<td>0.83 (0.20)</td>
<td>0.61 (0.27)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Arabidopsis</td>
<td>0.99 (0.01)</td>
<td>0.92 (0.02)</td>
<td>0.91 (0.02)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Oats</td>
<td>0.46 (0.40)</td>
<td>0.72 (0.28)</td>
<td>0.42 (0.36)</td>
</tr>
<tr>
<td></td>
<td>LemnaTec Oilseed rape</td>
<td>0.87 (0.05)</td>
<td>0.96 (0.02)</td>
<td>0.84 (0.06)</td>
</tr>
</tbody>
</table>

species and colours. Table 7.1 shows the results of all three methods on four plant datasets. This plant species-neutral method is a robust way to separate plant from background for many phenotyping applications.

### 7.4 Key Contribution

The key contributions of the project are as follows:

- A supervised comparison between all popular colour spaces to determine an optimal colour space for plant image processing in visible light range, see Chapter 4.

- An unsupervised plant segmentation method based on plant growth to segment plants from time series images without relying on knowledge of the plant colour, species or location, see Chapter 5.

- A method which combines superpixels with the unsupervised segmentation to further improve segmentation results and track a plants over a set of time series images, see Chapter 6.
- A new Arabidopsis time-lapse image dataset from seed to flower, see Chapter 3.
- A number of manually segmented ground truths made from various plant datasets, which can be used for method comparison and evaluation in other plant research, see Chapter 3.

7.5 Limitations

- The developed image acquisition system is not capable of working outdoors. Our method is heavily influenced by the weather and shadows as they affect the colour of images, which brings difficulty for the method to detect growing clusters.

- When recording top-down view images, overgrown plants may block the webcam lens and cause the webcam to lose its focus on the plant. Any data captured after that are not usable and is discarded.

- The plant segmentation methods provide good segmentation results on any time series image dataset, but the type of the dataset and plant the method can segment is limited to images with plants that have visible growth that is meaningfully measurable in pixels, and the method also has difficulty in segmenting overgrown or overlapped plants.

7.6 Future Work

There is a lot of scope to improve the developed software and hardware.

- The hardware setup can be modified to record time-lapse 3D data by adding more cameras at different viewpoints. Different types of cameras such as fluorescence cameras or depth cameras can be used to record non-visible plant information.

- Further analysis and testing with different plant types is possible, especially in the case of plants that evince significant changes in colour over time.
• By observing plant size and colour, the method may also be used to segment mixed plant species from the image, by separate and model each plant with multiple clusters.

• Plant tracking can be modified to segment and separate overlapped plants. By further study the plant growth, we can build models to estimate the overlapped area of plants.

• The segmentation method could also be improved to deal with plants flowering, during which phase colours change considerably.

• Senescence has not been investigated in this project, although extending this work to deal with senescence should be possible. The life cycle of a plant which senescence could be seen as growing then turning brown. In this context that would involve the “green” cluster growing until the plant becomes full sized, then a brown cluster growing whilst the green clusters shrink. Detecting and modelling this should be possible in this framework.
Bibliography


[16] Christine Granier, Luis Aguirrezabal, Karine Chenu, Sarah J. Cookson, Myriam Dauzat, Philippe Hamard, Jean-Jacques Thioux, Gaëlle Rolland, Sandrine Bouchier-Combaud, Anne Lebaury, Bertrand Muller, Thierry Simonneau, and François Tardieu. PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in Arabidopsis


