Alma Mater Studiorum - Università di Bologna
Campus Di Cesena

SCUOLA DI INGEGNERIA ED ARCHITETTURA

Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

IMAGE ANALYSIS METHODS FOR SUGAR BEET PHENOTYPING

Tesi in:

Tecniche Avanzate per l'Analisi delle Immagini e Visione

Relatore:
Prof.
ALESSANDRO
BEVILACQUA

Presentata da:
MARCO PARI

Correlatore:
Ing.
ALESSANDRO
GHERARDI

Sessione III
Anno Accademico 2014-2015

To my family, for the constant support,
and to Marianna, *for being exactly like she is ...*

Contents

List of Figures

List of Tables

Abstract

In un mondo che richiede sempre maggiormente un'automazione delle attivitá della catena produttiva industriale, la computer vision, rappresenta uno strumento fondamentale per ciò che viene giá riconosciuta internazionalmente come la Quarta Rivoluzione Industriale o Industry 4.0. Avvalendomi di questo strumento ho intrapreso presso l'azienda Syngenta lo studio della problematica della conta automatica del numero di foglie di una pianta. Il problema è stato affrontato utilizzando due differenti approcci, ispirandosi alla letteratura. All'interno dell'elaborato é presente anche la descrizione progettuale di un ulteriore metodo, ad oggi non presente in letteratura. Le metodologie saranno spiegate in dettaglio ed i risultati ottenuti saranno confrontati utilizzando i primi due approcci. Nel capitolo finale si trarranno le conclusioni sulle basi dei risultati ottenuti e dall'analisi degli stessi.

Chapter 1

Introduction

Computer Vision approaches allow to extract, in an automatic way, information that could only be obtained, until few years ago, using visual analysis. Manual operations introduce errors and, sometimes, require long elaboration times. These operations can be improved through using an image of a subject of interest, to automatically get important information that could be used in the next productive steps. Having this goal I started my experience in Syngenta, in collaboration with an Agronomy Ph.D. student, Joalland Samuel. Samuel has been working on the development of methods to non-destructively evaluate differences in the growth of the canopy of sugar beet plants. More precisely, he focused his research on the effect of nematodes on the plant growth. Nematodes are soil-borne pathogens occurring naturally in the soil. Nematode infestation on the roots of sugar beet causes delay in the leaf development (leaf apparition) and canopy growth. Since counting the number of leaves and calculate the canopy area of single plant is extremely time-consuming , it appeared crucial to develop a fast, cheap and reliable tool to perform those tasks. The objective of my Master Thesis project was to study and develop a computer vision based tool to automatically detect and count leaves of sugar beet plant, starting from an image. More specifically, the main objectives were:

- investigating the state of the art in the field of plant phenotyping

- applying the most promising techniques. 3 main methods were identified:
  - a leaf detection approach recently developed by [1];
  - several image based transforms to extract plant properties;
  - implementing the method based on "midrib" detection.

The first method was identified after a thorough research of the state of the art regarding leaf detection. I immediately found the Leaf Segmentation Challenge [2], a competition (carried out during ECCV - the European Conference on Computer Vision) where all the participants have to show the performance of their automated approach to detect leaves on a given dataset. Among all the participants, authors of [1] showed the most interesting approach because it could effectively detect leaves with a certain level of accuracy.
The second approach implemented an holistic approach, starting from my theoretical background to find the best solution to my problem. Besides trying finding the best method to enhance certain characteristic of the leaves I also tried to elaborate a logic operator chain to archive a certain mid-goal and be a step closer to the main objective.
The third approach was the first one that I believed could be really applicable, since the midrib is much easier to find in a complex environment compared to the shape of the whole leaf. Nevertheless the literature did not reporting convincing studies about the use of midrib to detect leaves and estimate the leaf area.

The Thesis work is divided in the following sections. In Chapter 2, I will analyse the context of the Company and of the environment where the study was made and the problematic found. In Chapter 3, the problems analysis followed by the domain independent state of the art and consequent open problems will be presented, combined with the design of the solutions. In Chapter 4 the implementation of the approach will be described. In chapter

5, I will analyse the results obtained using the implemented solutions and the main differences comparing them. Finally, in Chapter 6, I will draw general conclusion based on the results analysis. Possible future developments will be also discussed.

Chapter 2

Context And Background Analysis

In this chapter I will give a brief explanation about the company where I spent my internship. I will also give more details about the agronomical background and the specific problem I had to deal with, the current domain-specific state of the art and the related open problems.

2.1 Initial Situation

Syngenta AG is a world-wide agriculture company, with premises in Switzerland that produces agrochemicals and seeds, also carrying out genomic researches. It was formed in 2000 by the merger of Novartis Agribusiness and Zeneca Agrochemicals. As of 2014 Syngenta was the world's largest crop chemical producer, the strongest in Europe. As of 2009 it ranked third in seeds and biotechnology sales. Sales in 2014 were approximately USD 15.1 billion, over half of which were in emerging markets.

Figure 2.1: Syngenta Logo

Syngenta has eight primary product lines which it develops, markets and sells worldwide; its five product lines for pesticides are selective herbicides, non-selective herbicides, fungicides, insecticides and seed care. Three product lines for seed products include corn and soya, other field crops and vegetables. In 2014, sales from crop protection products accounted for USD 11.381 billion, i.e. 75% of total sales. Field crop seeds include both hybrid seeds and genetically engineered seeds, some of which enter the food chain and become part of genetically modified food. According to Syngenta, in the US their "proprietary triple stack corn seeds expanded to represent around 25 percent of units sold." In 2010, the US EPA approved insecticidal trait stacks including Syngenta's AGRISURE VIPTERA gene, which offers resistance to certain corn pests. Syngenta cross-licenses its proprietary genes with Dow AgroSciences and thus is able to include Dow's Herculex I and Herculex RW insect resistance traits in its seeds. It sells a VMAX soybean that is resistant to glyphosate herbicide. Key Syngenta brands include Actara (Thiamethoxam), Agrisure (corn with Viptera trait), Alto (Cyproconazole), Amistar (azoxystrobin), Avicta, Axial, Bicep II, Bravo, Callisto, Celest, Cruiser (TMX, Thiamethoxam), Dividend, Dual, Durivo, Elatus, Fusilade, Force, Golden Harvest, Gramoxone, Hilleshoeg, Karate, Northrup-King (NK), Proclaim, Revus, Ridomil, Rogers, Score, Seguris, S&G, Tilt, Topik, Touchdown, Vertimec and Vibrance. Syngenta is listed on both the Swiss stock exchange and in New York. Syngenta employs over 28,000 people

in over 90 countries [3].

2.2 Agronomical Background

The Research Unit (RU) where I worked is currently focusing on understanding the effect of Beet Cyst Nematodes on sugar beets plants and on the early detection of symptoms in order to timely adopt countermeasures [5]. To this purpose, the RU is working at the development of a specific platform called "pot-in-pot system" which is fully dedicated to the study of seedcare solutions against the sugar beet cysts nematode. Sugar beet is one of the most important crops in Europe. Its yield is formed by the beet, a storage root that is initiated from an early developmental stage on. Often the development of the beet is seriously affected by soil borne pathogens such as plant-parasitic nematodes. Worldwide, these parasites cause annually up to 20 %, or approximately 100 billion USD, of financial losses on crops such as soybean, cotton, cereals, tuber crops, legumes, fruit and vegetables [4]. Seinhorst[6] modelled and demonstrated the strong link between numbers and types of nematodes and crop performance. The sugar beet cyst nematode (BCN) Heterodera schachtii occurs in patches in the field and has a low mobility. Briefly, when stimulated by optimal soil moisture (80 to 100 % of the field capacity) and temperatures (20 to 23 ℃), second stage juveniles hatch from the cysts. After migrating through the soil to the host-plant roots, juveniles enter the roots and start establishing a feeding site which damages the vascular tissue. Hatching, penetration and infection of the roots can occur within a few days [7]. Belowground symptoms include a reduction of the beet growth and the appearance of many secondary roots to compensate those infested by nematodes. BCN also cause diverse aboveground symptoms like stunted growth, decreased chlorophyll content and wilting of the canopy due to water stress [7] [8]. These visible symptoms in the foliage make BCN an appropriate target for non-destructive phenotyping method development.

2.3 Non-destructive Nematode Stress Detection Principle

One of the main problematic inside Syngenta is to early detect stress in plants. Considering that stress can derive from several sources, the company uses different RU to study them. Various methods to detect stress caused by nematodes have already been tested successfully on many crops. Most of them are based on images and non-imaging multi and hyper spectral measurements with the calculation of spectral vegetation indices (SVI) [9], [10], [11]. [12] demonstrated the potential of normalized difference vegetation index (NDVI) to evaluate the symptoms caused by BCN on sugar beet plants under controlled conditions. Correlations of SVIs with other physiological or agronomical parameters (yield, nutrient supply) have also been reported for greenhouse and field experiments [13], [14] and [15]. However, spectrometry requires the use of complex and expensive devices and data analysis is generally time consuming. After pre-processing the data using reference measurements and percentage of soil coverage, more than sixty SVIs per spectrum can be calculated. In contrast, visible imaging technology uses low cost sensors which are easy to handle and calibrate [16]. Such a technological approach requires a standard digital camera and an image analysis software that allows for the calculation of suitable parameters of a single plant. It is important to point out that all those experiments has been carried out under green house conditions, a completely controllable environment. In contrast, our problematic is situated on a semi-field environment, a mid-way between a green house and an open field. In this situation we have a few plants in a shared pot and slightly changeable lighting conditions. So the study has to be done on a pot composed of, in our case, three plants and not on a single plant or an open field. Moreover, one of the requirement of the project is to build a non-destructive final solution, so that it can be applied during the entire growth of the same plant. The most promising way to archive that in Syngenta was obtained studying the canopy of the plant.

2.4 Application Domain State Of The Art

The complexity of the plant phenotyping field forces researcher to completely change their approach depending on the condition which they are working with. If the research is done under lab condition, they can use some constrains or "truths" which are not granted under field conditions. The best results obtained in a controlled environment had been achieved by [20]. The authors archive a precise result detecting, for each plant, centre, middle point, and five coordinates for each leaf: tip, center, base, right and left edge. This goal is reached using a light-field camera that simultaneously provides a focus image and a depth image, which contains distance information about the object surface. Although results obtained are precise, this approach is not applicable under semi-field conditions, because of the varying light conditions. Furthermore, this solution does not deal with overlapping leaves between plants but only with overlapping leaves of the same plant. Another intresting work is [21]. This is a Machine Learning-based approach. It could lead to precise and time-improving results, although a good ground truth on which building the knowledge base is needed. As the previous one, this approach is influenced by lab condition and could yield misleading results if used under real-world enviromental conditions. One last example is the work presented in [1], submitted on the stage of a very important leaf segmentation contest. Sadly this works on single plant too, but the work flow applied is very interesting and, thanks to the architectural similarity between rosette plants and sugar beets, could be used with the necessary precautions. Even if all those solutions contains techniques that could lead to good partial results, they can not be taken as a whole solution to our problem. Due to a lack of tools and different environmental conditions they can not be adapted. Nevertheless some similarity can be found. In fact, all the studies are based on rosette plants. This particular kind of plant has an architecture (leaf shape, plant shape and leaf organization) very similar to sugar beets. A comprehensive overview about phenotyping software can be found at [48].

2.5 Domain Specific Problems

Canopy area is a global and general parameter which has several disadvantages. When two leaves are overlapping, the part of the area covered by the upper leaf is not calculated. Another further neglected detail is that leaves are not completely flat but have a 3-D shape. This can lead to a miscalculation of the total area. In order to avoid those errors and go deeper in the analysis of the sugar beet canopy, it appeared necessary to use a more precise parameter related to canopy area. Milford [17] was able to describe the canopy as the sum of 3 main parameter:

Organogenesis: measured through the leaf number;

Morphogenesis: related to leaf surface;

Leaf Organization: leaf orientation compared to the centre of the plant.

From these three parameters,the question is to identify the one that is the most affected by the nematode stress. In the literature, there is no reference regarding the potential impact of nematode stress on the leaf organization. The two other factors are most likely affected by nematode stress. In the end, the main idea is to build up a solution able to precisely measure the leaf number and the leaf area. Considering that, to measure a leaf area, we firstly have to correctly detect the leaf, and to measure the leaf number we have to detect each leaf of a single plant, the focus of the Thesis will be the leaves detection.

Chapter 3

Design of the Solution

In this chapter, I describe firstly the problem analysis, then the state of the art regarding the object detection under similar conditions I met. After that, detailed description of the state-of-the-art algorithm [1] that inspired this Thesis work, the open problems and can be found in this chapter as well. Furthermore, a description of a new Midrib-focused approach can be found. That was the approach regarding leaf counting considered during the first Thesis period. However, this approach could not be further developed during my internship because some specific equipment were required. In the end a detailed description of the algorithms developed can be found.

3.1 Problem Analysis

The only instrumentation available to create a computer vision solution were a single compact camera. Neither an automated way of acquiring images and fixed acquiring location were present. To have an effective representation of the canopy of a plant on an image I proceeded to capture a single image from above the pot. Another way could have been capturing a couple of images and apply some stereoscopic vision technique, in a way to add some depth information. Sadly, due to the short time of the project, this path was too long to be applicable. In this kind of domain, the Thesis approach will

try to extract information about the canopy using a picture taken from the top of each pot. There are three plants per pot (per picture). Half of the pots were infested by nematodes. Pictures were taken at different growth stages (plant overlapping or not). The overall aim of this Thesis work is to examine the potential of a single image based phenotyping method to discriminate nematode infested and non-infested sugar beet plants and to predict the beet biomass - as a key component of the final yield - in the early growing stages under semi-field conditions. More specifically, the main objectives are to:

- test the ability of top-view digital images to estimate the leaf biomass;
- examine the possible correlations between leaf and beet biomass;
- evaluate the ability of top-view digital images to discriminate between nematode infested and non-infested plants and to quantify in a non-destructive manner the damage caused by nematodes.

In conclusion, the main challenge is to detect presence of BCN in the plants in a non-destructive way using only a picture captured from above of the canopy and extracting information for the canopy analysis. This approach was chosen because of his non-invasive behaviour, ideal to apply during the whole life cycle of the plants. A manual count of the leaves using existing pictures from the top allowed us to discover that nematodes delay the apparition of leaves (see figure 1.3).

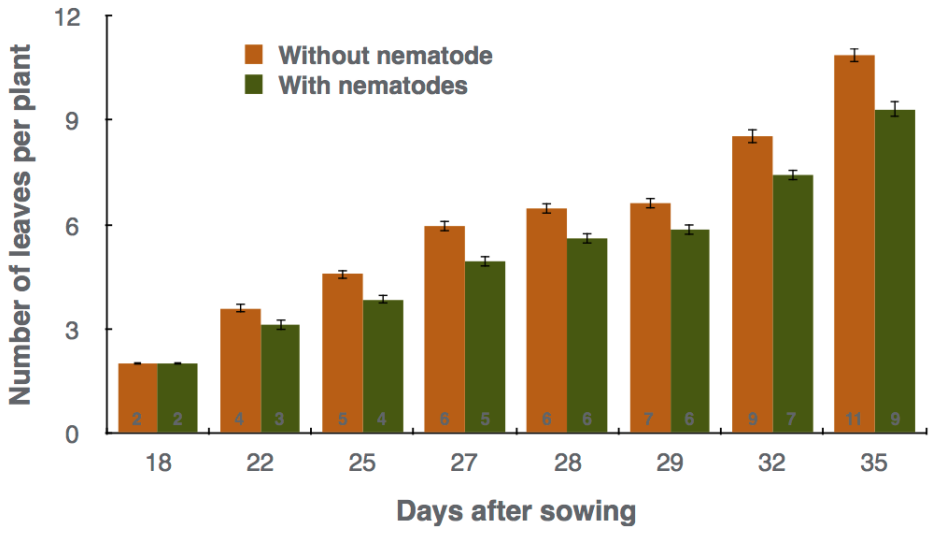


Figure 3.1: Leaves number comparison between infested and non-infested plants

Leaf number seemed to be a great parameter to discriminate nematode infested and non-infested plant. Building a reliable tool that will automatically count the number of leaf could help to early identify stress caused by nematodes.

3.2 State Of The Art

In Computer Vision, detecting and segmenting objects is in general a challenging task under different conditions, depending on the object itself, the environmental conditions, the image acquisition equipment, and so on. Some of the principal approaches that could be considered depends on the context (or object class) where we are going to operate and generally we have three main scenarios:

- the general case: where we are not very specific on the class of the object to detect. In this case distortions may not be crucial. The main

approach here is to be able to describe every possible object in the scene using some general purpose descriptor. This reasoning is based on the concept that for a similar objects we have similar descriptor. In this case there are various descriptor we can count on: SIFT[27], SURF[28], HOG[29] etc.

- If the problem is more practical, template matching may be the best approach. Of course, this assumes there is only very slight distortion. Template matching is used in combination with machine learning techniques, so that our program can learn while is running, in order to become every time "smarter".

- Finally, if the problem is more shape-based (handwritten characters, etc.), shape contexts are a great solution. It requires low clutter, but scale, rotation and lighting are rarely a significant issue.

In more recent years, the field of computer vision has shifted in focus toward the object detection problem, in which the input image is searched for a specific target object. One reason for this lies in the development of machine learning algorithms that leverage large amounts of training data to produce robust classification results. This led to rapid progress in the development of object detection systems, enabling them to handle increasing levels of background noise, occlusion and variability in input images [30]. This development established the standard practice of working with input domains of real images of cluttered scenes, significantly increasing the applicability of object recognition systems to real problems. Another way to engage the problem is utilize some description methodology. This trick allows to use a "sliding window" over the total picture and, calculating the descriptor of the current window, see if it matches the one of the object we are looking for (with a certain level of discrimination). This is also the fist step implemented by [31], followed by a tracking algorithm.

3.3 Open Problems

Scene understanding is still a huge open problem in computer vision. To accomplish this task object detection is the first step. But even to archive that no unique solution is implementable and still an environmental analysis is required to identify the best approach. In some environments we can alterate the environmental conditions to have, as an example, uniform illumination or better contrast. In other situations, the environmental conditions are not modifiable, like in [32]. In this case an external illumination system would scare the fishes. Because of the semi-field conditions does not allow to change the environmental situation, the most probable segmentation approach will be the shape-based one. Indeed, thanks to the fact that the image is captured from above, we can use the canopy area shape to determinate the number of leaves on a single plant. This way we are even able to utilize the same approach even to other kinds of plants. The only constraint is that the shape must be similar to the one the algorithm has been developed for. Although there is a second problematic inside the detecting problem: how to deal with overlapping object. This kind of problem is relevant in our case since that, while the plants grow, leaves can overlap among each others.

There have been many approaches to separate overlapping objects. These include the watershed algorithm [34] - [35], the gradient or edge detection method [36], morphological erosion [37], the active contour method [38] - [39], the sliding band filter approach [40] - [41], and others. A nice review and comments on some of these approaches can be found in [37]. The gradient or edge method apparently does not work well in cases where there is no obvious intensity difference between the overlapping objects or if the objects are strongly textured. The active contour method is quite computationally demanding, making it unsuitable for a case in which the number of objects is large, which is in fact the most meaningful case for computer-aided segmentation. The sliding band filter approach requires that the range of object size be known beforehand, and it does not work well if the size range is wide. While the watershed method is still an effective and efficient method to sep-

arate overlapping objects, improvements can be made to the algorithm. The advantages of the watershed approach are

- it can provide the natural growth of the region corresponding to each object independent of object shape and size;
- it automatically provides a closed contour as well as computational efficiency.

However, directly applying the watershed algorithm to the image or its gradient can lead to severe over-segmentation due to large numbers of local minima/maxima in the image or its gradient version. Many remedies have been proposed to overcome this issue [42], [43], [44], [45]. Hierarchical watershed segmentation aims to merge the over-segmentation hierarchically to form meaningful object regions, for example based on the mosaic image transform and associated graph [44] or by multi-scale filtering of the image and segmenting on the filtered and simplified image [47]. Some other studies have proposed using the pattern classification and object model learned from the data to direct the region-merging [42], [43], [46]. Compared with the methods aiming to conduct a blind watershed on the image first and then merge the over-segmentations afterward, it would be better not to over-segment the image in the first place. I believe the best way to conduct watershed segmentation is not directly from the original image or its gradient version. It is better to first find the marker corresponding to each object in the image and then to conduct the watershed based on those markers. This approach gives a much better guarantee of object counts and approximate locations in the image. Therefore automatic detection of markers is the most critical step in using watershed segmentation. There are several approaches to detecting markers, including the distance transform, morphological erosion, and the gradient transform. Under the appropriate condition, such as for convex object, the first two approaches can be shown to be essentially similar, as the final result of morphological erosion is also the local maximal distance region [37]. Both approaches are based on purely geometrical information

and require the overlapping objects to display a bottleneck region as the hint for the location of separation. Both also require the individual object to be more or less convex in shape, and may lead to over-segmentation when this requirement is violated. An alternative way to detect markers is the gradient transform. It is based on the assumption that the inter-object gradient is larger than the intra-object gradient, and connected low gradient regions are detected as markers. However, this method is very sensitive to image noise and often leads to over-segmentation. Therefore using the distance transform as the basic framework and combining gradient information into the system would be a good option. To combine gradient information into the watershed process based on the distance transform framework, one study uses the gradient-weighted distance transform [42] to alter the "distance" at a certain pixel regarding its gradient. Such a method is free of parameter tuning, but the incorporation of the gradient into the geometric framework is based on heuristics. So it is not immediately apparent where the watershed or boundary will be, or whether it will be accurate.

3.4 Midrib Method

The midrib can be described as the central vein of a leaf (see figure). It runs from the leaf base to the extremity of the leaf (apex).

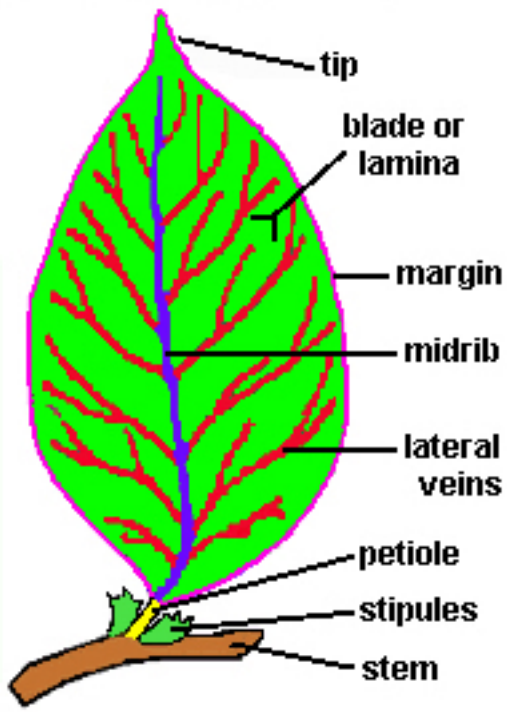


Figure 3.2: Leaf Points of Interest.

Detecting midrib was the first approach hypothesized as a good approach to count leaves. It is based on the assumption that we can detect the midrib of each leaf. Using that parameter as a feature, a well-built algorithm should be able to easily count leaf and, knowing the length of the midrib, estimate the total Leaf Area. A description on how a midrib-based approach should be organized is described below.

Acquisition In this part the acquisition methodology to achieve the best result will be explained. It is important to point out that the midrib of a leaf could be a really small feature to detect. A way to detect features is achieved by comparing each pixel with his neighbour, looking if the specific feature is still present or not. Following this reasoning, it is clear that the higher the picture resolution the easier the detection. Otherwise we can have a "blur"

effect, and a reasoning is needed on where consider a feature "not present" and where "present". According to the manual calculation on the semi-field, a Midrib can even have a thickness of 0.05 mm ± 0,01 mm. A tolerance of 10% of the features size is needed to properly detect it. Also, the feature must be detected with 3 pixels (in the middle the feature is present and in the neighbour is not). So it can be tricky to have a sensor that can detect, with 1 pixel, 0.0003 mm of the object of interest. Indeed, 1 mm^2 is needed to have at least 3000x3000 pixels, which means 9 Mpx to cover one mm^2. Moreover it is worth considering that the whole pot has an area equal to the one of a circle with 52 cm diameter. Adding another 10% of tolerance, it became 57,2 cm. It is easy to calculate:

$$\frac{0{,}0003[mm]}{1[pixel]} = \frac{572[mm]}{x[pixel]}$$
$$x = 1.906.666,6[pixel] = 1,907[Gpx]$$

Considering that such a high resolution is not achievable with a single camera, we decided to build up a single high resolution picture stitching together several lower resolution pictures, using a technique called Mosaicing [22]. Although this goal could be achieved with a standard compact camera, this kind of tool would add a not negligible percentage of human error. To avoid it the right tool to use would be an industrial camera specially designed for Computer Vision applications. With this instrument, it is possible to take several pictures at the highest frame-rate allowed by the camera, having as output a collection of images instead of a single video file. With this capability an operator has only to "cover" the whole surface while snipping picture. Of course a camera calibration algorithm has to be executed in order to have extrinsic and intrinsic and perform an optimal stitching and, by consequence, mosaicing.

Pre-Processing Once the right picture is ready, a filtering of the image is necessary to highlight those details that then will become the features to identify.

Segmentation A simple colour thresholding using L*a*b colour space can be used for midrib identification. This is the hardest part. The plant properties (more developed leaves had lighter Midrib compared to the rest of the blade) and the semi-field light conditions caused a not easy to parametrisation to properly detect midribs by colour. If this step is properly implemented, the leaf number (equal to the number of the midribs) should be accurately calculated.

Feature Extraction Once midribs are properly detected, an evaluation of the length of each midrib is necessary. It is worth considering that the leaf is not a flat object but is slightly concave/curved. Once in possession of the length of each Midrib, due to the linear correlation early proven, it is immediate to calculate single leaf areas.

Post-Processing To properly calculate each canopy area parameter, it should be feasible to draw straight segments on the input picture according to the centre and the length of each midrib. That should be enough to calculate each leaf orientation according to the centre of the plant.

3.5 LSC Paper Method

This algorithm has been developed and tuned for the Leaf Segmentation Challenge (LSC) organized for a Plant Phenotyping (CVPPP 2014) workshop that took place in conjunction with the 13th European Conference on Computer Vision (ECCV). It is a 3D histogram-based segmentation and recognition approach for top view images of rosette plants such as Arabidopsis thaliana and tobacco (the dataset was provided by the event organizer).

Acquisition This methodology requires plant images and manually labeled images as input for the training phase. Images were provided as dataset within the LSC.

Pre-processing All RGB images are converted into the L*a*b* colour space. Afterwards a simple colour threshold is applied to keep only the pixels related to leaves. Therefore a 3-D histogram creation is performed for all training images (with labels) from given dataset. Each pixel from the training image is categorized into foreground pixel or background pixel by inspecting the provided label data. The corresponding L*a*b* pixel color values are used as indices for the 3-D histogram cubes. For each pixel the corresponding histogram bin is incremented. During this procedure, an overall foreground and background 3-D histogram is accumulated. To improve the robustness of the threshold approach, all input images for the cube calculation were filtered in the pre-processing phase by a Gaussian blur operation. A direct look-up in the 3-D histogram cubes is used for segmentation purpose.

Feature Extraction Results of the segmentation serve as input for the leaf detection step. In this phase the Euclidean distance map (Edm) method is processed by a maximum search. A skeleton image is calculated for the subsequent analysis steps. To detect split points for leaf-separation, a graph structure for efficient traversal of the plant mask image skeleton is generated. Before generating the graph, values of the calculated distance map are mapped on the skeleton image. The result image is used for creation of the skeleton graph: Leaf center points, skeleton end-points and skeleton branch-points are represented as nodes in the graph. Edges are created if the according image points are connected by the skeleton. Additionally, a list of the positions and minimal distances of each particular edge segment is saved as an edge-attribute. This list is used to detect the exact positions of the leaf split points. To separate all leaves from each other, all paths between the leaves are investigated using the corresponding graph structure. The minimum distance points (points where the distance to the image background is minimal) between any two leaf center point nodes are determined by investigating the path edges minimum distance attributes and saved as leaf split points. The according edges are removed from the graph structure.

This procedure continues until all leaf centre point nodes in the graph are disconnected from each other. For each split point the nearest background point is searched. The second coordinate of the split line is searched at the opposite position relative to the split point. After the split line estimation, a region filling, considering the segmentation result and the split line positions, is performed starting from the leaf centre points. The result represents the leaf labels.

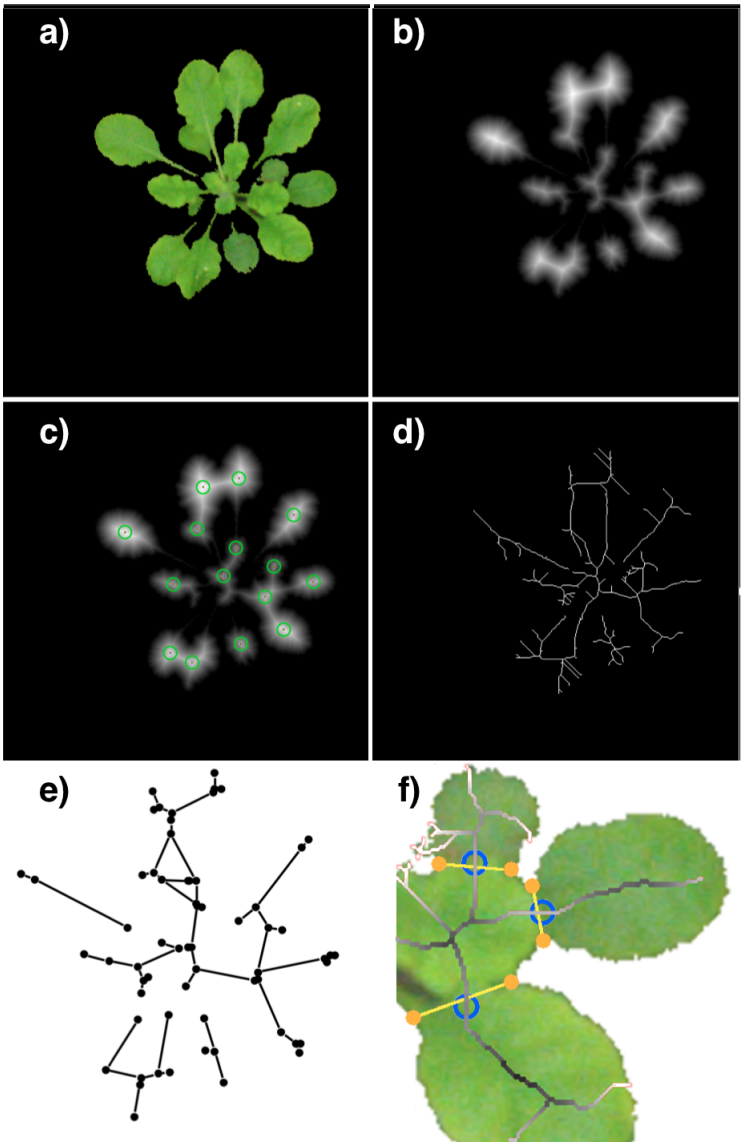


Figure 3.3: Mid stages of paper method approach.

In Figure 4.2 can be seen: a)Segmentation, b)Euclidean distance transform, c)Edt center point, d)Skeleton, e)Resulting graph, f)Split points and splitting lines.

Results on challenge context and problems in our case This method, applied in the context of the LSC, gives good results concerning the leaf detection. Nevertheless, several semi-field characteristics makes more difficult the application of the method on the pot-in-pot platform:

- 3 different plants are present in the same pot;
- growth of the plants also increase the complexity of the system;
- light condition cannot be controlled.

It is easy to detect the defects of this method. Indeed, the split points detection would not be easy to achieve due to the complexity of the skeleton and of the resulting graph.

3.6 The Final Method

In this section I will explain the logic chain of reasoning that leads to the final implementation of the Thesis project. This approach only needs as input a single image taken from the top of the canopy. Due to his versatility it can be used for several kinds of plants. They only need to share a similar plant architecture.

Acquisition This is a main point where every computer vision project should start. The acquisition is not only the very first action every computer vision approach starts with, but can also have a great influence on the work that as to be done after that. As said, Samuel used a common compact camera. For the aim of this project is suggested to use a tool more oriented on computer vision use. With the approval of the company, an industrial camera as been purchased. This instrument allows to control every parameter over the acquisition of the pictures. Because of a lack of resources and time I was not able to use the tool aforementioned. So the acquisition was improved as Syngenta internal step but not used as starting point for this project. Instead I used pictures already captured by Samuel.

Pre-Processing When I received the image database I immediately noticed that picture were acquired during different time of the day. This is translatable as different light condition. Also the picture did not contain only the main subject but also plants close by. To resolve this kind of problems I thought of adopting the following technique. Starting from the input image, I apply a simple ROI resize. This allows me to remove al the neighbour plants. Since the pot size is the same regardless of the growth stage or the infested/non-infected state of the plants, this resize can be done using fixed parameters. After that, I convert it in the L*a*b* color space. This operation is fundamental because of the different light condition of the various picture taken. If the color threshold were done on RGB color space, the results would greatly vary according to the aforementioned conditions and could be not parametrized. This is manageable under L*a*b* color space because it use a separate channel (the L channel) to handle the light. Immediately after the conversion we build up an output image composed by only the green parts of the input image. After the thresholding the image is converted back to RGB color space. Before we finish the pre-processing, a Gaussian blur is applied to the thresholded image to increase the focus on the main subject (the canopy).

Segmentation With blurred version of the thresholded image, the algorithm proceeds to calculate the Euclidean Distance Transform of the picture. This method gives as output a gray scale image where the more distant the pixel from the black background, the more lightly they are represented. Thresholding this output allows to take only the "peaks", helping to better differentiate leaf. At this point, for every single leaf, the output image will have a single white blob. It is worth considering that, at this point, overlapping leaves are detected with a single blob.

Ultimate Erosion And Feature Extraction At this point a method to discriminate between single leaf and overlapping leaves is needed. The Ultimate Erosion method has exactly this purpose. It applies several times

an Erosion morphological operation, and every time it keeps track of the centre of each white blob detected. If, during the erosion, a blobs gets divided in two blobs, that means that it was composed by two overlapping leaves. So it deletes the previous saved centre point (the one relative to the bigger blob) and saves the two new ones. To avoid errors due to the shape of the leaves, if one blob gets divided in two but one of those two has a smaller area than a certain threshold, the too small blobs are on purpose not detected. At the end, the output will be a collection of centre points, each one detecting the centre of each leaf.

Post-Processing As final step, each detected centre point is drawn on the input image to give to the user a visual effect of the result obtained and to immediately test the goodness of the algorithm. In Figure 4.3 can be seen: a)Thresholding, b)Black and White conversion, c)Euclidean Distance Transform, d)Peaks, e)Ultimate Erosion result, f)Final Output.

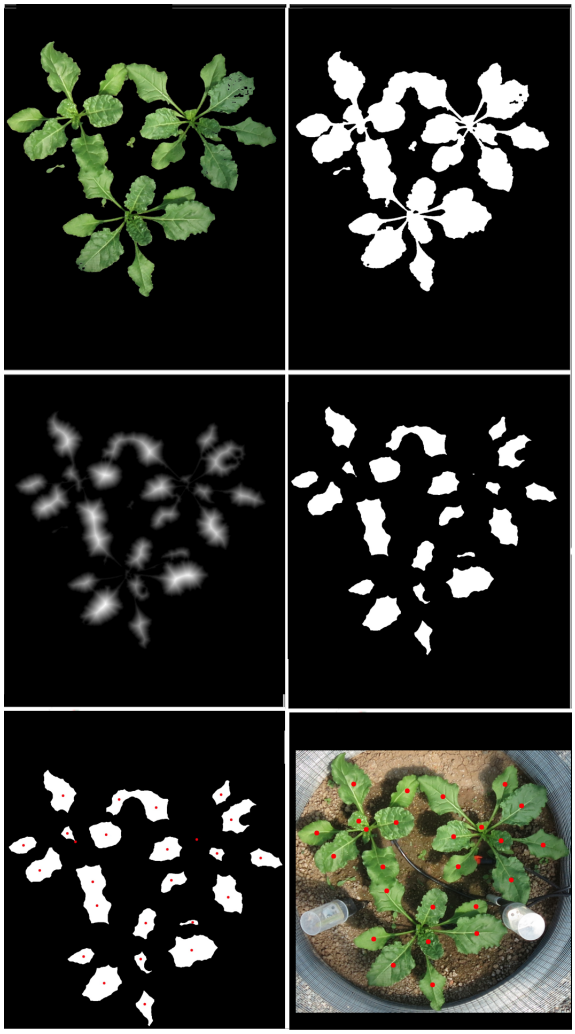


Figure 3.4: Mid stages of my method approach.

Chapter 4

Implementation

In this part the implementation of the solution project will be explained. This means here can be found the logical flow starting from the input image ending with the expected output. Every method inserted in the project will be followed by proper motivation and a picture to illustrate the eventual result after the appliance of the described method.

4.1 Work Environment

The first thing every engineer should do every time he approaches a new challenge, is to verify if the working environment is able to give him every necessary tool to deal with the problems he is going to face. In my case, the OS was Mac OS X 10.11, running on a MacBook Air with 4 GB of RAM and a dual-core Intel i5 processor with a clock of 1.7GHz. The IDE was Xcode 7.2. The libraries used to take advantage of various computer vision technique were OpenCV version 3.

4.2 Implementation of the Work-flow

Below an example of input image can be seen. Due to the semi-field environment, light condition are not optimal. Furthermore there are neighbour

plants on the sides, that could influence the analysis.

Figure 4.1: Example of Input Image

Once we are in possession of the input image, the work-flow executed is showed below. The reasoning followed is the same explained during the design phase.

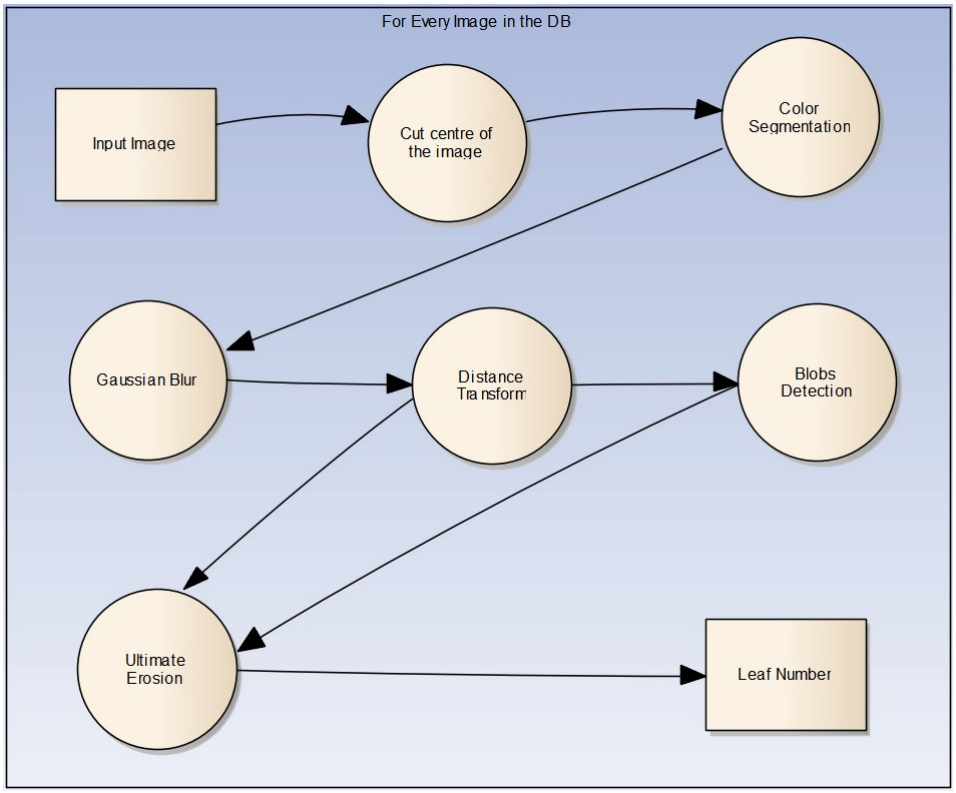


Figure 4.2: Implemented Algorithm Flow

ROI Modification The first pre-processing operation implemented was the reduction of the Region Of Interest. This allow to eliminate the neighbour plants and change the focus of our analysis on the central pot. This operation was easy to standardize because every pot was almost in the center of each picture. Thanks to that, taking the center and leaving out all the rest was enough.

Figure 4.3: Region Of Interest

Image Segmentation Following the ROI modification, the second necessary operation was to apply a segmentation through color threshold. The color threshold is aimed to have as output a black background with only the canopy as foreground. This segmentation is done using a different color space rather than the RGB color space used to read the original pixel values from the input image. HSV (Hue-Saturation-Value) was firstly implemented but a quick research made it clear that a L*a*b* approach would seem more reasonable. Both those color spaces put brightness/luminosity on a separate channel but L*a*b* remaining channel has to be subtracted among them to obtain a real color. HSV instead has all colors on the Hue channel so that a precise range of values has to be chosen.

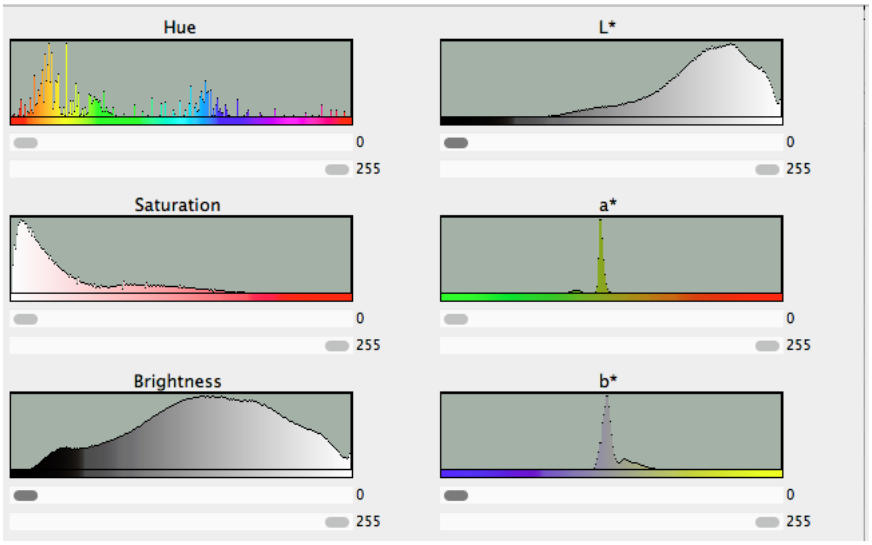


Figure 4.4: Color Space composition. Right: HSB/HSV; Left: L*a*b*.

As can be seen above, the "a*" channel has two peaks, corresponding on the soil area and to the canopy area. Since they are well separated it is easier to threshold compared to the Hue channel in HSV color space.

HSV Segmentation

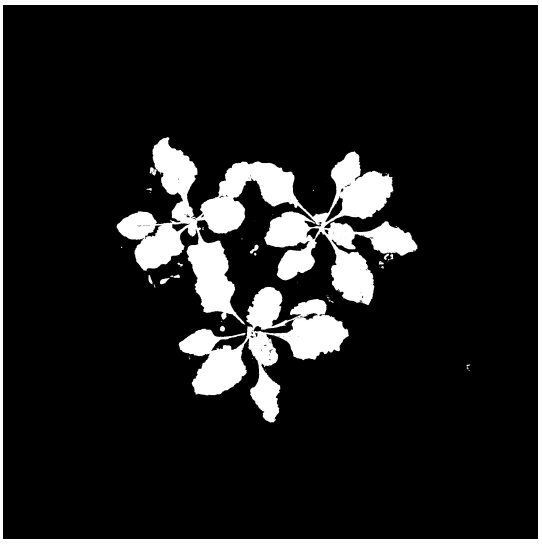


Figure 4.5: Image Segmentation using HSV Color Space

L*a*b* Segmentation Compared to the previous color space, color threshold using this color space capture a wider color range. I preferred to use a wider range instead of a too stringent one, with the risk of leaving out some detail.

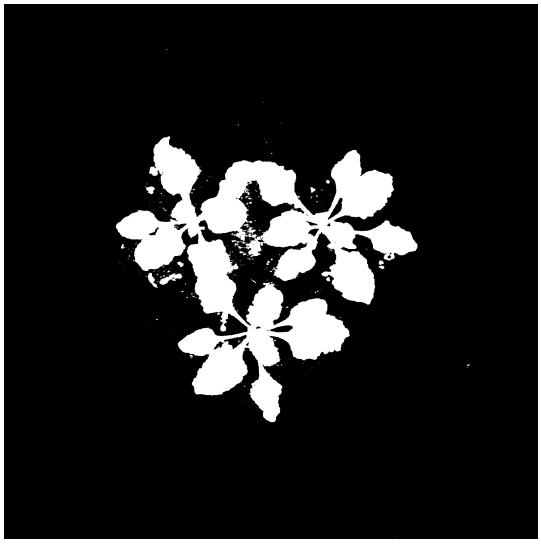


Figure 4.6: Image Segmentation using Lab Color Space

Morphological Operations Once the segmentation has given the proper output, it would probably be noisy due to the different light condition of each picture. A morphological operation is applied on black and white images and allows to fill holes or delete outsiders, but, in general, it is used to modify the shape of a white object over black background. The more simple morphological operation are called Erosion and Dilation. Here follows a brief explanation of those two operation:

- Dilation: This operations consists on the convolution of an image A with some kernel (B), which can have any shape or size, usually a square or circle, but even an ellipse. The kernel B has a defined anchor point, usually being the center of the kernel. As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by B and replace the image pixel in the anchor point position with that maximal value. As you can deduce, this maximizing operation causes bright regions within an image to "grow" (therefore the name dilation).

- Erosion: This operation is the sister of dilation. What this does is to compute a local minimum over the area of the kernel. As the kernel B is scanned over the image, we compute the minimal pixel value overlapped by B and replace the image pixel under the anchor point with that minimal value.

By using those two basic operation we can build up the Opening and Closing operations. Indeed the first one is obtained by the erosion of an image followed by a dilation. The second one instead is obtained by the dilation of an image followed by an erosion. In the following example an exaggerated version of the last two operators is applied, only to make the concept more clear.

Opening Useful for removing small objects (it is assumed that the objects are bright on a dark foreground). Used to remove outsiders like weed or to divide slightly overlapping objects.

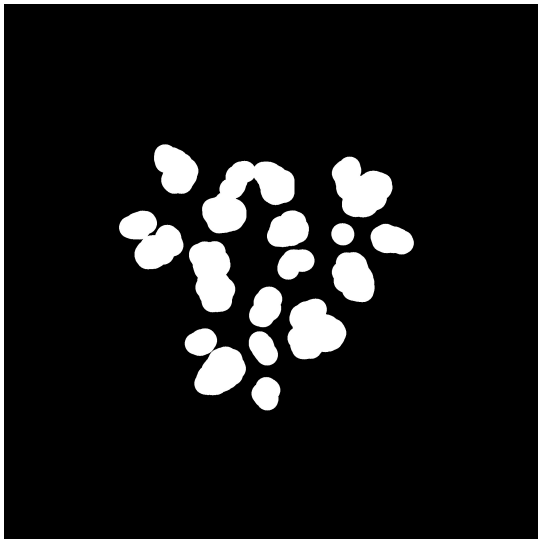


Figure 4.7: Morphological operation: Opening

Closing Useful to remove small holes (dark regions). It has the same assumption as the previous explained operation.

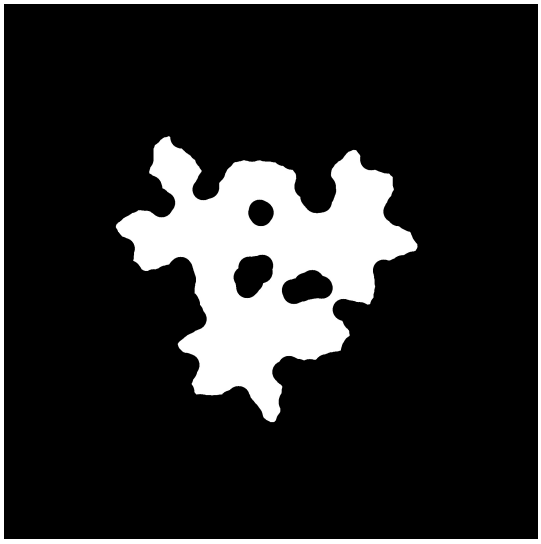


Figure 4.8: Morphological operation: Closing

Distance Transform The function Distance Transform calculate the approximate or precise distance from every binary image pixel to the nearest zero pixel (black pixel). For zero image pixels, the distance will obviously be zero.

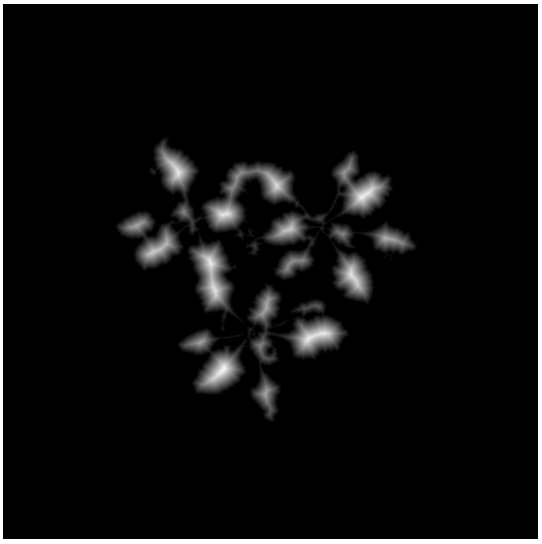


Figure 4.9: Distance Transform Output

Applying a simple clipping it is possible to take only the "peaks" of the Distance Transform. This operation allows us to take virtually only the central part of each object in our image (depending on the threshold), improving overlapped object separation.

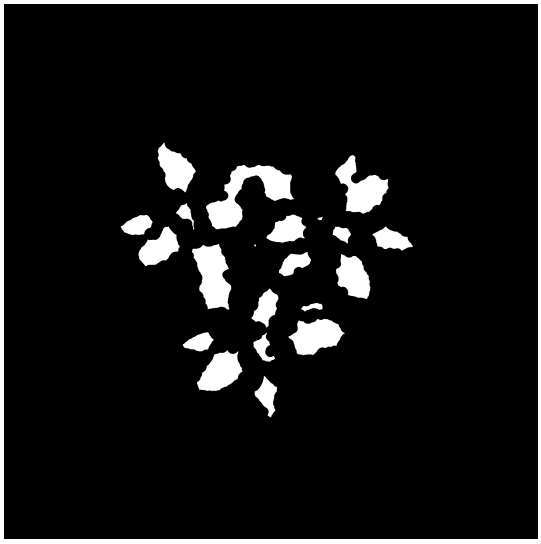


Figure 4.10: Peaks of the Distance Transform

Simple Blob Detection This simple operation finds in a black and white image all the connected white pixels and treat them as a "blob". This operation is fundamental since allows us to count objects, even discriminating between shape, area size, inertia or other parameters .

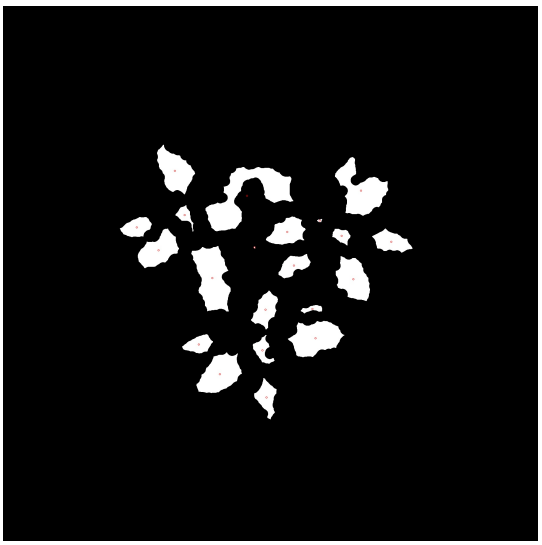
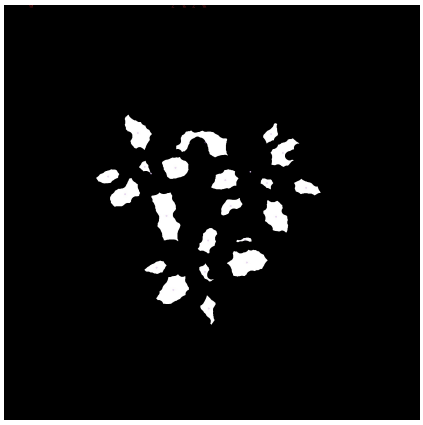
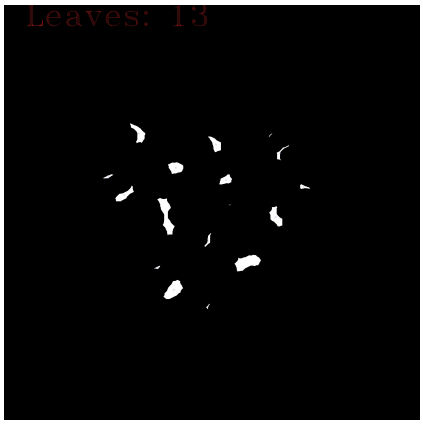


Figure 4.11: Simple Blob Detector

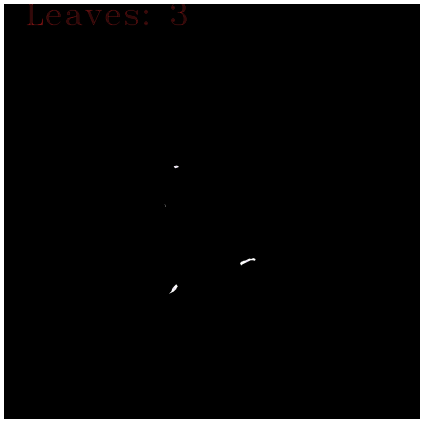
Ultimate Erosion This is the last operation implemented. Once we have got the peaks from the distance transform and the relative centre points we can proceed with the ultimate erosion. The operation in performed combining an erosion operation and a blobs detection to track the centre of each blob. This is repeated until we obtain a completely black image. We keep track of each blob centre and, as soon we observe that a blob in the next step is separated in two distinguishable blobs, we delete the previous centre and add the two new found.



(a) One of the first steps



(b) One of the middle steps



(c) One of the last steps

Figure 4.12: Steps of the Ultimate Erosion

4.3 Implemented LSC method

On the LSC method side, the work-flow was implemented as follows.

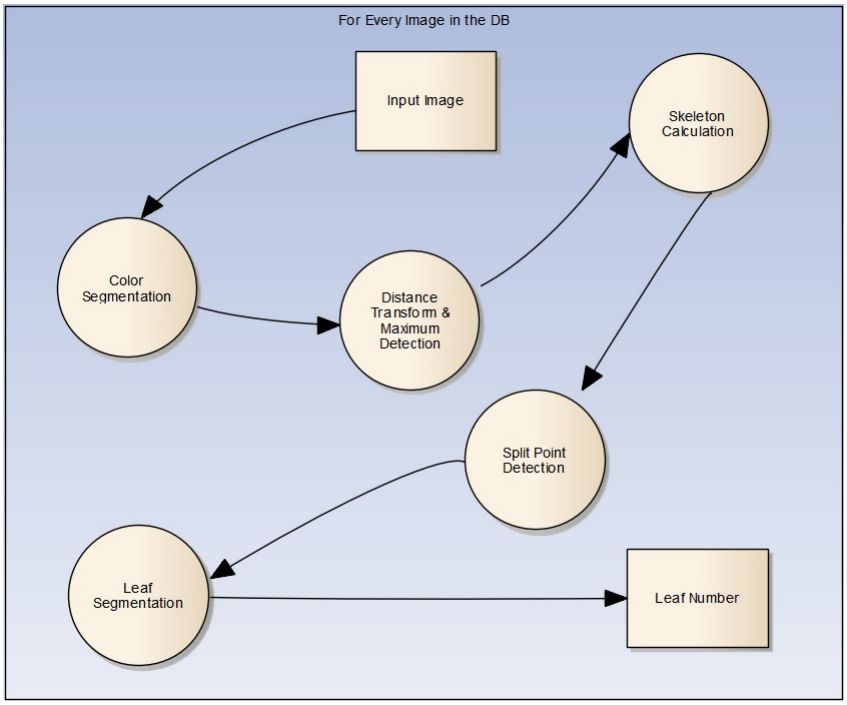


Figure 4.13: Implemented LSC Algorithm Flow

Since some of the methods are the same used on the developed algorithm, now only the new parts will be explained.

Skeleton In shape analysis, skeleton (or topological skeleton) of a shape is a thin version of that shape that is equidistant to its boundaries. The skeleton usually emphasizes geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width. Together with the distance of its points to the shape boundary, the skeleton can also serve as a representation of the shape (they contain all the information necessary to

reconstruct the shape). Skeletons have several different mathematical definitions in the technical literature, and there are many different algorithms for computing them. During my studies I have tested the two main implementation present in OpenCV, briefly explained in the next two sub-paragraphs. In the end I used the Zhang method because it was slightly faster than the Guo one.

Zhang Method The algorithm operates on all black pixels P_1 that can have eight neighbours. The neighbours are, in order, arranged as:

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

This methodology explains inside [25] the following work-flow:

while points are detected **do**
  for all pixels *p(i,j)* **do**
    if $(a)2 \leq B(P_1) \leq 6$
      $(b)A(P_1) = 1$
      (c) Apply one of the following:
      $1.P_2 * P_4 * P_6 = 0$ in odd iterations
      $2.P_2 * P_4 * P_8 = 0$ in even iterations
      (d) Apply one of the following:
      $1.P_4 * P_6 * P_8 = 0$ in odd iterations
      $2.P_2 * P_6 * P_8 = 0$ in odd iterations **then**
      Delete pixel *p(i,j)*
    end if
  end for
end while

Where $A(P_1)$ is the number of 0 to 1 transitions in a clockwise direction from P_9 back to itself, and $B(P_1)$ is the number of non-zero neighbours of P_1. The result of the previous algorithm can be seen in the following image.

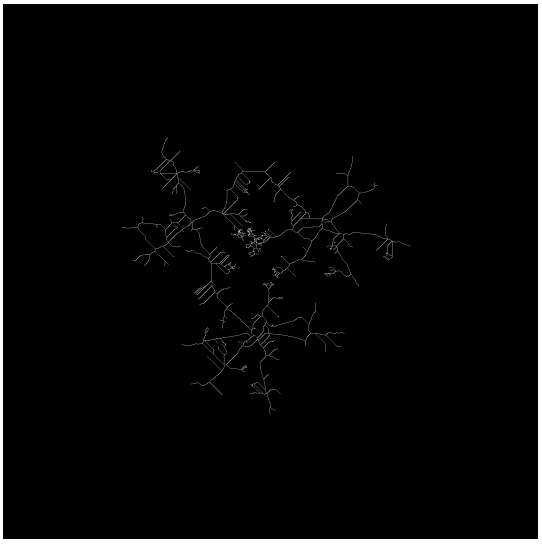


Figure 4.14: Skeleton output using Zhang algorithm

Guo-Hall Method This approach consideres the same pixel mask as Zhang-Suen but, differently from that approach, the Guo-Hall algorithm [26] reason as follows:

while points are detected **do**
  for all pixels *p(i,j)* **do**
    if $(a)C(P_1) = 1$
      $(b)2 \leq N(P_1) \leq 3$
      (c) Apply one of the following:
      $1.(P_2|P_3|P_5)\&P_4 = 0$ in odd iterations
      $2.(P_6|P_7|P_9)\&P_8 = 0$ in even iterations **then**
      Delete pixel *p(i,j)*
    end if
  end for
end while

Where:

- $C(P_1) = !P_2\&(P_3|P_4)+!P_4\&(P_5|P_6)+!P_6\&(P_7|P_8)+!P_8\&(P_1|P_2)$

- $N_1(P_1) = (P_9|P_2) + (P_3|P_4) + (P_5|P_6) + (P_7|P_8)$
- $N_2(P_1) = (P_2|P_3) + (P_4|P_5) + (P_6|P_7) + (P_8|P_9)$
- $N(P_1) = \min[N_1(P_1), N_2(P_1)]$

The result of the algorithm is slightly different from the previous one but reaches anyway the same objective.

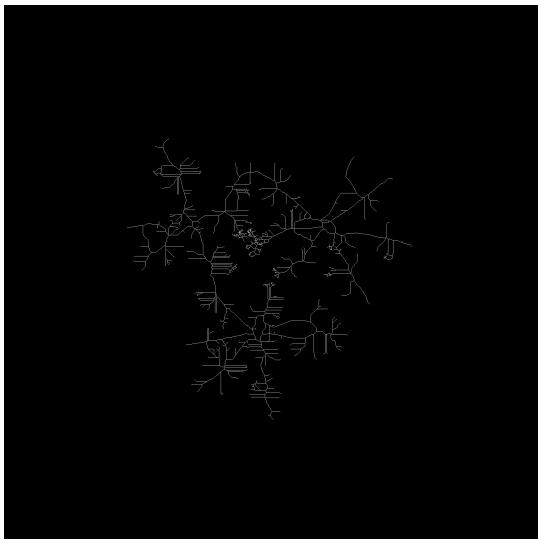


Figure 4.15: Skeleton output using Guo-Hall algorithm

Getting Joint Points Once obtained the skeleton of the subject, this function uses a LBP mask to explore the entire skeleton structure. Every time it finds an end point pixel or an intersection pixel it save them in the apposite structures. In the following image can be seen a representation of the result of the skeleton analysis. The algorithm has coloured every founded end point yellow and every founded intersection points red.

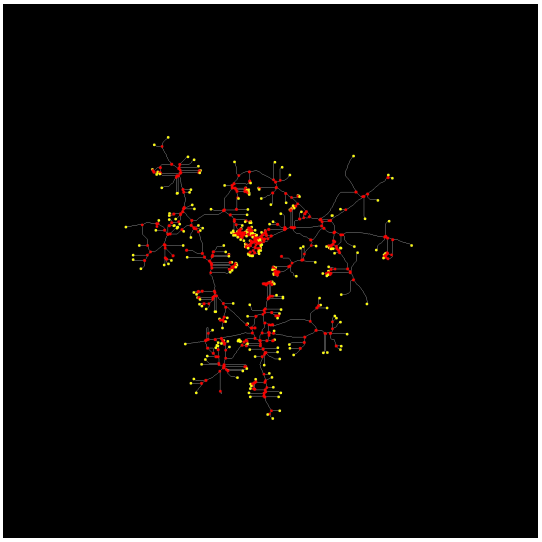


Figure 4.16: Joint Points detected on Skeleton image

Watershed Any grayscale image can be viewed as a topographic surface where high intensity denotes peaks and hills while low intensity denotes valleys. You start filling every isolated valleys (local minima) with different colored water (labels). As the water rises, depending on the peaks (gradients) nearby, water from different valleys, obviously with different colors will start to merge. To avoid that, you build barriers in the locations where water merges. You continue the work of filling water and building barriers until all the peaks are under water. Then the barriers you created gives you the segmentation result. This is the "philosophy" behind the watershed. But this

approach gives you over-segmented result due to noise or any other irregularities in the image. So OpenCV implemented a marker-based watershed algorithm where you specify which are all valley points are to be merged and which are not. It is an interactive image segmentation.

Figure 4.17: Watershed Algorithm

This is the crucial point were this algorithms fails under semi-field conditions and has not as the same good performance as under lab condition. Indeed we got a nice segmentation of leaf except where leaves overlaps. To deal with this problem in the paper the researcher uses the skeleton of the plant merged with the distance transform in order to obtain a graduated skeleton. They detect overlapping leaves by analysing this skeleton and detecting the minimum point, so the nearest point to the background along the skeleton. In our case we have got several of those points and I found no rules to discriminate between them in a unique way. Some time I was able to distinguish some but not others.

4.4 Implemented Midrib method

Canny The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works. The appliance was considered to enhance the leaves borders and midribs. The parameter were too different on each situation to generalize them so the final solution does not contain this method.

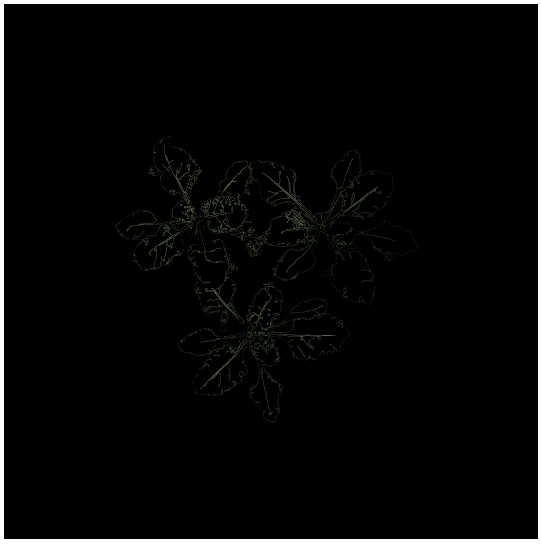


Figure 4.18: Canny Edge Detector

Segment detection Following the midrib approach, once enhanced the midribs of each leaf, those methods were implemented to properly detect enhanced midribs and count them. In addition, with this mechanism, we should have had a first mensuration of the length of the midrib. One of the most challenging activities in computer vision is the extraction of useful information from a given image. Such information, usually comes in the form of points that preserve some kind of property (for instance, they are scale-invariant) and are actually representative of input image.

Hough Line Detector The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform.

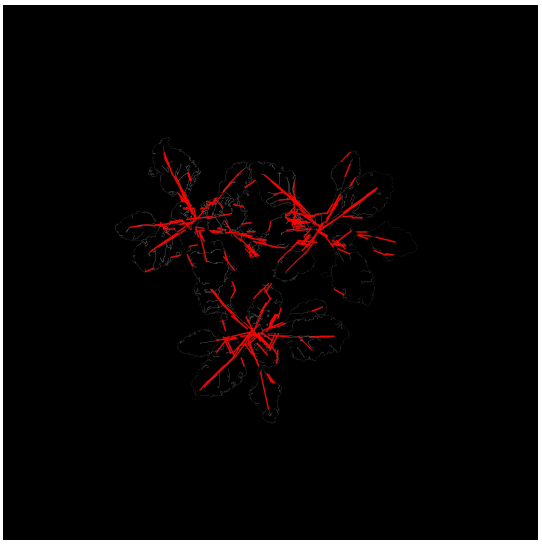


Figure 4.19: Hough Line Detector

Binary Detector The description of this algorithm is present in [23]. The BinaryDescriptor Class present in OpenCV implements both functionalities for detection of lines and computation of their binary descriptor.

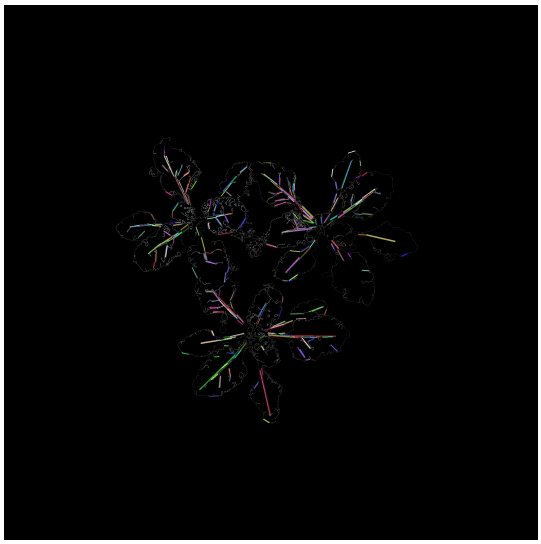


Figure 4.20: Binary Line Detector

LSD: Line Segments Detector The lines extraction methodology showed in the following is mainly based on [24]. The extraction starts with a Gaussian pyramid generated from an original image, downsampled N-1 times, blurred N times, to obtain N layers (one for each octave), with layer 0 corresponding to input image. Then, from each layer (octave) in the pyramid, lines are extracted using LSD algorithm. Differently from EDLine lines extractor used in original article, LSD furnishes information only about lines extremes; thus, additional information regarding slope and equation of line are computed via analytic methods. The number of pixels is obtained using LineIterator. Extracted lines are returned in the form of KeyLine objects, but since extraction is based on a method different from the one used in BinaryDescriptor class, data associated to a line's extremes in original image

and in octave it was extracted from, coincide. KeyLine's field classid is used as an index to indicate the order of extraction of a line inside a single octave.

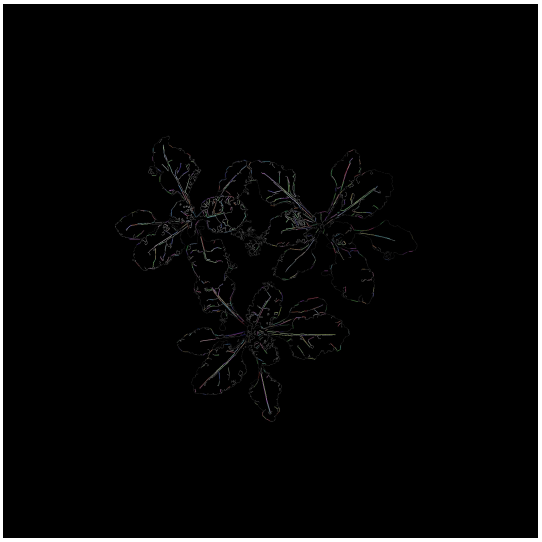


Figure 4.21: LSD Result

Problem: More than one segment to single Midrib Although this approach could be interesting because it give us, in one shot, both the counting and the lengths of the midribs, there is at least one downside. Each one of this methods detected, in fact, more than one segment per midrib. This is due to the non-straightness of the midribs. Some segments are even overlapped, so there is no easy way to distinguish/merge/concatenate them in a way to keep all important information in all the situations.

Contours Finder It is easy to understand the goal of this function. It aims to trace the contour of every gray-scale object over black background present in the input image. This method is also able to trace inner contours, depending on the amount of data that we need. It is also possible to obtain gross or fine hierarchy between contours. During the experimental part I needed this method while either trying to enhance the midrib and focusing on the external shape of the leaf. In the last part I was aiming to draw the convexity hull of each plant, to count the leaf number by the convexity defects.

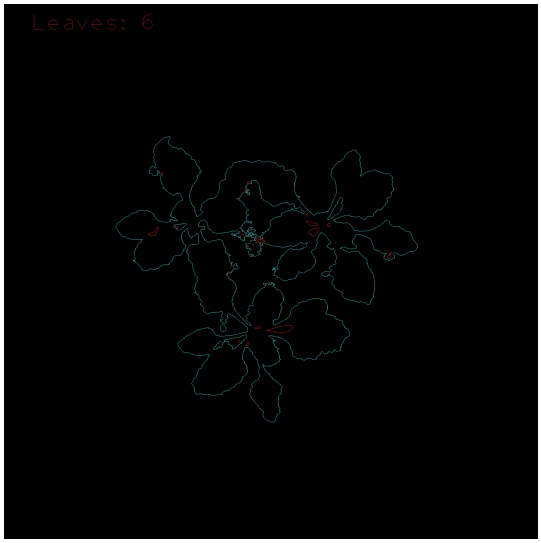


Figure 4.22: Find Contour algorithm result

Histogram Equalization And Clahe Since midribs are not always clear over the leaf, because of the variable light conditions, I looked for some techniques to enhance the contrast of the pictures. Histogram equalization is a method that improves the contrast in an image, in order to stretch out the intensity range. To make it clearer, sometime, from the histogram of an image, you can see that the pixels seem clustered around the middle of the available range of intensities. What Histogram Equalization does is to stretch out this range.

Figure 4.23: Histogram Equalization Algorithm result

But not always the image histogram is confined to a particular region. To solve this problem, adaptive histogram equalization is used. In this, image is divided into small blocks called "tiles" (tile size is 8x8 by default in OpenCV). Then each of these blocks are histogram equalized as usual. So in a small area, histogram would confine to a small region (unless there is noise). If noise is there, it will be amplified. To avoid this, contrast limiting is applied. If any histogram bin is above the specified contrast limit (by default 40 in OpenCV), those pixels are clipped and distributed uniformly

to other bins before applying histogram equalization. After equalization, to remove artefacts in tile borders, bilinear interpolation is applied.

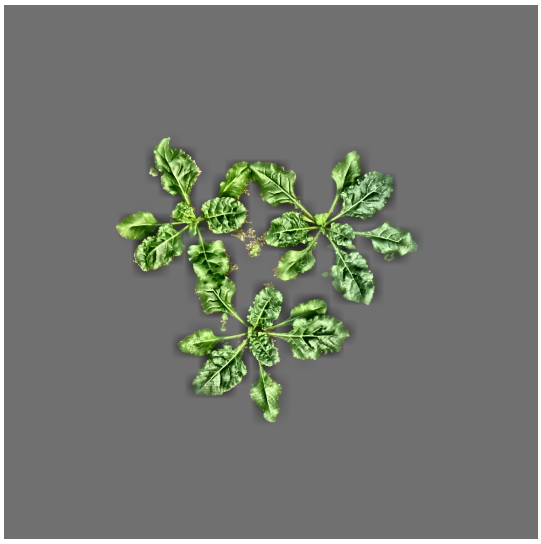


Figure 4.24: Clahe output

Gabor Trying to sharpen the midribs, I went deep in the edge detection. A Gabor filter, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. It follows the following formula:

$$g_{\lambda,\theta,\phi,\sigma,\gamma} = exp(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2})cos(2\pi\frac{x'}{\lambda}+\phi)$$

with:

$$x' = xcos\theta + ysin\theta$$
$$y' = -xsin\theta + ycos\theta$$

It uses several parameters which are:

- wavelength (λ): This is the wavelength of the cosine factor of the Gabor filter kernel and herewith the preferred wavelength of this filter. Its value is specified in pixels. Valid values are real numbers equal to or greater than 2. The value λ=2 should not be used in combination with phase offset -90 or 90 because in these cases the Gabor function is sampled in its zero crossings. In order to prevent the occurrence of undesired effects at the image borders, the wavelength value should be smaller than one fifth of the input image size.

- Orientation(s) (θ): This parameter specifies the orientation of the normal to the parallel stripes of a Gabor function. Its value is specified in degrees. Valid values are real numbers between 0 and 360.

- Phase offset(s) (ϕ):The phase offset in the argument of the cosine factor of the Gabor function is specified in degrees. Valid values are real numbers between -180 and 180. The values 0 and 180 correspond to center-symmetric 'center-on' and 'center-off' functions, respectively, while -90 and 90 correspond to anti-symmetric functions. All other cases correspond to asymmetric functions.

- Aspect ratio (γ):This parameter, called more precisely the spatial aspect ratio, specifies the ellipticity of the support of the Gabor function. For $\gamma = 1$, the support is circular. For $\gamma < 1$ the support is elongated in orientation of the parallel stripes of the function. Default value is $\gamma = 0.5$.

- Bandwidth (b):The half-response spatial frequency bandwidth b (in octaves) of a Gabor filter is related to the ratio $\frac{\sigma}{\lambda}$, where σ and λ are the standard deviation of the Gaussian factor of the Gabor function and the preferred wavelength, respectively, as follows:

$$b = \log_2 \frac{\frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}}, \frac{\sigma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}}\frac{2^b+1}{2^b-1}$$

The value of σ cannot be specified directly. It can only be changed through the bandwidth b. The bandwidth value must be specified as a

real positive number. Default is 1, in which case σ and λ are connected as follows: $\sigma = 0.56\ \lambda$. The smaller the bandwidth, the larger σ, the support of the Gabor function and the number of visible parallel excitatory and inhibitory stripe zones.

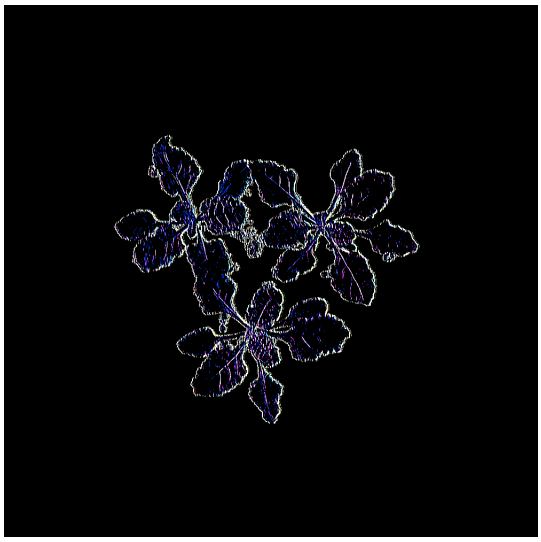


Figure 4.25: Gabor output, results may vary depending on parameters

LaPlace Transform Another edge detector is the LaPlace transform. The LaPlacian operator is used to highlight edge inside a picture. The reasoning behind the implementation is the one that follows. It is known that in edge areas, the pixel intensity shows a "jump" or a high variation of intensity. Getting the first derivative of the intensity, we can observe that an edge is characterized by a maximum. This can be archived using the Sobel operator. If we take the second derivative, we can observe that this is zero in the previous maximum point. So we can also use this method to detect edges. It is right to point out that zeros will not only appear in edges, but can actually appear in other meaningless locations. This can be solved by applying filtering where needed.

Filtering Applying The principal filers implemented inside OpenCV are the following. In the images is showed how each filter affects (on the left) the segmented image and (on the right) the result of the LaPlace operator.

- Homogeneous Filter
  This kind of filter is the most simple kind. The kernel K of this filter can be described as follows:

$$K = \frac{1}{ksize.width * ksize.height} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

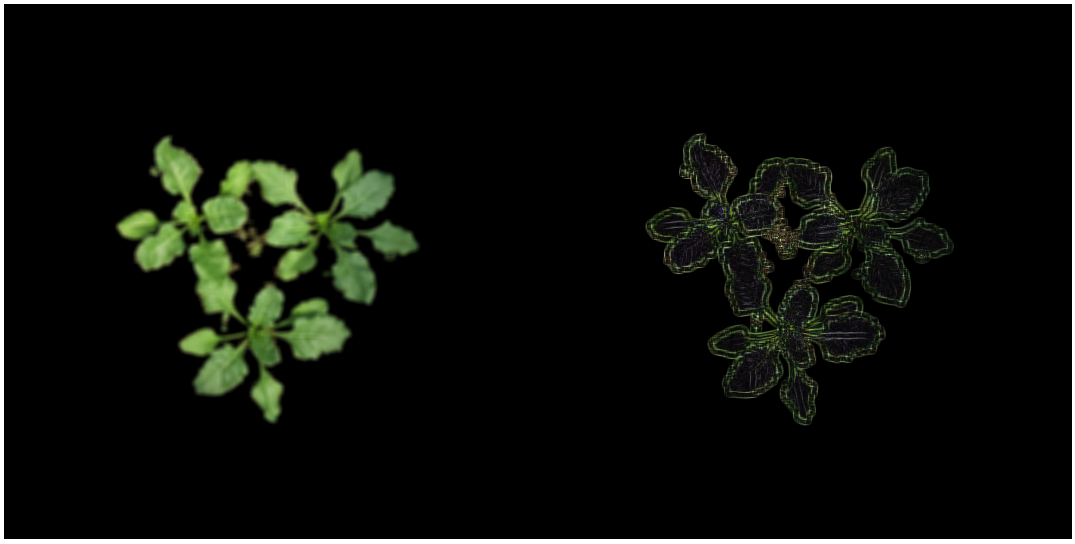


Figure 4.26: Homogeneous Filter

- Mean Filter
  The filter smooths an image using the median filter with the $ksize * ksize$ aperture. Each channel of a multi-channel image is processed independently.

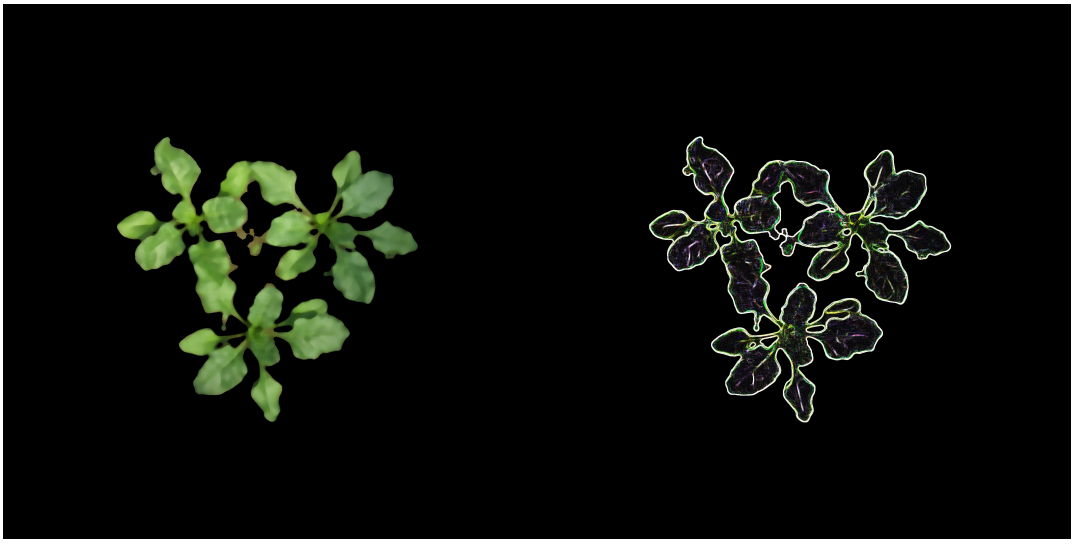


Figure 4.27: Median Filter

- Gaussian Filter
  The function convolves the source image with the specified Gaussian kernel. The main goal of this kind of filter is to aim the focus on the principal subject of the picture.

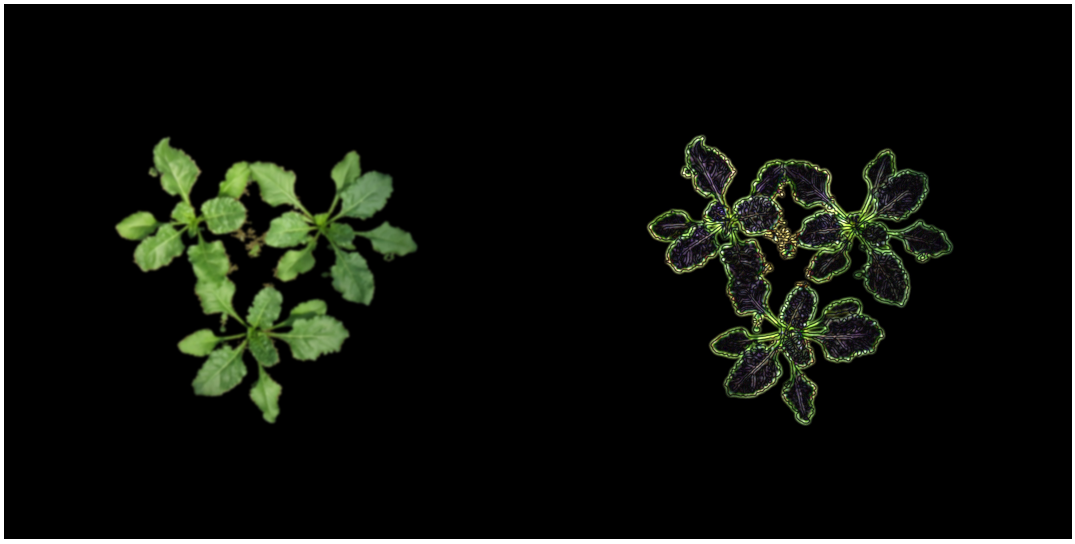


Figure 4.28: Gaussian Filter

- Bilateral Filter

  The last type of filter aim to prevent averaging across edges, while still averaging within smooth regions. Bilateral filtering is a simple, non-iterative scheme for edge-preserving smoothing. It can reduce unwanted noise very well while keeping edges fairly sharp. However, it is very slow compared to most filters.

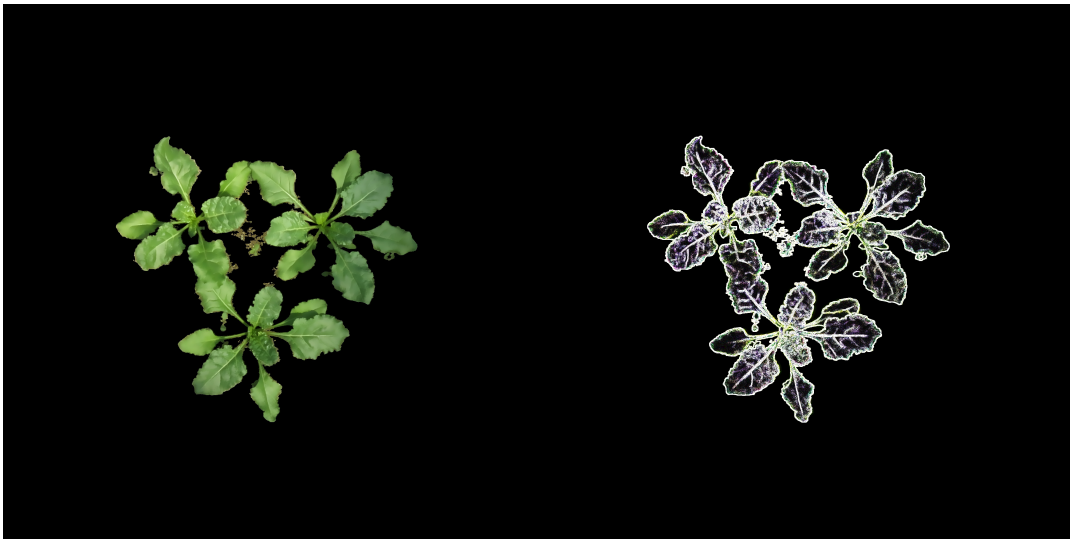


Figure 4.29: Bilateral Filter

Scharr My last attempt with edge detection techniques was the Scharr operator. Starting with the same goal as a Sobel operator, Scharr operator aims to optimize the rotational symmetry propriety. Similarly to Sobel, Scharr is used within edge detection algorithms where it creates an image emphasising edges. From a mathematical point of view, this operator generates two images, starting from an input picture, as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A$$

In those formulas A represents the input image, $*$ is the convolution operator and finally G_x and G_y are the output images, the horizontal and vertical derivative approximations respectively.

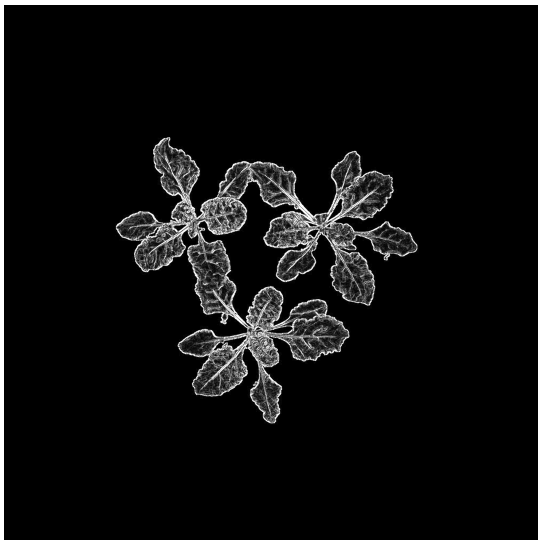


Figure 4.30: Scharr output

Chapter 5

Experimental Results

In this chapter will be explained how the ground truth was collected in order to measure the goodness of the developed algorithm. Starting from that bases, a performance test will be executed, testing the sensibility and specificity of both the algorithm implemented. At the end of the chapter could be found some conclusions on the data analysis and some recommendations and ideas for future implementations/extensions.

5.1 Previous Solution

This approach has the same weakness as those discussed above, but has some similarities with the setup described on this Thesis. Due those similarities it will be taken as example to compare the results obtained. This thesis basic idea is based on semi-field conditions, an intermediate conditions between a controlled environment and a field-like condition.

Researchers in Syngenta used a digital camera Canon S100 (Canon, Tokyo, Japan). Visible images were captured from emergence every two or three days. The camera was placed on a static monopod positioned on an easily movable trolley. Images were obtained from 0.8 m above plant canopy with a resolution of $0.2\frac{cm^2}{pixel}$. Plants were handled by hand from the trolley to the monopod frame. Images were taken under low light using the automatic

settings of the camera. Taken pictures were processed using the software ImageJ [18]: those were split according to the 3 RGB channels, than was created a new image according to the Excess Green Index (EGI) proposed by [19] as: EGI = 2G - B - R. A threshold was then applied to segment the newly generated image. Pixels with intensities from 55 to 255 were considered "green pixels". The total number of "green pixels" was calculated and subsequently re-scaled to obtain the final parameter "digital canopy area" expressed in cm^2. As reference, in every taken picture was present a 5 cm length bar. This kind of approach allowed Samuel to obtain the following results:

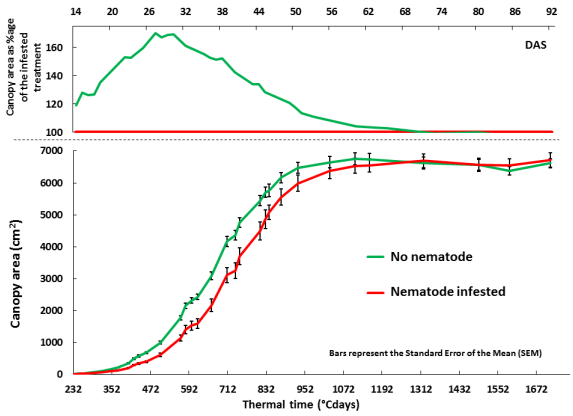


Figure 5.1: Green line represent a healthy bunch of plants and the red line represent an infested bunch of plants.

The upper graph represents the difference in terms of canopy area during the whole life cycle between nematode infested and non-infested treatments. The lower graph enhances the stunted growth of the canopy area in infested plants.

5.2 Midrib - Leaf Area Correlation

During the first week in Syngenta, personal of the RU and me spent some time in the semi-field capturing pictures of leaves at different growth stages. These pictures were captured trying to make the leaf as flat as possible and capturing the picture with and high contrast background. Then, pictures were processed to calculate the leaf area and the corresponding midrib length. To do so, a reference were present in every picture, known this fixed length I was able to calculate the leaf area. As shown in the table below, there is a strong correlation between midrib length and total leaf area.

Table 5.1: Leaves data from manual calculation

Midrib Length (cm)	Pixel Area (pixel)	Calculated Area (cm^2)
12.11	224,603	108.8
12.34	208,593	106.92
7.75	92,274	41.14
22.57	169,276	388.12
16.6	97,779	192.98
11.84	211,438	96.44
13.15	91,276	132.12
18.81	140,719	218.13
11.74	218,692	87.71
6.76	244,537	27.32
12.79	219,145	101.42
16.66	82,781	167.87
7.16	120,716	54.85
4.21	141,406	16.29
15.23	79,889	138.31
21.82	169,497	348.89
21.59	147,035	238.42
12.91	171,338	84.92
16.04	83,030	143.18
7.1	63,064	30.86
16.4	101,200	175.73
17.2	84,790	143.41
12.65	219,558	96.46
22.56	170,550	286.47
13.79	264,684	112.25
6.28	75,259	37.21
23.89	162,047	312.84

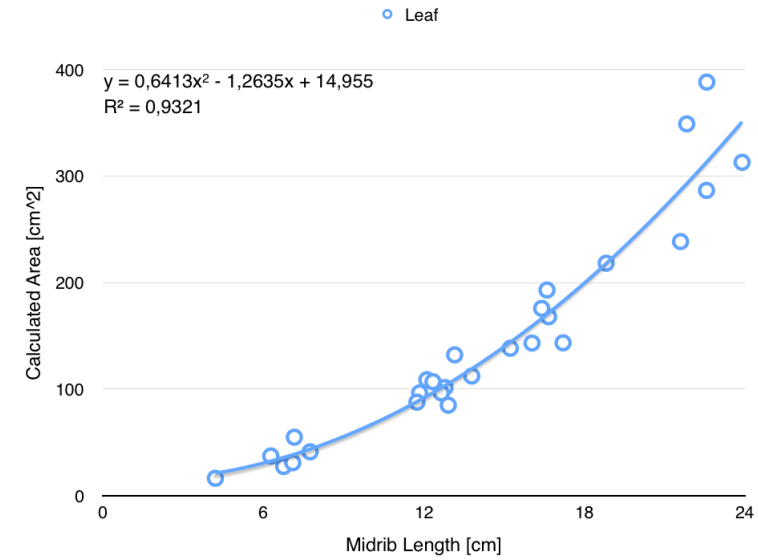


Figure 5.2: Trending line graph - strong correlation detected.

5.3 Ground Truth

The ground truth is composed by either healthy and nematode infested plants canopy picture. Picture has been taken during 4 different stage of growth: 22 DAS, 27 DAS, 29 DAS and 32 DAS with "DAS" equals to Days After Sowing. For each period we have got 10 picture per both healthy and nematode infested plants. Each picture represents a pot containing 3 plants, according to the semi-field condition. The leaf of each plant has been counted manually by looking at the taken picture and not in the field. As additional detail, it is good to point out that cotyledons are easily detectable in every plants because of the early stage of growth. A cotyledon is a significant part of the embryo within the seed of a plant. Upon germination, the cotyledons usually become the embryonic first leaves of a seedling. The number of cotyledons present is one characteristic used by botanists to classify the flowering plants (angiosperms). Sugar beet is classified as dicotyledon plant, meaning that it has 2 cotyledons.

Figure 5.3: Example of taken pictures. On the left a 22DAS plant. On the right the same plant at 32DAS.

Since we did not discriminate among leaves and cotyledons, the algorithm identifies both of them. In this study we will consider the detection of a cotyledon as a False Positive. In future implementation could be interesting to add a discriminative mechanism. The rows represent the upper left, upper right and bottom plant of an non-infested and infested pot.

Table 5.2: Number of Leaves Ground Truth

		22										27										29										32									
Healthy	First	2	3	3	4	3	3	4	3	4	3	4	6	6	6	6	5	6	4	6	6	5	6	7	6	6	6	6	5	7	6	7	8	9	9	7	8	7	7	9	9
	Second	3	4	3	4	3	4	3	4	3	4	5	6	5	7	5	6	5	6	5	7	6	7	6	7	6	7	6	6	6	8	7	9	8	10	8	8	8	8	8	11
	Third	2	4	3	3	4	3	4	3	4	3	4	6	5	6	5	6	6	6	7	5	5	6	6	6	6	6	7	6	7	6	7	8	8	7	7	9	10	7	10	7
Infested	First	2	4	3	2	2	4	3	4	2	3	4	6	5	3	4	6	4	6	5	5	5	6	6	4	4	7	4	6	7	6	6	8	7	5	6	8	8	7	9	8
	Second	3	3	2	2	3	3	3	3	2	3	5	5	4	4	5	5	6	5	3	4	6	5	4	4	5	6	6	6	3	5	7	6	6	5	7	8	9	7	3	6
	Third	3	3	4	3	3	2	2	3	3	4	4	5	5	5	5	4	5	4	5	5	6	6	6	6	5	5	5	6	5	6	7	8	8	7	6	7	8	7	7	7

5.4 Results Generation

Here represented the manual counting of the results of the implemented algorithms:

		22										27										29										32									
Healthy	First	6	4	5	4	6	4	4	5	5	5	5	6	6	6	5	6	6	4	7	5	5	6	7	6	6	5	6	5	7	6	8	8	7	8	7	7	6	6	7	8
	Second	5	6	3	5	5	5	5	5	4	5	5	6	5	6	5	6	5	6	5	8	6	7	6	7	6	7	6	7	7	9	9	10	7	7	5	8	6	8	8	10
	Third	4	5	5	5	6	5	4	7	5	5	5	5	5	6	6	5	6	5	6	6	5	6	5	6	7	6	6	6	7	6	8	8	7	6	7	6	7	9	9	6
Infested	First	4	4	6	3	3	3	4	4	2	3	4	5	5	4	5	7	5	5	8	7	4	6	5	4	6	6	3	6	6	6	6	8	7	4	5	7	7	5	8	8
	Second	5	5	4	4	5	4	5	6	4	6	5	5	4	5	5	5	6	5	5	4	6	5	4	4	5	8	6	6	3	3	6	7	6	6	7	6	7	6	2	5
	Third	5	5	4	4	3	3	3	5	5	5	4	5	5	5	5	4	5	4	5	5	5	5	6	6	5	5	5	5	5	6	9	10	8	6	5	5	7	6	8	7

Table 5.3: Results of the final method.

		22										27										29										32									
Healthy	First	2	5	5	6	4	4	5	2	5	4	4	4	5	7	5	5	5	4	7	6	5	6	7	6	6	5	5	5	5	5	7	8	10	6	8	6	7	5	5	6
	Second	2	5	4	4	2	4	4	4	4	6	5	5	5	7	6	6	5	6	5	7	6	7	6	6	6	6	6	6	5	6	6	8	5	6	4	5	6	7	9	6
	Third	2	5	4	2	5	3	4	3	5	4	6	6	4	5	5	5	6	5	7	5	5	5	6	6	6	6	6	6	5	9	7	4	6	4	8	10	5	6	5	
Infested	First	3	5	5	2	0	4	2	6	3	3	4	6	4	3	5	5	3	5	4	4	4	6	6	4	4	5	3	6	7	6	7	16	7	3	5	5	7	5	6	8
	Second	5	2	2	2	4	4	4	4	1	3	4	4	4	4	5	5	5	4	2	4	5	5	4	4	5	6	6	6	3	4	5	6	5	6	4	5	5	7	2	5
	Third	2	3	3	2	4	4	3	4	3	5	5	5	5	5	3	4	4	4	5	5	5	5	6	6	5	5	4	5	5	5	6	13	7	4	5	5	4	5	5	6

Table 5.4: Results of the paper method.

I was able to extract the following conclusions:

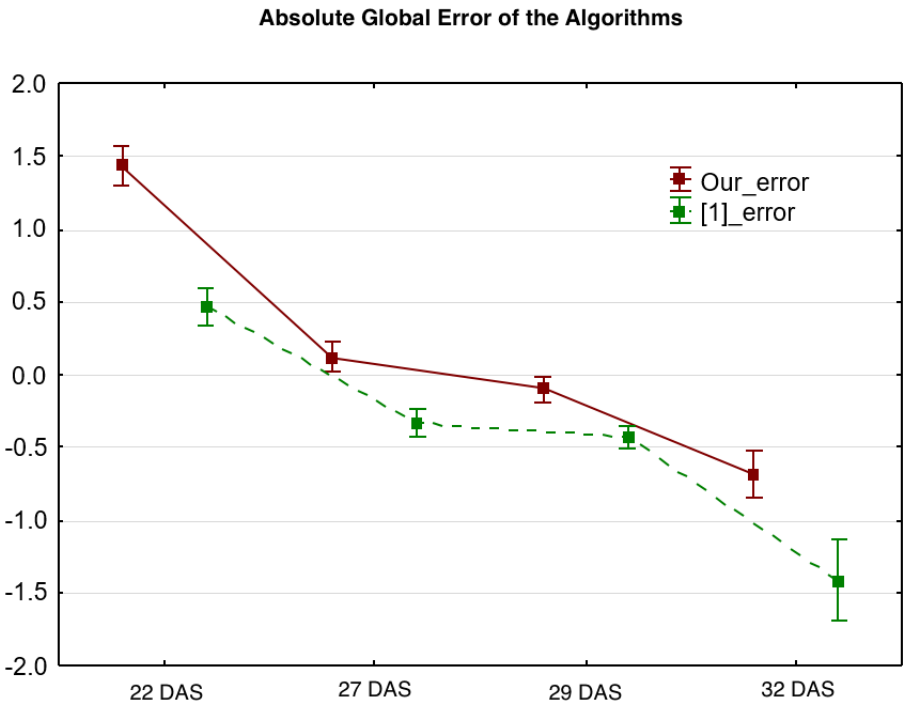


Figure 5.4: Leaves number absolute estimated error over DAS.

The graph above shows the leaves number absolute estimated error. It is easy to calculate that the proposed algorithm has a global absolute estimated error of 0.85. This result, compared to the 0.98 scored by the paper algorithm indicate that we are more accurate, over the analysed DAS, with the proposed algorithm. Moreover the absolute estimated error could be further decreased using two precautions:

- eliminate weed from the pots. Weed can lead to false positive recognition;
- add a leaf shape-based mechanism, which is able to discriminate from cotyledons and real leaves.

5.4.1 Performance Tests

Going deeper in the analysis, the performance parameter where as follows:

- Total Leaf Number = 1309: Obtained by manually counting the leaves on every plant on picture.
- Total Detection Number = 1351: Obtained, after visualizing the detection point over the input image, counting them on every plant on picture. Every point was associated to the respective plant.
- True Positive (TP) = 1194: Calculated, among the detected points, as the leaf correctly detected by the algorithm.
- False Positive (FP) = 157: Calculated, among the detected points, as the total number of detection not corresponding to leaves.
- False Negative (FN) = 115: Calculated as the total number of leaf not detected.

With those parameters we are able to calculate some performance index:

- Sensitivity / True Positive Rate (TPR): 0,91
  Sensitivity refers to the test's ability to correctly detect leaf. This index represent the total real leaf detected over the total number of detection.
- Precision / Positive Predictive Value(PPV): 0,88
  The positive predictive values (PPV) is the proportion of positive results in statistics and diagnostic tests that are true positive results. A high result can be interpreted as indicating the accuracy of such a statistic. The PPV is not intrinsic to the test; it depend also on the prevalence (that is, generally, the proportion of a population found to have a condition).
- False Negative Rate (FNR): 0,09
  The false negative rate is the proportion of events that are being tested

for which yield negative test outcomes with the test, i.e., the conditional probability of a negative test result given that the event being looked for has taken place.

- False Discovery Rate (FDR): 0,12
  The False discovery rate (FDR) is one way of conceptualizing the rate of type I errors (the incorrect rejection of a true null hypothesis [1] , a "false positive") in null hypothesis testing when conducting multiple comparisons. Null-hypothesis means a general statement or default position that there is no relationship between two measured phenomena, or no difference among groups. FDR-controlling procedures are designed to control the expected proportion of rejected null hypotheses that were incorrect rejections ("false discoveries"). FDR-controlling procedures provide less stringent control of Type I errors compared to family-wise error rate (FWER) controlling procedures (such as the Bonferroni correction), which control the probability of at least one Type I error. Thus, FDR-controlling procedures have greater power, at the cost of increased rates of Type I errors.

- F1 score: 0,90
  F1 score considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results, and r is the number of correct positive results divided by the number of positive results that should have been returned. The F1 score can be interpreted as a weighted average of the precision and recall, where a F1 score reaches its best value at 1 and worst at 0.

The absence of the True Negative parameter does not allow to calculate Specificity as much as other important performance indexes. The True Negative parameter could be calculated manually pointing out, at plant level

[1] null hypothesis = no relationship between two measured phenomena

and not pot level, the areas where NOT to find leaves and then count all the pixel that respected the definition of True Negative.

5.5 Results Analysis

In this section I am going to explore the statistical results obtained analysing the results obtained by both the algorithm inspected. It is important to distinguish among to different kinds of results we are looking for: the ability of precisely count leaves and the capacity to discriminate among infested and non infested plants.

5.5.1 Application Domain

After the statistical analysis we have every element to answer the main question of this project: can we always discriminate between infested and non infested plants? Using the paper algorithm we are not always able to distinguish between infested and non infested, due to the poorer detection performance. Instead, using the developed algorithm, even if the leaf number is not always correct, in the dates examined, we are always able to distinguish among infested and non infested plants.

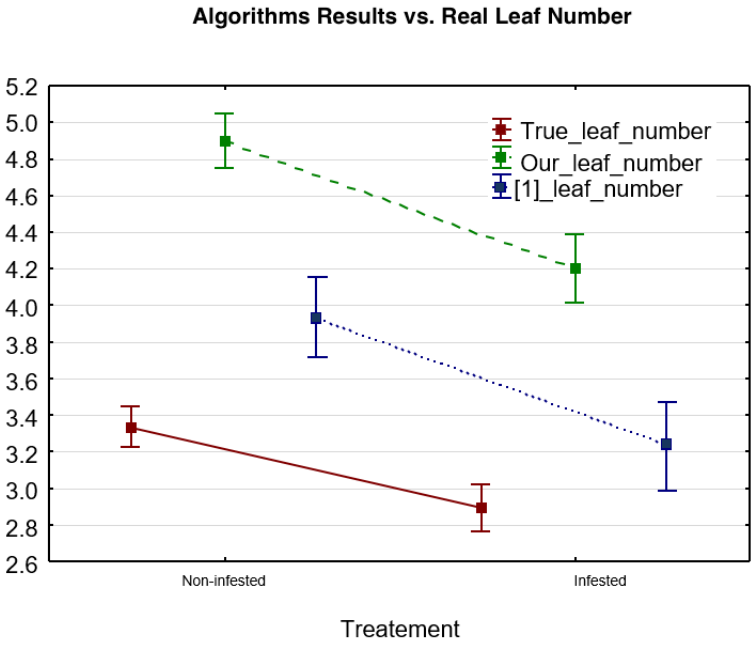


Figure 5.5: 22 DAS results.

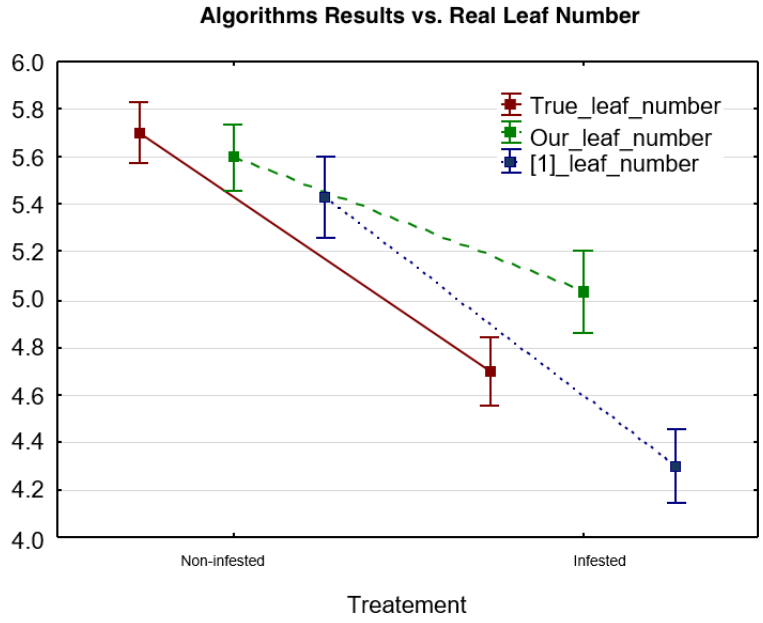


Figure 5.6: 27 DAS results.

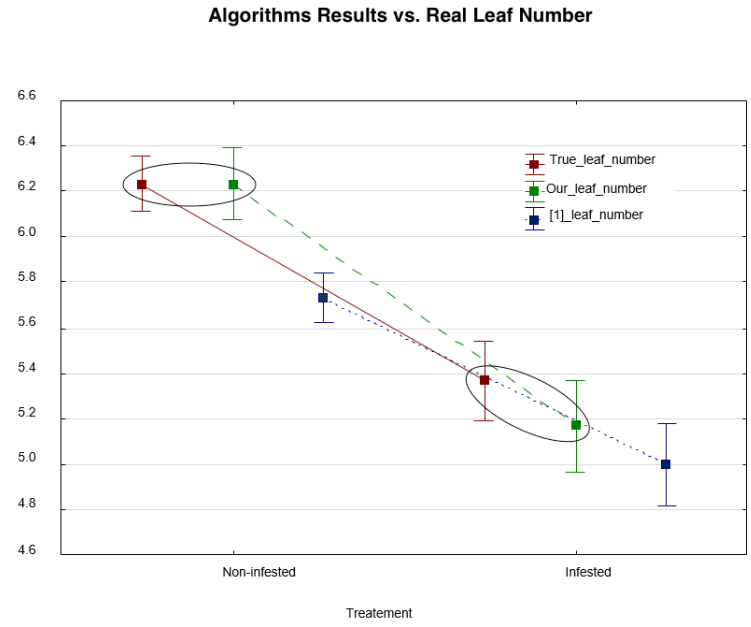


Figure 5.7: Special case: 29 DAS perfect result.

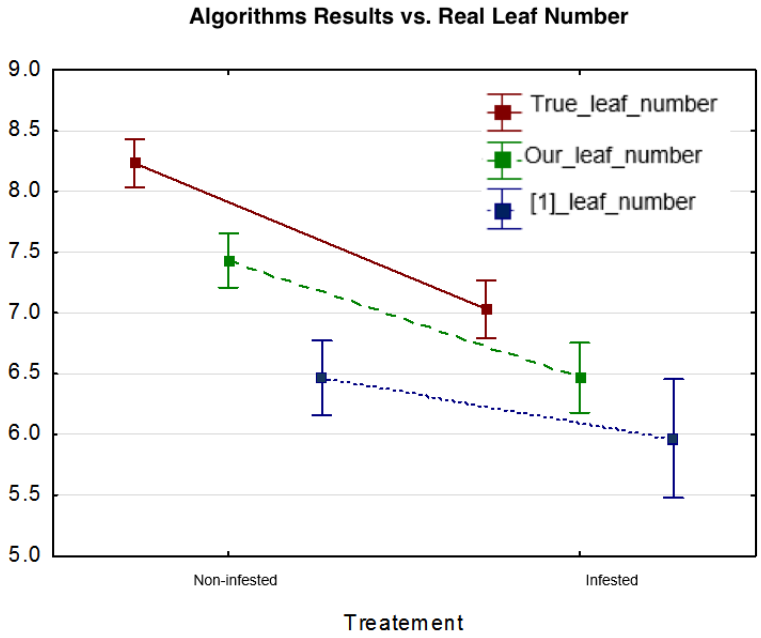


Figure 5.8: 32 DAS results.

The 29 DAS analysis showed a perfect leaves detecting on both infested and non-infested regarding the proposed algorithm. It is important to point out that the average score archived by the developed algorithm is the nearest to the real leaf number average, being able to always discriminate among infested and non-infested plants. The [1] algorithm instead does not discriminate infested and non-infested for 32 DAS.

5.5.2 Application Independent

In this part the statistical sensitivity and precision of both the algorithms is compared in a way to give a quantitative measurement of the goodness of the work done. Applying the same ground truth as input for both the algorithms gave me different detection results. Here follows the graphs that shows a comparison between real leaves number, the detected leaves number by the developed algorithm and the detected leaves number by the compared algorithm. Each bar of each graphs represents a pot.

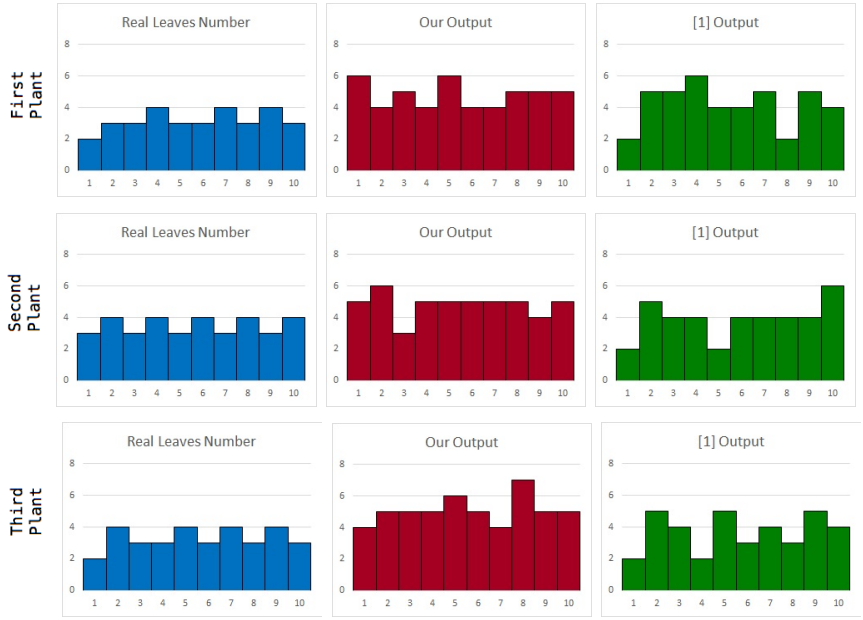


Figure 5.9: 22 DAS, Non-Infested

Figure 5.10: 27 DAS, Non-Infested

Figure 5.11: 29 DAS, Non-Infested

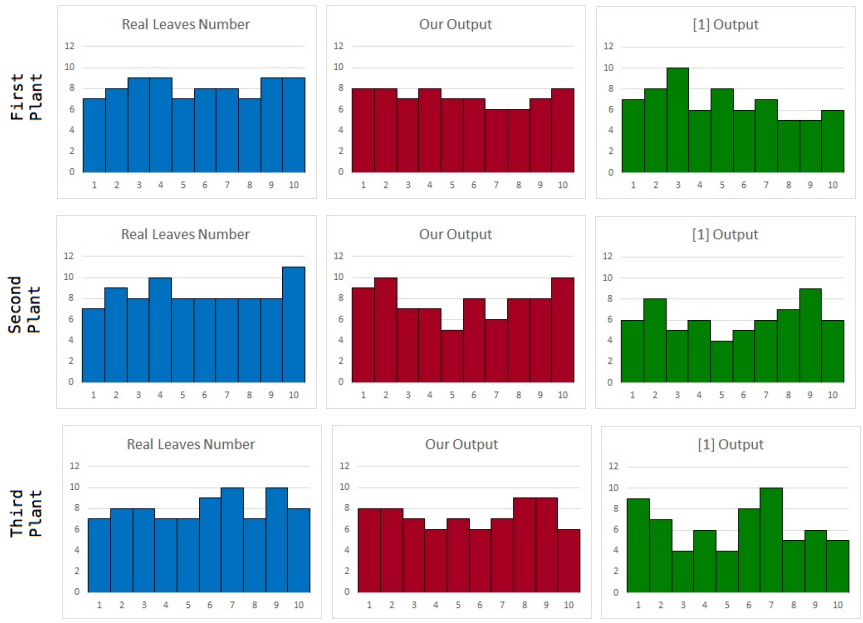


Figure 5.12: 32 DAS, Non-Infested

Figure 5.13: 22 DAS, Infested

Figure 5.14: 27 DAS, Infested

Figure 5.15: 29 DAS, Infested

Figure 5.16: 32 DAS, Infested

The global absolute error for my algorithm was of 0.85 leaf with an overestimation in the first two periods and an underestimation in the seconds two periods. The paper algorithm has a similar behaviour, overestimating for 22 DAS plants and underestimating for the remaining DAS, but it scores global absolute error 0.98. This data in combination with a calculated RMSE of 1.24 for my algorithm and 1.5 for the paper one, indicates that, statistically, the developed method is more accurate compared to the paper algorithm. In addition, I proceeded to calculate mean and standard deviation, for both the algorithm, on with the following results:

- Our algorithm:
  - Measurement:
    - Mean: 5.63
    - SD: 1.43
  - Residual:
    - Mean: 0.19
    - SD: 1.23
- Paper algorithm:
  - Measurement:
    - Mean: 5.01
    - SD: 1.74
  - Residual:
    - Mean: -0.43
    - SD: 1.43

Our data allows us to calculate even the skewness, respectively of 0.44 and 0.26 for measurement and residual of our algorithm, and 1.41 and 0.44 for measurement and residual of the paper algorithm. All this data indicate that both the algorithms tent to an overestimation but the developed algorithm performs better than the paper algorithm.

Chapter 6

Conclusions

6.1 General Conclusion

An innovative method of leaf detection has been presented. Comparing the results obtained with a state-of-the-art algorithm over the same dataset we demonstrated a significant improvement over the leaf detecting and counting. The most relevant improvement is on overlapping leaf, where we can take advantage of the shape of the leaves to differentiate them. Of course this can even became a drawback since if an image is taken with some leaf in a particular position (i.e. rotated of 90° on the central axis) this could lead to under or over-counting. It is worth mentioning that this entire solution was developed under semi-field conditions and probably why it out performs his competitor. The compared method was indeed developed under lab or green house conditions, so it was not studied to deal with multiple close plants. As final remarks it is right to specify that the initial goal has been archived since, with the developed algorithm, we are always able to distinguish between infested and non infested plants, even if the count of the leaves is not always precise. This project will allow a significant boost in the analysis of the plant yield drastically reducing the time needed to study the canopy area.

6.2 Future Works

In the future the midrib technique should be improved and implemented. Combined with a good quality acquisition process, it could lead to great results. Anyway a way to deal with completely hidden leaves should be studied. Maybe obtaining a 3D model of the plant could be a good way to fully analyse a single plant, but I see no way on how apply some sort of technique in semi-field or open field conditions.

Chapter 7

Acknowledgment

I would like to thank prof. Alessandro Bevilacqua for constantly supporting my work. I would also like to thank all the other members of the Syngenta group: Joalland Samuel, Screpanti Claudio, Gaume Alain, Plec Tobias, Castelli Laure, Reber Beat. They were a constant source of support throughout this process. Also thanks to my university, the University of Bologna, Cesena campus, that has allowed me to have this amazing experience. Finally, I am especially thankful of the constant support of my parents, Roberto and Oriella, my girlfriend, Marianna, and also of all those friends that helped me during the whole experience in a foreign country. Their patience, encouragement and closeness throughout this project has been amazing.

Bibliography

[1] Pape J M, Klukas C: *3-D histogram-based segmentation and leaf detection for rosette plants* (2015);

[2] Computer Vision Problems in Plant Phenotyping (CVPPP): *http://www.plant-phenotyping.org/CVPPP2014*

[3] Source: *https://en.wikipedia.org/wiki/Syngenta*

[4] Luc M, Sikora RA, Bridge J (2005) *Plant parasitic nematodes in subtropical and tropical agriculture.* CABI Bioscience, Egham

[5] Samuel Joalland (2015) *Belowground biomass accumulation assessed by digital image based leaf area detection.*

[6] Seinhorst JW (1965) *The relation between nematode density and damage to plants.* Nematologica 11:137–154.

[7] Cooke DA (1987) *Beet cyst nematode (heterodera schachtii Schmidt) and its control on sugar beet.* Agricultural Zoology Reviews 2:135–183

[8] Schmitz A, Tartachnyk II, Kiewnick S, Sikora RA, Kühbauch W (2006) *Detection of heterodera schachtii infestation in sugar beet by means of laser-induced and pulse amplitude modulated chlorophyll fluorescence.* Nematology 8:273–286

[9] Heath WL, Haydock PPJ, Wilcox A, Evans K (2000) *The potential use of spectral reflectance from the potato crop for remote sensing of infection by potato cyst nematodes.* Asp Appl Biol 60:185–188

[10] Laudien R (2005) *Entwicklung eines GIS-gestützten schlagbezogenen Führungsinformationssystems für die Zuckerwirtschaft. (Development of a field and GIS-based management information system for the sugar beet industry).* PhD thesis University of Hohenheim. Germany

[11] Nutter FW, Tylka GL, Guan J, Moreira AJD, Marett CC, Rosburg TR, et al (2002) *Use of remote sensing to detect soybean cyst nematode induced plant stress.* J Nematol 34:222–231

[12] Hillnhütter C, Mahlein AK, Sikora RA, Oerke EC (2012) *Use of imaging spectroscopy to discriminate symptoms caused by heterodera schachtii and Rhizoctonia solani on sugar beet.* Precis Agric 13:17–32.

[13] Bajwa SG, Mishra AR, Norman RJ (2010) *Canopy reflectance response to plant nitrogen accumulation in rice.* Precis Agric 11:488–506.

[14] Mahlein AK, Steiner U, Dehne HW, Oerke EC (2010) *Spectral signatures of sugar beet leaves for the detection and differentiation of diseases.* Precis Agric 11:413–431

[15] Yang C, Everitt JH (2002) *Relationships between yield monitor data and airborne multidate digital imagery for grain sorghum.* Precis Agric 3:373–388.

[16] Li L, Zhang Q, Huang D (2014) *A review of imaging techniques for plant phenotyping.* Sensors 14:20078–20111

[17] Milford GFJ, Pocock TO, Riley J (1985) *An analysis of leaf growth in sugar beet. II. Leaf appearance in field crops.* Ann Appl Biol 106:173–185.

[18] ImageJ source: *http://imagej.nih.gov/ij/*

[19] Woebbecke DM, Meyer GE, Von Bargen K, Mortensen DA (1995) *Color indices for weed identification under various soil, residue and lighting conditions.* Transactions of the ASAE 38:259–269

[20] Apelt F, Breuer D, Nikoloski Z, Stitt M, Kragler F (2015) *Phytotyping^{4D} : a light-field imaging system for non-invasive and accurate monitoring of spatio-temporal plant growth.* The Plant Journal 82, 693–706

[21] Pape J M, Klukas C (2015) *Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images*

[22] Capel D P(2001) *Super-resolution and Image Mosaicing*

[23] Lilian Zhang and Reinhard Koch (2013). *An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency.* Journal of Visual Communication and Image Representation, 24(7):794–805.

[24] Von Gioi, R. Grompone, et al. (2010) *LSD: A fast line segment detector with a false detection control*, IEEE Transactions on Pattern Analysis and Machine Intelligence 32.4: 722–732.

[25] T.Y. Zhang and C.Y. Suen (1984) *A fast parallel algorithm for thinning digital patterns*, Robert M. Haralick Editor.

[26] Zicheng Guo and Richard Hall (1989) *Parallel thinning with two subiteration algorithms*, Robert M. Haralick Editor.

[27] David G. Lowe (2004) *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision

[28] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool (2008) *Speeded-Up Robust Features (SURF)*

[29] Navneet Dalal and Bill Triggs (2005) *Histograms of Oriented Gradients for Human Detection* International Conference on Computer Vision & Pattern Recognition (CVPR '05)

[30] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich *feature hierarchies for accurate object detection and semantic segmentation.* CVPR 2014. Available online: http://www.cs.berkeley.edu/rbg/papers/r-cnn-cvpr.pdf (accessed on 20 July 2015).

[31] Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L., 2009. *Robust tracking-by-detection using a detector confidence particle filter.* In: IEEE 12th International Conference on Computer Vision. pp. 1515-1522.

[32] Y. Song, C.A. Glasbey , G.W. Horgan, G. Polder, J.A. Dieleman, G.W.A.M. van der Heijden (2013)*Automatic fruit recognition and counting from multiple images.*

[33] Yun-Heh Chen-Burger , Gayathri Nadarajan, Robert B. Fisher *Detecting, tracking and counting fish in low quality unconstrained underwater videos*

[34] S. Beucher, (1992) *The watershed transformation applied to image segmentation*, Scanning Microsc. 6 (Suppl.) (1992) 299-314.

[35] N. Salman, (2006) *Image segmentation based on watershed and edge detection techniques*, Int. Arab J. Inf. Technol. 3 (2).

[36] N. Jamil, H.C. Soh, T.M.T. Sembok, Z.A. Bakar, (2011) *A modified edge-based region growing segmentation of geometric objects, visual informatics: sustaining research and innovations*, Lect. Notes Comput. Sci. 7066 99-112.

[37] J.Z. Chiwoo Park, J.X. Huang, Yu Ding Ji, (2013) *Segmentation, Inference and classification of partially overlapping nanoparticles*, IEEE Trans. Pattern Anal. and Mach. Intell. 35 (3) 1, http://dx.doi.org/10.1109/TPAMI.2012.163.

[38] P. Bamford, B. Lovell, (1998) *Unsupervised cell nucleus segmentation with active contours, Signal Process.* 71 (2) 203-213.

[39] Y. Zhang, B. Matuszewski, (2009) *Multiphase active contour segmentation constrained by evolving medial axes*, Proceedings of the 16th IEEE International Conference on Image Processing. Cairo, Egypt, pp. 2993-2996.

[40] M. Marcuzzo, P. Quelhas, A. Mendonca, A. Campilho, (2009) *Evaluation of symmetry enhanced sliding band filter for plant cell nuclei detection in low contrast noisy fluorescent images*, Image Anal. Recognit. 824-831.

[41] P. Quelhas, M. Marcuzzo, A. Mendonca, A. Campilho, (2010) *Cell nuclei and cytoplasm joint segmentation using the sliding band filter*, IEEE Trans. Med. Imaging 29 (8) 1463-1473.

[42] G. Lin, U. Adiga, K. Olson, J.F. Guzowski, C.A. Barnes, B. Roysam, (2003) *A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks*, Cytometry Part A 56A 23-36.

[43] P.S. Umesh Adiga, B.B. Chaudhuri, (2001) *An efficient method based on watershed and rule-based merging for segmentation of 3-D histopathological images*, Patt. Recognit. 34 1449-1458.

[44] S. Beucher, (1994) *Watershed: hierarchical segmentation and waterfall algorithm*, in: J. Serra, P. Soille (Eds.), Mathematical Morphology and Its Applications to Image Processing, Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 69-76.

[45] C. Wählby, I.-M. Sintorn, F. Erlandsson, G. Borgefors, E. Bengtsson, (2004) *Combining intensity, edge, and shape information for 2D and 3D segmentation of cell nuclei in tissue sections*, J. Microsc. 215 67-76.

[46] K.Z. Mao, P. Zhao, P.H. Tan, (2006) *Supervised learning-based cell image segmentation for P53 immunohistochemistry*, IEEE Trans. Viomed. Eng. 53 1153-1163.

[47] F. Meyer, Levelings, (2004) *Image simplification filters for segmentation*, J. Math. Image Vis. 20 59-72.

[48] Plant Image Analysis: *http://www.plant-image-analysis.org/*