Agricultural Crop Monitoring with Computer Vision

James Ian Burns

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Al L. Wicks, Chair
John P. Bird
Craig A. Woolsey

September 11, 2014
Blacksburg, Virginia

Keywords: Computer Vision, Machine Learning, Agriculture, Multispectral, Automation

Agricultural Crop Monitoring with Computer Vision

James I. Burns

(ABSTRACT)

Precision agriculture allows farmers to efficiently use their resources with site-specific applications. The current work looks to computer vision for the data collection method necessary for such a smart field, including cameras sensitive to visual (430-650 nm), near infrared (NIR,750-900 nm), shortwave infrared (SWIR,950-1700 nm), and longwave infrared (LWIR,7500-16000 nm) light. Three areas are considered in the study: image segmentation, multispectral image registration, and the feature tracking of a stressed plant.

The accuracy of several image segmentation methods are compared. Basic thresholding on pixel intensities and vegetation indices result in accuracies below 75% . Neural networks (NNs) and support vector machines (SVMs) label correctly at 89% and 79%, respectively, when given only visual information, and final accuracies of 97% when the near infrared is added.

The point matching methods of Scale Invariant Feature Transform (SIFT) and Edge Orient Histogram (EOH) are compared for accuracy. EOH improves the matching accuracy, but ultimately not enough for the current work.

In order to track the image features of a stressed plant, a set of basil and catmint seedlings are grown and placed under drought and hypoxia conditions. Trends are shown in the average pixel values over the lives of the plants and with the vegetation indices, especially that of Marchant and NIR. Lastly, trends are seen in the image textures of the plants through use of textons.

Acknowledgments

I would like to thank those who have guided me throughout my graduate career. Namely, my committee including Dr. Al Wicks, Dr. John Bird, and Dr. Craig Woolsey for their advice and leadership, Dr. Jerzy Novak and his graduate student John Weekly for their help in constructing experiments, Jeffery Burr for providing the greenhouse space and plants for the mentioned experiments, and, of course, my family and friends for putting up with me.

Contents

List of Figures

List of Tables

List of Abbreviations

Acronym	Definition
2D	Two Dimensional
3D	Three Dimensional
CCD	Charge Coupled Device
CV	Computer Vision
ExG	Excess Green Index
F	Marchant Index
FOV	Field of View
HSL	Hue-Saturation-Lightness
HSI	Hue-Saturation-Intensity
HSV	Hue-Saturation-Value
IR	Infrared
LWIR	Long Wave Infrared
ML	Machine Learning
NDI	Normalized Difference Index
NDVI	Normalized Difference Vegetation Index
NIR	Near Infrared
NN	Neural Network
RGB	Red-Green-Blue (Visual)
SDK	Software Developer Kit
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
SWIR	Shortwave Infrared
VIS	Visible (RGB)

Chapter 1

Introduction

Precision agriculture is the use of information technologies in agriculture [1]. It allows farmers to efficiently use their resources by applying site-specific fertilizers, pesticides, and watering [2]. A smart field that could autonomously collect health and growth data of the crops and then react appropriately, could help farmers remain competitive and keep up with the increasing food demand [3]. Computer vision has been used in this area, mainly with airborne imagery, fruit harvesting, and post-harvest inspection, e.g. [4–6], but also with crop health inspection with visual (430-700 nm), near infrared (700-900 nm), and hyperspectral (within 385-1000 nm) data. This paper aims to explore the potential for computer vision in the agricultural setting by experimenting with plant image segmentation and registration, and with tracking image features over the life of a stressed plant.

1.0 Chapter Outline

The current work begins with an argument for the use of computer vision in agriculture in Section 1.1, and its current use and employment throughout the past few decades is reviewed in Section 1.2. From this, a problem statement is defined for the paper in Section 1.3. Lastly, an outline of the current work is presented in Section 1.4.

1.1 Motivation

As technology increases, farmers have been producing more crop per capita. In other words, fewer farmers have been making more food, and to keep this trend they will need to embrace new technologies. Currently, farmers visually inspect individual plants and treat an entire field (with pesticide, irrigation timing, etc.) based on this small sample [7]. In order to use treatment resources efficiently, sensors are needed in the field to collect quantified data about

stress or growth distribution that will allow farmers to use location specific treatments [7].

Over the past couple decades, this level of accuracy in agriculture has become known as Precision Agriculture, or Site Specific Crop Management. Many sensors can be used to achieve precision agriculture, but vision has many advantages including being nondestructive, inexpensive, and requiring little infrastructure. Additionally, more farming equipment is now being constructed with vision sensors for robotic navigation already [8].

Precision Agriculture

One area where precision agriculture has many opportunities is in stress control, such as for diseases and water stresses [9]. Location specific application of pesticides, irrigation, and anti-fungal treatments would reduce the cost of materials for the farmer as well as the chemical runoff into the nearby environment. More importantly, having a quantified and precise description of the stresses in fields would allow for quick responses, which has a large effect on yield [10].

For example, a common method of water stress monitoring is inter-node length measurement done for cotton [7]. Tests such as this one cannot be conducted on a frequent enough basis. The severity of short and long term damage is directly related the treatment response time, and, at least with water related stress, there is often damage before the human eye can see it [11]. Creating an automated system would give farmers the ability to react appropriately.

Similarly, precision agriculture can be used for invasive species control. The overuse of herbicide can lead to chemical waste, crop damage, and environmental pollution [12]. Some weeding is even accomplished manually, requiring a lot of labor hours [13], and conventionally the measuring of weed invasion is done by visual inspection and is expensive and time consuming [14]. The precise spatial distribution of invasive species in a field can help tremendously when doing weed control, and the level of accuracy needed cannot be obtained through conventional means [14].

Early stress detection is a challenge also in the greenhouse setting (an ideal environment for plants, but for pests and diseases as well). Typically plants are inspected by an expert on a weekly basis [15]. Clearly, this process should be automated, especially given the level of autonomy greenhouses already have, such as controlled temperature, humidity, lighting, irrigation, watering, and even carbon dioxide levels [15, 16].

There are many other applications related to precision agriculture that will not be studied in this paper. The sheer gathering of information about a field can be used to predict yield versus field attributes such as plant spacing [17–19]. The quantization of plant data can be used in plant phenomics, the study of how plant genes and their environments affect the physical and biochemical traits of the organisms [20]. The precision control of water level of Sunagoke moss has even been used to mitigate the urban heat island phenomenon [21].

Precision agriculture can be used not only for battling plant stresses, but for the application of nutrients at the right time as well. The specific delivery of nutrients to plants at different growth stages lowers the input resources needed to increase the yield [3]. For ideal crop cultivation, it is important to know the growth information for each growing stage of the plant [22]. It allows for the specific application of fertilizers and other chemicals, reducing the cost of surplus material, and saving of the environment [22]. More specifically, overfertilization is one of the leading causes of fertilizer runoff into streams and rivers, which can create an excess amount of algae resulting in a depletion of oxygen that kills the other wildlife. Overfertilization can be the result of farmers or gardeners not knowing the nutrient content of the fertilizer or incorrectly diagnosing bad plant health as nutrient deficiency.

These technologies can also be applied to the surveying of large areas such as national forests. One example of this is the Virginia Department of Agriculture and Consumer Services Slow the Spread program where they are trying to slow the spread of Gypsy moths in the Blue Ridge mountains. They monitor the situation by placing traps all over the mountains. This task could be accomplished with a group of drones with vision sensors. Similarly, another example is the surveying tree diameters and types within a forest for a paper mill; a process that is done manually and can often result in the overbidding of tens of thousands of dollars.

Agriculture Machine Vision

As mentioned earlier, there are many ways to achieve precision agriculture, but machine vision has many advantages. It is easily automated when implemented with a drone as it is often used for robotic navigation, it allows for instant, real-time data with no damage to the crops [22], and it is also the most versatile of sensors allowing for the measurement of plant height, leaf area, and chlorophyll-like response. Additionally, it creates a quantitative measurement of stresses within an individual plant that would not be attainable through conventional means [23], which can include judging the brownness of leaves, their curl, or the dryness of the soil.

In terms of the robotic navigation mentioned above, machine vision sensors are already largely in use for autonomous harvesting. Studies with vision in agriculture have already been done in fruit visibility [24–27], citrus harvesting [28, 29], mushroom identification [30], and row guidance [31]. This proves that cameras are already making an appearance in agriculture, and that their versatility should be put to use.

More specific to this paper, multispectral computer vision also has many advantages. Plants typically absorb a large portion of visible light and reflect a large amount of the near infrared spectrum [11]. In this paper we will see this is also true of the shortwave infrared band. The use of cameras, even multispectral cameras, can often be a far less expensive alternative to other sensors, such as radiometers [15].

Throughout this paper, vision will be examined as solution to obtaining the measurements

necessary for precision agriculture. Namely, it will be determined whether they provide the information necessary for plant stress classification. It is an avenue worth exploring due to their precision, versatility, and their current use in agriculture.

1.2 History

In order to gain perspective of what is accomplished in this paper, a brief history of the subject material is presented. Given that this paper is a study of computer vision in agriculture, an overview of the history of computer vision in general is stated for comparison in how quickly the technology came to agriculture.

Computer Vision

The first digital image was created in 1957 by Russell Kirsch, and it was a 176 x 176 pixel scan of a photograph of his three-month-old son. It was not long after that two dimensional pattern recognition and optical flow was developed by Gibson.

In the 1960s, Roberts, a MIT PhD candidate wrote his thesis on creating 3D models from two dimensional views, and in 1966 professor Minsky assigned computer vision object recognition, a still unsolved problem today, to an undergrad for a summer project. Over the next two decades, the 1970s and 1980s, a more mathematical definition of computer vision is formed and researchers create methods such as edge detection, image segmentation, and optical character recognition; also, the first digital camera is invented in 1975.

In the 1990s, computer vision takes off as in industry and is introduced in areas such as manufacturing. Algorithms such as face recognition are developed, and in the 2000s, computers have become much more powerful allowing for large annotated datasets, video processing, and object recognition. This also resulted in it becoming closely tied to machine learning.

Computer Vision in Agriculture

The idea to use computer vision in agriculture first appeared in a 1968 paper by Schertz and Brown [32] where they proposed a robotic citrus harvester that used vision to find the fruit, and it was implemented in 1977 by Parrish and Goksel [33]. Computer vision guided robotic harvesting did not gain real traction, however, until the late 1980s with papers involving fruit visibility [26], citrus harvesting [29], and other crop visibility [30]. In the 2000s, the area is still being explored in the same areas: fruit visibility [24,25,27] and citrus harvesting [28], as well as row guidance [8,31]. Most of the earlier work of computer vision within agriculture was done with fruit detection which is explained to be due to simplicity in the difference in color of the fruit and the foliage [30].

The focus of this paper, however, is of plant monitoring and stress classification. An important subsection of this area is plant segmentation and research into this separating of plants from their backgrounds started in the 1980s. Interest in this area was sparked by a hand-held prototype device that sprayed herbicide and was activated by spectral differences in crop and weed canopies measured by an optical sensor [34].

The importance of the near infrared and visual ratios was realized by Guyer et al. in 1986 who then used a camera and found the plant pixels to have higher ratio intensities than that of soil [35]. Similarly, Meyer and Davison discovered that the red plane of a Red-Green-Blue (RGB) image could be subtracted from the green or blue plane to result in brighter leaf pixels as compared to soil [36]. Additionally, in 1988 a near infrared camera and image processing were used to evaluate the leaf area of lettuce in a greenhouse [37]. Basic thresholding techniques were used to find the leaf pixels and then this information led to the crude calculation of the plant's biomass.

In the late 1990s, plants were being segmented from their backgrounds using the excess green color index by Meyer, the normalized difference index by Perez, and even a Bayes classifier by Tian and Slaughter.

During the 2000s, experiments with computer vision in agriculture started to become more common. In controlled, mostly greenhouse environments and with visual band information only, disease detection was done using color and texture features [9, 10, 23], automatic corn counting and spacing was implemented [19], pests on leaves were detected [15], and gravitropism was studied [38]. Additionally, satellite, hyperspectral imagery was used to find invasive plant species.

Since 2010 the study has grown with experiments in 3D modeling, leaf identification, image segmentation, and stress detection all in computer vision. The 3D modeling was accomplished with trinocular vision as well as hyperspectral vision [20, 39]. Studies on images of leaves have been investigated to find plant species and disease classification [40, 41]. The effect of water stress has even been observed with computer vision on hyperspectral data of apple trees as well as visual data of sunagoke moss [11, 21].

The current work looks to add to this list through studies of image segmentation, stress detection, and multispectral vision in the agricultural setting.

1.3 Problem Statement

This thesis is an investigation of the possible use of computer vision in the agriculture setting and contains three main thrusts: (1) segmentation of the plant, (2) multispectral image registration, and (3) plant health classification. As mentioned by [39], plants are non-rigid biological objects that contain complex structures resulting in a non-ideal computer vision problem, so the current work aims to test a variety of methods for their effectiveness

in the area.

Within each of these areas, an investigation is held into which features describing the images are most useful. These features can include the multispectral information given by our cameras, vegetation indices created by previous authors, and textural information produced by image processing techniques. Feature Selection, as defined by [21], is the choosing of a feature subset that maximizes the prediction accuracy and allows the removal of irrelevant and redundant features. Learning which features are important can save on cost in additional cameras (especially with the multispectral) and the computational cost of the computer vision descriptors.

1.3.1 Segmenting the Plant

Before the other two sections of this problem statement can be accomplished, the plants must first be found within the images. This means that each pixel in an image needs to be identified and labeled as plant or non-plant. Typically, each pixel is described by a set of features for comparison, and these could include pixel intensities such as RGB or Hue-Saturation-Value (HSV) values, intensity derivatives, filter responses, or a combination of any of the above. A threshold is then applied to any of these features or to the output of a machine learning algorithm used on these features. If the pixel's feature or algorithm result is within a certain range it is a plant. If it is not, it is not a plant.

Segmenting a plant's pixels from the background is still an unsolved problem because of several obstacles. Plants are usually in natural scenes where objects and their backgrounds often contain common intensities causing failure of basic thresholding techniques [7]. Additionally, plants are usually in uncontrolled environments where the weather can have a large effect on resulting images. [42] found this to be particularly true when it comes to using color features.

As described in later chapters, a wide variety of features will be studied. Commonly used features such as RGB and HSV values will be tested. Index features described by previous authors in the plant segmentation domain will also be put to the test, such as the Normalized Difference Index (NDI), Excess Green (ExG), Normalized Difference Vegetation Index (NDVI), and the Marchant index.

This paper will aim toward using machine learning to interpret the feature space and segment the image. Two types of methods exist for this, supervised and unsupervised [41]. Supervised learning requires the hand labeling of images as training examples and, for this paper, algorithms including neural networks and support vector machines will be investigated. Unsupervised techniques, such as K-means clustering as used in this paper, automatically looks at the feature space and splits it into common groups.

The ultimate goal of plant segmentation is an image mask that will label each pixel in an image as plant or not plant. This will allow the pixels representing the plant to be used in

further studies, as mentioned in the next two subsections, without the interference of the background pixels.

1.3.2 Multispectral Image Registration

A wealth of information can be obtained through the use of these multispectral cameras, however, to use this the images must be registered. In their raw form, the images have different resolutions, field of views, and physical positions and rotations in space. This can be seen in Figure 4.6 by the size of each image and what is contained within each one, e.g. the LWIR camera can see out the door on the left, and the monitor in the extreme foreground causes occlusion in the scene. Even if the cameras had all the same parameters, because of the position translations, objects in the scene will shift dependent on their depth.

It is an intention of this paper to create a system to automatically match points in each image and warp them accordingly so pixel values can be compared. This involves finding key points in an image that are significant as well as unique. Once these points are located in every image, they must be matched in between. This requires the description of the points in a way that makes any given point different from the others in the image while at the same time being similar enough to the proper point in the adjoining image. After enough points are matched, one image can be warped to another, allowing for the comparison of the spectral data.

1.3.3 Plant Health Classification

An experiment is designed and implemented to use computer vision to classify the stress state of a plant within an image. A group of plants are grown in a greenhouse and placed under various stress conditions, namely under and over watered. Images of these plants are taken on a daily basis to produce a multispectral dataset that computer vision algorithms can be tested on.

Given the mass of information contained in the collection of images for each plant, the goal of the current work is to find what within this collection can be used to determine the stress state of the plant. Features describing the image will be varied to find which spectrum and which computed features are important. Namely, pixel intensities are tracked over the days as well as several vegetation indices. Additionally, textural information is observed through textons.

The texton method is used to automatically learn the image textures of a class within an image. This method involves the use of k-means clustering on the responses of a texture filter set for each class to obtain a texton dictionary of centers describing the class. This texton dictionary is then used to describe and compare new samples. For more information see Section 3.2.5.

1.4 This Document

This paper is generally structured in such a way that background information is given in Chapters 1: Introduction, 2: Literature Review, and 3: Related Theory, the plan for the current work is detailed in Chapter 4, and the results are presented and discussed in Chapters 5: Results and 6: Conclusions.

Introduction

The paper begins with the reasoning behind the current work. In Section 1.1 the field of agriculture is looked at from the perspective of growing technologies in robotics and computer vision. Farming is becoming more automated and the role of vision as a key sensor is considered as a definite. In the following section, Section 1.2, a brief history of the areas of computer vision, machine learning, and vision in agriculture is reviewed to reveal that vision is only becoming more relevant in the world and specifically in agriculture.

The goals of the current work are then outlined in Section 1.3 and it is considered how they fit within the work done by peers. The three different branches of this project are discussed with subsections for image segmentation, image registration, and stress classification.

Literature Review

Chapter 2 contains the important work of reviewing what others have accomplished in the field of the current work. Many authors have attempted to segment plants from their backgrounds in images and they are observed in Section 2.1. Their methods range from simple thresholds to neural networks and the resulting accuracies and example segmentations are shown for comparison to the current work.

Some authors have also attempted to monitor the stress level of a plant and these are detailed in Section 2.2. Mostly work has been done on finding pests and disease on images of leaves and some work has been done on water stress of an entire plant. In both sections within Chapter 2 the authors using multispectral vision are reviewed.

Related Theory

The current work uses many machine learning and computer vision algorithms. Section 3.1 reviews the three different machine learning methods used, SVMs, NNs, and K-means clustering, as well as the metric used for accuracy comparison of the algorithms.

The second section, Section 3.2, builds computer vision from the ground up starting with the definition of a camera and details the computer vision methods used in this study.

Experimental Design

The chapter of Experimental Design steps through the plans for the current work with a section for each of the three thrusts: image segmentation, image registration, and stress classification. It first discusses the multispectral camera rig used in this paper in Section 4.1. Namely, each of the cameras are described in details including resolution, field of view, and spectral sensitivity.

Section 4.2 reviews the features, algorithms, and testing methods that will be used for segmentation. The features include pixel values as well as indices employed by various authors. These features are used in segmentation methods involving simple thresholding, neural networks, and support vector machines. Lastly, the plan on how the effectiveness of these features and algorithms will be tested and compared is detailed.

Next, Section 4.3 compares several point finding and describing methods as well as image warping schemes for use in multispectral vision. Most significantly, a plan to compare EOH descriptors against the more common SIFT and ORB is described, as well as the methods used to obtain a ground truth.

Lastly, Section 4.4 details how a group of plants are placed under various stresses and images of which are used to determine the state of stress. The physical setup of the plants, their watering schemes, and the locations of the cameras are detailed. More importantly, the computer vision methods used to find the telling features of the images are described.

Results and Analysis

Chapter 5 details and reviews the results of the experiments previously explained. For image segmentation, Section 5.1 compares the method types, specifically thresholding and machine learning, and presents the accuracies of the methods within. An analysis of the resulting failure modes is shown for a few choice examples.

Section 5.2 presents the results of the multispectral image registration experiments. That is to say, the feature finding and description algorithms are compared for m atching, and the image warping methods are visually inspected.

Last of all, Section 5.3 reviews the resulting data from the plants that were tracked over the experiment. Trends are shown in the simple features such as pixel intensities and vegetation indices, as well as in the texton analysis.

Conclusions

Finally, Chapter 6 reviews the current work and its possible impact in the field of computer vision in agriculture. Additionally, Section 6.2 considers improvements to the experiments

as well as the directions future work could go in.

Chapter 2

Literature Review

This document is looking into how computer vision could be used to monitor plants in an agricultural or horticultural setting. Before diving into data collection and analysis it is first necessary to research into what others have done. This chapter outlines and details the work of other computer vision scientists that have investigated the imaging of plants. This review will supply the current work with ideas for feature descriptors, filters, and algorithms to employ.

2.0 Chapter Outline

As shown in Section 1.2, researchers have been using computer vision in agriculture since the late 1970s and it has recently become more popular. Within this chapter, the work done by these vision scientists are divided into the interest areas of the current work, specifically plant image segmentation and stress detection.

The work in image segmentation in the agricultural setting is detailed in Section 2.1. It begins by reviewing authors that have done simple segmentation of leaves with plain backgrounds and moves to full plants surrounded by soil. Lastly, the section considers authors that have used multispectral or hyperspectral imaging to find plants within an image.

Section 2.2 looks at the papers of authors attempting to classify a stressed plant. It again begins with work containing images of only leaves with clear disease stresses. It continues then into a similar study with images filled with sunagoke moss in an attempt to classify its water content. A hyperspectral study of the water stress of an apple tree is lastly reviewed.

Additionally, related papers are referenced in Section 2.3 to present an idea of other work in the field.

2.1 Segmenting the Plant

Nearly all of computer vision in agriculture requires the segmentation of the plant within the image. Image segmentation is defined as the labeling of each pixel of an image into a class - in this case, plant or not plant. The output of a segmented image is usually a binary image where white pixels represent pixels that have been classified as plant.

Early on, researchers such as [36] and [35] used basic thresholding techniques to distinguish plant from soil. Either requiring the plant be brighter than the soil in a grayscale image, or that it have more green than red. This was enough for their purpose of roughly measuring biomass from overhead, but as the problem became more complex different tactics were needed.

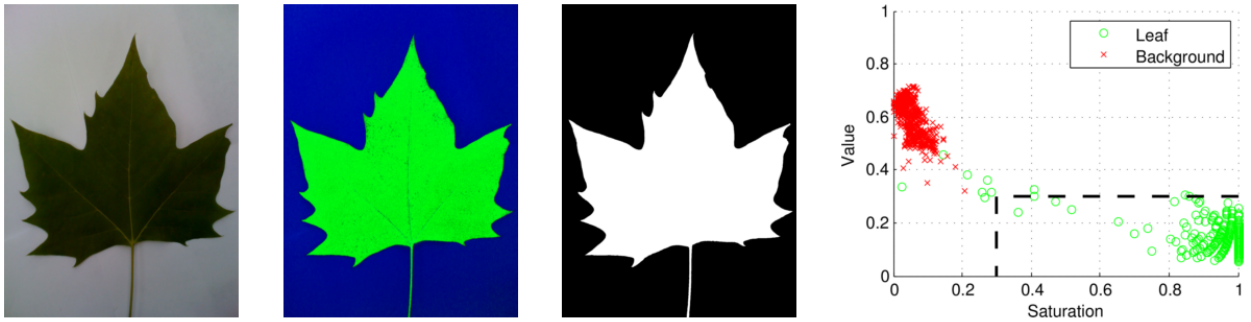
Many of the methods used by the following authors are tested for the current work, including simple thresholds, vegetation indices, and machine learning algorithms.

2.1.1 Leaf Analysis

Some of the simpler examples of segmentation take place on the analysis of leaves. [43], and later [40], classified plant species from an image of a single leaf. The image was required to be largely the leaf on a bright, untextured background for ease of segmentation. They used Expectation-Maximization to classify each pixel as plant or non-plant in the saturation space, and found that saturation was the most useful of the HSV space and that the hue of the background was often too green due to reflections in the scene. As a post-processing step, they conducted connected components on a dilated version of the image. This mitigated any problems with floating blobs and pixels that were not part of the leaf.

A resulting example is shown in Figure 2.1 with (a) an original leaf image, (b) the image in the HSV space, (c) the resulting binary image, and (d) a plot of the hue and saturation of the image with segmenting boundaries shown.

Figure 2.1: Leaf Segmentation Example From Literature: Neeraj Kumar, PN Belhumeur, and Arijit Biswas. Leafsnap: A computer vision system for automatic plant species identification. *Computer Vision-ECCV*, pages 1-14,2012. [Under Fair Use, 2014]



(a) (b) (c) (d)

Similar to [43] and [40], the next two papers referenced are on the analysis of a single leaf. [41] segmented a leaf image to find diseased section of the leaf, and [15] did the same to find pests on the leaf. [41] used k-means clustering on textures in the CIE L*a*b* color space to find the different sections of the image. [15] used a self-designed cognitive classification system to determine if a pixel was background, leaf, or pest.

2.1.2 Plants with Simple Backgrounds

A series of experiments were conducted by [44] with plant image segmentation using a variety of index thresholds and neural networks. These indices, collected from various authors, are used in the current work and are explained here. The first index is the Excess Green Index (ExG) and can be calculated as in Equation 2.1, where R,G, and B are the pixel's red, green, and blue values, respectively. This index was created by [45] and emphasizes pixels that are largely green and are not another color, even colors with green in it, such as white. [44] notes that this index works well, but often fails on shiny leaves where the green becomes saturated and his resulting segmentation can be seen in Figure 2.2a.

$$ExG = 2 \times G - R - B \tag{2.1}$$

An index by [45] is also employed called the Normalized Difference Index (NDI) which is calculated as in Equation 2.2. This index emphasizes green as well, but recognizes that plants absorb more blue than they do red. It has the advantage over ExG by being normalized between -1 and +1, and the closer the value is to +1 the more likely it is a plant. A segmented image generated by [44] can be seen in Figure 2.2b. Readers will note the similarity to NDVI mentioned in the later Section 2.1.4. NDI is an alternative that does not require near infrared capabilities.

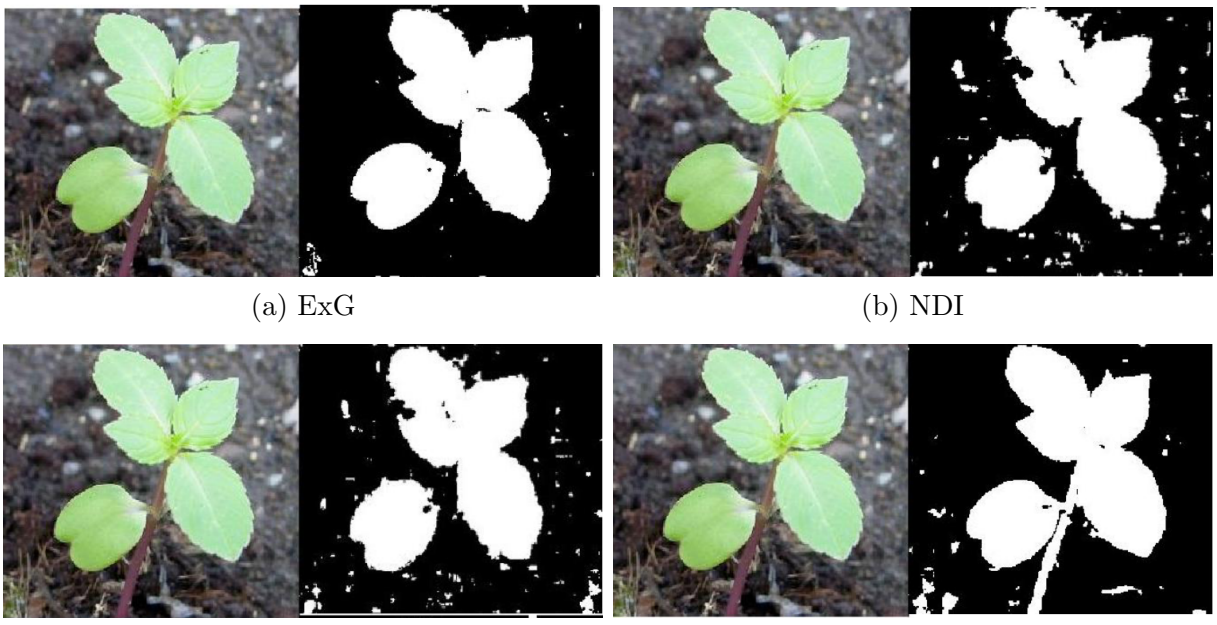
$$\text{NDI} = \frac{G - R}{G + R} \tag{2.2}$$

The final index [44] examines is the Marchant Index, F, as shown in Equation 2.3 with $r_m = R/B$ and $g_m = G/B$. The value A is a tunable parameter. This index normalizes both the red and green channels with the blue one and is lower with more green in a pixel. The result can be seen in Figure 2.2c.

$$F = \frac{r_m}{g_m^A} \tag{2.3}$$

The work done by [44] is the effectiveness of putting a threshold on these indices versus using a neural network to segment the image. For the NN, [44] uses the red, green, and blue values as input features and has one hidden layer with three nodes. The NN was trained with random pixels from several scenes containing weeds with a background of either dirt or brick. A resulting segmentation can be seen in Figure 2.2d.

Figure 2.2: Plant Segmented Images From Literature: Floris De Smedt, I Billiauws, and Toon Goedeme. Neural networks and low-cost optical filters for plant segmentation. *International Journal of Computer Information Systems and Industrial Management Applications,* 3:804-811, 2011. [Under Fair Use, 2014].



(a) ExG (b) NDI

(c) Marchant (d) NN

Ultimately, accuracies above 90% were obtained for each method and the highest of 98% belonging to the neural network. Additionally, many of the incorrectly classified pixels

in [44] were in small blobs in the background and could have been removed in automated post-processing. Further details are shown in Table 2.1 copied from [44].

Table 2.1: Segmentation Accuracies Reported by Literature.

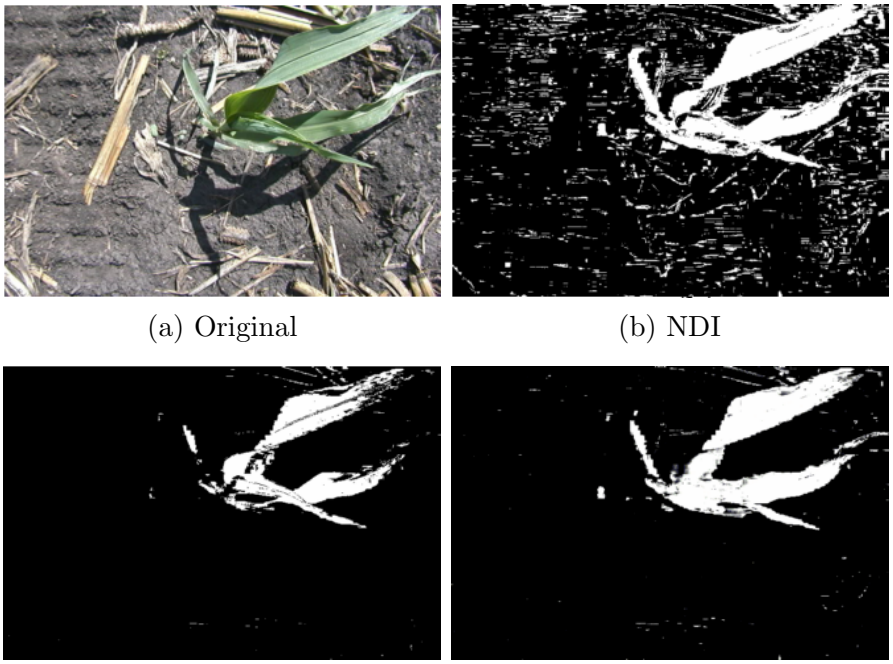
	Correct classified pixels (%)	**Plant pixels found (%)**	**Ground pixels found (%)**
Modified ExG	96.2210	90.9946	98.6880
NDI	92.4784	88.1333	94.5251
Marchant	93.7070	89.8840	95.5116
Neural network	97.2912	96.6627	97.587

A more complex approach created by [46] involves a truncated ellipsoidal surface given by Equation 2.4 with tunable parameters D, E, and F. While the statement is true and the left hand side of the equation is less than one, the pixel is labeled as that of a plant, and if it is greater than one, background. A 3-layer neural network was used to estimate the three parameters required. A sample segmentation can be seen in Figure 2.3d.

$$\frac{R^2}{D^2} + \frac{(1-G)^2}{(E \times B + F)^2} < 1 \tag{2.4}$$

Twenty images of young corn plants in natural scenes were hand labeled for testing of the new algorithm and it was compared against other standards. Thresholding constants were applied to the NDI response of the image ranging from 0.05 to 0.25. The author found that the lower the threshold, the more background pixels were incorrectly classified, and the higher the threshold, the more foreground pixels were incorrectly classified. An example is shown in Figure 2.3b.

Figure 2.3: Plant Segmentation Example From Literature: DS Shrestha, BL Steward, and E Bartlett. Segmentation of Plant from Background Using Neural Network Approach. *Proc. of the Artificial Neural Networks in Engineering Conference,* (4-7):903-908,2001. [Under Fair Use, 2014].



(a) Original (b) NDI

(c) Bayes (d) NN

A Bayes Classifier was also employed for comparison and similar results were found, shown in Figure 2.3c. Details of the average classification methods can be seen in Table 2.2, but the author ultimately found that dynamic decision surfaces with parameters that were altered on a per image basis significantly outperformed that of a static nature. The same author used this method later on for automatic corn plant population measurement [19].

Table 2.2: Segmentation Accuracies Reported by Literature.

	Correct classified pixels (%)
NDI	93.57
Bayes	98.22
TE	98.41

2.1.3 Plants in Natural Scenes

The previous papers have been in relatively controlled environments, with backgrounds that are bright and untextured or are dark soil; [42] designed an algorithm for an unstructured background. He notes that the related work will not perform well in realistic environments and fail with illumination changes, similar colored or textured backgrounds, or with an unhealthy plant.

Using a mix of Self-Organized Mapped (SOM) neural networks and Bayesian classifiers, [42] attempts to tackle the realistic problem. The new method seems to work well in typical cases, but does not drastically outperform other methods when dealing with the problems related to a natural scene, such as a dead spot on a leaf. The author claims this method cannot be quantitatively compared to related work.

In a related paper, [8] segments images of a field from the perspective of a ground vehicle. He notes that in the automatization of the farming industry optical sensors will play a large role with many uses. One such use is the navigation of any unmanned system on the field. The work proposes an approach where a broad texture analysis will break the image into larger sections and supervised fuzzy logic on finer, sub-textures will find more specific areas in the image.

Going in the opposite direction, [47] investigates ways to segment even the parts of the plant, attempting to distinguish overlapping leaves from one another. This method was created for younger plants and specifically the tobacco plant seedling. Ultimately, [47] found active shape models and ellipse detection to work well for all but extreme overlapping.

2.1.4 Multispectral and Hyperspectral

Since plants do not absorb most infrared light but do with visual, multispectral and hyperspectral vision are natural selections to make for image segmentation and stress analysis, and many authors have chosen to do so [11, 20, 38].

In 1998, [37] used an NIR camera setup to calculate the growth of several vegetable plants. A single Charge Coupled Device (CCD) camera was used to collect 2-bit grayscale images in the near infrared spectrum, specifically 750-1100 nm. From the resulting images, the plant's leaf area was calculated and compared against the plant's weight. Even though this experiment was basic by the standards of computer vision today, the author was able to correlate the two parameters and introduce the value of the NIR spectrum in agriculture.

Additional to the indices mentioned by [44] shown previously, the Normalized Difference Vegetation Index (NDVI) greatly improves plant image segmentation. As seen in Equation 2.5, NDVI uses the near infrared and red channels to score each pixel as plant or not plant. Since plants reflect most infrared and absorb most red, NDVI emphasizes the difference between the two.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{2.5}$$

Another author, [20], conducted an experiment in plant phenomics using hyperspectral imaging. To measure characteristics of the plant, such as biomass and surface traits, images were collected surrounding the subject plant with a range of spectrum. Namely, images were captured full circle at every three degrees and at every 10 nm wavelength interval from 400 nm to 1000 nm. The images were segmented and points were matched between adjacent frames in order to achieve the final goal of a 3D model of the plant.

During this study, [20] found the hyperspectral images to be noisy due to the narrow bandwidth of light being captured, and the authors even found the images correlating to 400 nm to 590 nm (all of the visual spectrum except lower red) to be unusable. This led to problems during the 3D modeling and the author found the SIFT features to be insufficient for the process. The authors augmented the keypoints with SIFT features found on the Canny edge result of each image.

For segmentation, the author tested k-means clustering with three clusters and an SVM classifier. While the author did not provide accuracies for each method, they reported the SVM to outperform the k-means clustering. Ultimately the author stated that hyperspectral imaging was useful for image segmentation (against a plain background) and the resulting 3D models for each spectral plane complement each other.

2.2 Stress Detection

Once the plant has been segmented from the image it can be analyzed for information such as species classification, biomass growth, and stress state. Stresses can include water starvation, hypoxia, diseases, or pests. A few studies dealing with these have been mentioned in the previous section: [41] segmented the dying parts of a leaf for disease identification, and [15] found whiteflies on rose leaves. Neither of these authors, however, went further into stress monitoring than segmentation. Other authors however, [10] and [9], have done the further stress analysis on leaves.

2.2.1 Leaf Analysis

The authors of [10] found leaves bearing diseases required the subjective, manual examination from an expert, so they sought to automate the process. The new process developed starts with preprocessing. The image is transformed into the Hue-Saturation-Intensity (HSI) space and the intensity values are histogram equalized. After this, the diseased sections of the image are segmented out through use of fuzzy c-means.

The segmentation results in a binary image labeling the diseased areas. The binary image is split into sections using connected components and each blob is described by a set of features including descriptors such as lengths, diameter, eccentricity, compactness, extent, Euler's number, and orientation.

Three hundred images were gathered and labeled with three diseases with examples in Figure 2.4: Downy mildew (left), Powdery mildew (middle), and Leafminer (right), as well as healthy leaves. These images were collected using a 3-CCD color camera with a distance of 60 mm from the subject and have a resolution of 720 x 540.

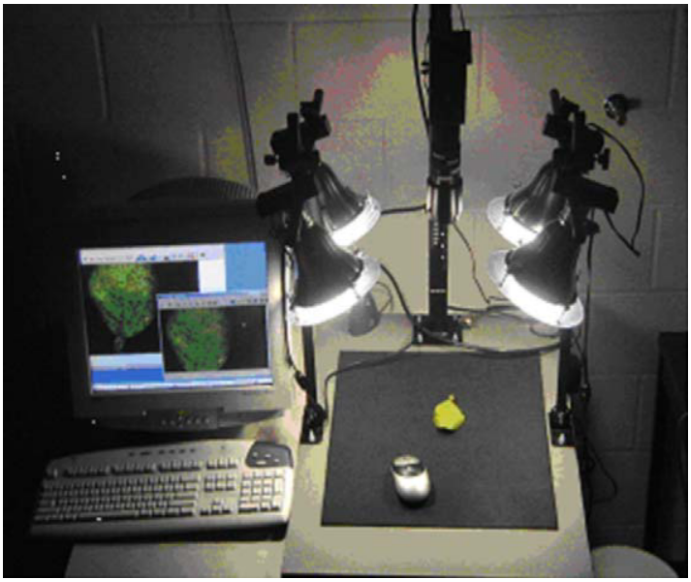
Figure 2.4: Examples of Diseased Leaves From Literature: M El-Helly, AA Rafea, and S El-Gammal. An Integrated Image Processing System for Leaf Disease Detection and Diagnosis. *IICAI,* 2003. [Under Fair Use, 2014].

With these examples, a matrix of neural networks were trained. Three training parameters were varied 1) number of training versus testing samples in steps of 50, 2) neurons in the network, either 5, 10, 15, or 20, 3) number of hidden layers, one, two, or three. The researchers found that the number of layers and training samples significantly affect accuracy while the number of neurons does not.

Ultimately, [10] was able to identify the disease state of a leaf with accuracies of 84% for Downy mildew, 94% for Powdery mildew, 74% for Leafminer, and 98% for healthy leaves.

In a similar study, [9] investigated the automatic identification of diseases on citrus leaves using color co-occurrence methodology. Images of grapefruit leaves with greasy spot, melanose, scab, and proper health were collected with a 3-CCD camera and had a resolution of 480 x 640. Their imaging setup can be seen in Figure 2.5.

Figure 2.5: Image Collection Setup From Literature: R. Pydipati, T.F. Burks, and W.S. Lee. Identification of citrus disease using color texture features and discriminant analysis. *Computers and Electronics in Agriculture,* 52(1-2):4959, June 2006. [Under Fair Use, 2014].



Interestingly, after segmenting the leaf from the background using Canny edges, [9] reduced the image size to 240 x 320; a 75% reduction. The author claims little textural information is lost and computation time is saved. After this, the image is translated into the HSI color space and a Spatial Gray-Level Dependency Matrix (SGDM) is generated for each hue, saturation, and intensity. This matrix includes thirteen values per image plane.

In order to reduce the features needed, [9] used STEPDISC, an SAS statistical procedure, to find redundancy in the feature set for each class. Another SAS procedure, DISCRIM, was then used for classification of the disease on a sample set containing twenty images for each of the four classes. Ultimately, the researcher found that accuracies above 95% were obtained when using hue and saturation features alone.

2.2.2 Full Image Plant

While not being a study on an individual leaf, [21] is very similar in nature to the work of [9]. The environment and image structure is still very controlled and example images can be seen in Figure 2.6. They are top down images containing only the plant, were taken from a distance of 330mm, and have a resolution of 1024 x 768. A total of 640 images were collected of the Sunagoke moss plant each tagged with a water content value calculated from the total weight and initial dry weight of the plant.

Figure 2.6: Example Images From Literature: Yusuf Hendrawan and Haruhiko Murase. Bio-inspired feature selection to select informative image features for determining water content of cultured Sunagoke moss. Expert Systems with Applications, 38(11):1432114335, May 2011. [Under Fair Use, 2014].

(a) Healthy (b) Dehydrated

The goal of [21] is to find a feature set that could identify the water content of the Sunagoke moss. To accomplish this the author generated a large set of features, including color and textural, and then used a variety of bio-inspired feature selection algorithms with neural networks to find the optimal feature subset.

The features generated were for an entire image. An easy example is the color features used of the color mean value of the image, namely the average of an entire color plane: red, green, blue, hue, saturation from Hue-Saturation-Lightness (HSL), lightness, saturation from HSV, and value. Additional to the eight color features used, three morphological features and 99 textural features were used.

The three morphological features were derived from RGB image segmentation that partitioned the image into two categories: green plant and brown plant. Indices were calculated for the amount of green plant and amount of brown plant as well as for the perimeter around the green sections of the plant. For the textural information, each image plane from the grayscale, RGB, HSV, and HSL images were analyzed using a co-occurrence matrix just as in [9].

To slim down the number of features and find redundancies in the set, the author used a variety of bio-inspired feature selection algorithms. These included Ant Colony Optimiza-

tion (ACO), Neural-Genetic Algorithms (N-GAs), Neural-Simulated Annealing (N-SA), and Neural-Discrete Particle Swarm Optimization (N-DPSO). The author ultimately found that the textural features of the HSL saturation plane were most significant to the water content of the Sunagoke moss.

2.2.3 Multispectral and Hyperspectral

An author mentioned earlier, [15] realized the potential of plant stress information beyond the visual spectrum. He references other work done with radiometers for early detection of physiological stress and their success, but states that the equipment is too expensive and hard to use for the task. This work proposes a compromise where the non-visual spectrum will be observed from multispectral cameras.

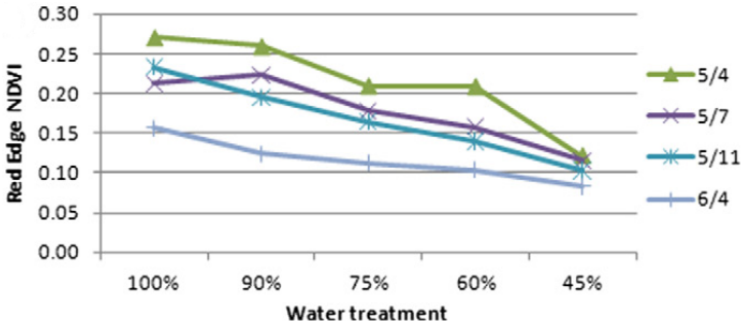
In a related study, [11] used a hyperspectral camera setup to find the spectral band most closely tied to a plant's water condition. The study involved twenty young apple trees split into five groups at different watering levels varying from 100% (healthy) to 45% (dehydrated). Data was acquired using two NDVI sensors, a digital color camera with resolution 1024 x 768, and a hyperspectral camera that produces 12-bit image with resolution 1392 x 373 for every 5 nm interval between 385 nm and 1000 nm. All the sensors were located, facing down, approximately 100 cm above the tree's canopy.

The goal of [11] was observe correlations between the water stress of an apple tree and a variety of narrowband and broadband spectral indices. The author noted the significance of two indices from the study: Red Edge NDVI and NDVI. These indices are the highest corresponding from the narrowband and broadband indices, respectively.

$$\text{Red Edge NDVI} = \frac{R750 - R705}{R750 + R705} \tag{2.6}$$

Red Edge NDVI, as calculated by Equation 2.6, was shown to be the most closely related narrowband index, and overall index, to water stress with a correlation of $r = 0.94$. The plot in Figure 2.7 shows the dramatic decrease in this index with respect to water treatment for all four days sampled.

Figure 2.7: Red Edge NDVI Response to Water Treatment From Literature: Yunseop Kim, David M. Glenn, Johnny Park, Henry K. Ngugi, and Brian L. Lehman. Hyperspectral image analysis for water stress detection of apple trees. Computers and Electronics in Agriculture, 77(2):155160, July 2011. [Under Fair Use, 2014].

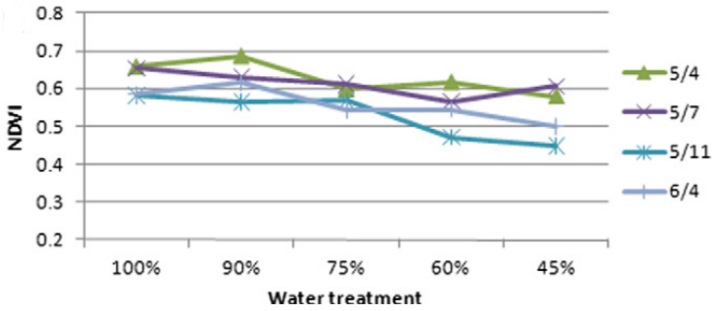


More relevant to the current work, [11] also studied several broadband indices, including the Normalized Difference Vegetation Index (NDVI), the Simple Ratio (SR), the Enhanced Vegetation Index (EVI), and the Atmospherically Resistant Vegetation Index (ARVI).

$$\text{NDVI} = \frac{R800 - R680}{R800 + R680} \tag{2.7}$$

The most significant broadband index found by [11] was NDVI as calculated in Equation 2.7. The plot in Figure 2.8 shows a gradual decrease in NDVI value with respect to water content, although mainly in the trees with less than 75% ideal water replacement.

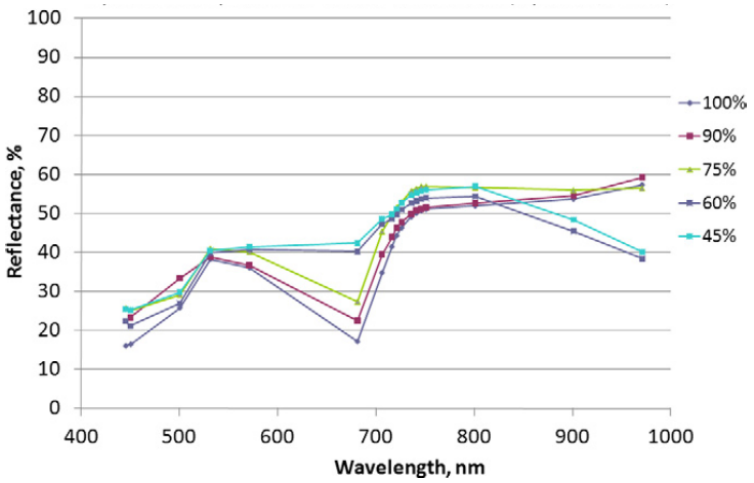
Figure 2.8: NDVI Response to Water Treatment From Literature: Yunseop Kim, David M. Glenn, Johnny Park, Henry K. Ngugi, and Brian L. Lehman. Hyperspectral image analysis for water stress detection of apple trees. Computers and Electronics in Agriculture, 77(2):155160, July 2011. [Under Fair Use, 2014].



Additionally to the study of the reported indices, [11] provided a spectral reflectance plot

seen in Figure 2.9. He notes that a healthy plant typically has a small peak in the green band around 550 nm, a small drop in the red band around 650 nm, and a rising plateau in the NIR band above 750 nm. A plant with a severe water drought, however, has higher reflectance in the red band as well as dropping a small amount in the upper NIR.

Figure 2.9: Spectral Response to Water Treatment From Literature: Yunseop Kim, David M. Glenn, Johnny Park, Henry K. Ngugi, and Brian L. Lehman. Hyperspectral image analysis for water stress detection of apple trees. Computers and Electronics in Agriculture, 77(2):155160, July 2011. [Under Fair Use, 2014].



The author found that the hyperspectral camera was severely sensitive to ambient illumination changes. The camera [11] used required dark and white calibration images to be taken before each set of data and found some hyperspectral images to be unusable due to pixel saturations caused by illumination changes during sampling.

In conclusion, [11] found that no significant findings were in the spectral information for the 100%, 90%, and 75% watered trees. The remainder of the water regimes, however, could be detected through hyperspectral imaging even though no symptoms were visible to the human eye.

2.3 Additional Studies

In addition to the papers just outlined, other related work has been done. [19] and then [17] used vision systems to calculate the spacing of corn in a field. This is done to compare crop yield to initial spacing. Videos were collected of the field with a down-facing camera about one meter away from the ground. The frames of the video were stitched together using correlation and SIFT methods, respectively. Once they had compiled the large image of the entire field, a segmentation method implemented by [46] was used to find the plants. [46]

used the median of the pixels to find the centers while Wang used the intersection of the binary skeleton to find the exact location of the stem.

[38] studies the use of computer vision as a tool in morphological phenotypes and gravitropism. In these areas a large number of plants are bred and studied. He notes that plant information is currently hand collected and often at infrequent time intervals. He uses the example of gravitropism, the measurement of the effect of gravity on a plant limb, to showcase this last point. As of then, 2009, only two measurements would be taken: before and after, but with a vision system the angle of the drooping plant could be continuously measured. Spalding goes more into detail on his segmentation and feature descriptor methods, and these will be discussed later in Section 2.1.

[39] investigated the use of 3D reconstructions of plants to describe the state of crop health for his dissertation. He used a trinocular vision setup to view the plants and create a 3D model. This allowed him to retrieve a wealth of information unattainable through monocular vision, including absolute height of the plant and area of individual leaves, while also mitigating occlusion which is a common problem in plant vision studies.

Chapter 3

Related Theory

Many of the algorithms referenced throughout the document are presented without explanation, but instead are detailed here. These include methods from computer vision and machine learning.

3.0 Chapter Outline

The current chapter is split into two major sections. The first of the two, Section 3.1, details the machine learning algorithms used in this paper. Neural networks and support vector machines are supervised learning algorithms that are used here for image segmentation. K-means clustering, on the other hand, is an unsupervised algorithm that is used within texton generation.

The second half, Section 3.2, reviews the computer vision methods used throughout the study. Image segmentation is outlined as well as the general overview of image registration. Additionally, textons and their generation is detailed.

3.1 Machine Learning

Machine learning is the deriving of a classifying algorithm whose parameters are not specifically programmed. It is designed for problems with large amounts of collected data. Realistically, to properly define machine learning the algorithms need to be split into two groups: supervised and unsupervised.

Supervised machine learning uses labeled examples to train the parameters of a system. This group involves algorithms such as Support Vector Machines (SVMs), Neural Networks (NNs), and even linear regression. These are useful when the user wishes to make a prediction in

the future given the set of training examples they currently have.

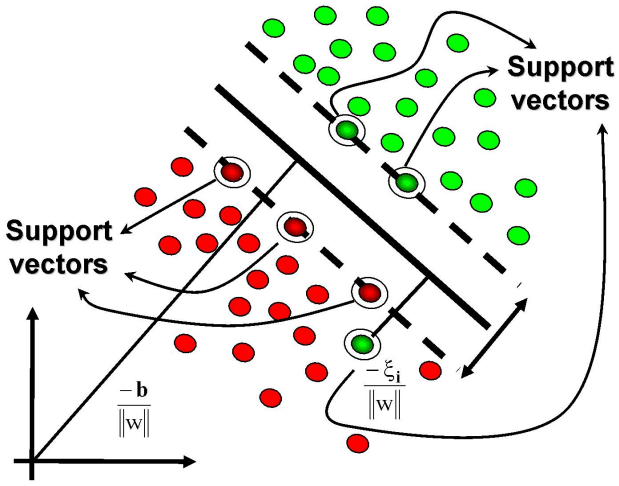
Unsupervised machine learning groups data into subsets. This involves algorithms such as k-means and Chinese Whispers, and is useful for when an unlabeled dataset needs to be classified or even for compression.

3.1.1 Support Vector Machines

Support Vector Machines (SVMs) have recently been the go-to ML algorithm for supervised learning. In related work they have been used to segment leaf images among other things [40]. In this document, they are used to segment plant images.

As the SVM is a supervised learning algorithm the idea is to train the parameters of a boundary between the different labels of the example set. In its most basic form, the SVM does this by maximizing the margin between the boundary and two sets of points. This is illustrated in Figure 3.1.

Figure 3.1: Illustration of a Support Vector Machine (SVM). http://www.cac.science.ru.nl/people/ustun/. [Under Fair Use, 2014]



The highlighted points become support vectors that alter the angle of the boundary. The image in Figure 3.1 is a simple example where the data is mostly linearly separable and the feature set only contains two dimensions. It also presents a linear kernel where the euclidean distance is used to weight the support vector. Alternatively as used in the current work, other kernels can be used such as the Radial Basis Function and allow for nonlinear separation.

For training a binary classifier with a linear kernal for example, the value $\mathbf{w}^T \cdot \mathbf{w}$ is minimized given Equation 3.1, where $\mathbf{w}$ is a vector of weights, x_j and y_j are the feature vector and label

of a training example, and b is a scalar bias value. The minimization is done through numerical means.

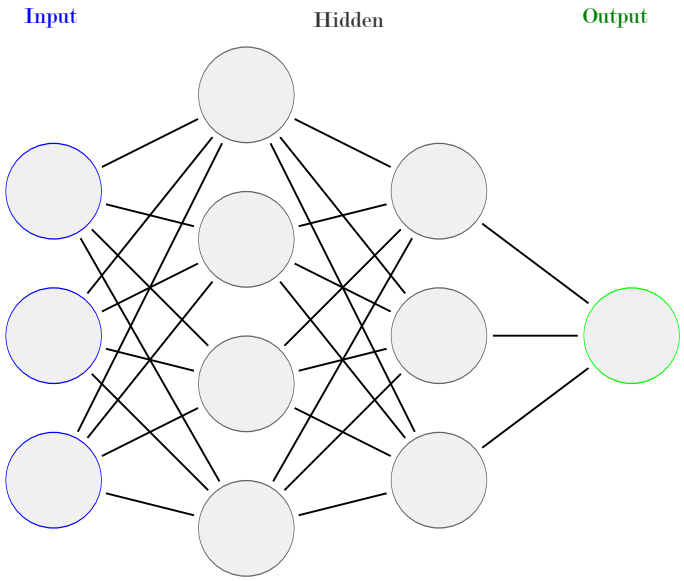
$$(\mathbf{w} \cdot \mathbf{x}_j + b)\, y_j \geq 1, \ \forall j \tag{3.1}$$

3.1.2 Neural Networks

Neural Networks (NNs) are designed to find highly complex and nonlinear relationships between features. They are often also referred to as Artificial Neural Networks or Multi-layer Perceptrons. Authors in the field of vision in agriculture have used them to do plant image segmentation [42, 44], leaf identification [48], and in other ways [10]. In the current work, they are used for segmenting the plant from their backgrounds.

To begin a discussion on NNs their structure must be reviewed. An example NN is shown in Figure 3.2 that has three input features, a hidden layer with four nodes, a hidden layer with three nodes, and an output layer with one.

Figure 3.2: Neural Network Illustration



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^T \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \end{bmatrix}$$

The first layer, the input layer, is simply a row vector of length N_i containing the input

features (of count N_i). These are the set of scalar values that describe each sample. In the above figure, the input features are x_i.

For each node in a hidden layer there is a collection of weights to multiply nodes of the previous layer with, mathematically it is simply a column vector; these are represented by w_{ij} and the lines in the illustration. Therefore, the entire layer is a matrix of size $N_i \times N_h$ with each column corresponding to a node, and the multiplication of the input layer with the first hidden layer is another row vector of length N_h. Additional hidden layers can be added and are constructed in a similar nature.

The process for the output layer is similar in that it is a matrix of size $N_h \times N_o$ where N_o is the number of output nodes; typically one as in this study. This resulting value, or vector of values, corresponds to the binary value(s) classifying the sample as positive or negative.

Similar to SVMs, neural networks are supervised learning algorithms, meaning they require a collection of example instances that are each labeled as a particular class. NN are trained through the process of backward propagation of errors, otherwise known as backpropagation. For each sample in the labeled dataset, a loop is executed. First in the loop, the sample is passed through the network and the output is calculated. The error is calculated between this output and the sample's label, and then is backward propagated to the last hidden layer by changing the weights by the calculated error scaled by the learning rate (a tunable parameter). This is executed for all the hidden layers and the loop is repeated for every training example.

3.1.3 K-Means Clustering

Since it is unsupervised machine learning, K-means clustering does not require the labeling of examples. It has been used to segment images by other authors, but in the current work it is used to generate texton dictionaries as shown later. The result of running k-means on an N-dimensional feature space is k locations that are described by an N-dimensional vector (just as the samples) and represent the centers of samples that have been grouped together.

The algorithm requires a dataset of samples described by N features, and begins by randomly initializing k points within the feature space. The method then enters a loop of iterations that move these k points to their proper centers.

This loop is split into two parts. First, each sample in the dataset is assigned a center, typically through shortest Euclidean distance, and therefore entering that center's cluster. After that, the mean values of every cluster are calculated and these means become the new centers.

The resulting k vectors represent the clusters of the dataset.

3.1.4 Algorithm Accuracies

Accuracy is evaluated with an f-score as shown in Equation 3.2. Precision, as used in the calculation, is ratio of the number of samples correctly predicted as positive to the total number of samples predicted positive. Recall on the other hand is the ratio of the number of samples predicted positive to the number of samples that are actually positive.

$$\mathrm{f} = 2\frac{precision \cdot recall}{prescision + recall} \tag{3.2}$$

While some experiments aim for high precision where it is more likely the positive predictions are correct, or for high recall where more of the positive samples are classified as such, the current work aims to balance these with a high f-score. In image segmentation, a completely white mask will have high recall and a mask that is nearly all black can have high precision. The f-score ensures that there is a balance of pixels that are correctly classified as plant and as not plant.

3.2 Computer Vision

Computer Vision (CV) involves the computational analysis on digital images usually to extract information on the image as a whole. Since images are simply an array of numbers, many digital signal processing and machine learning algorithms can be used to find patterns within an image. The current work involves methods of computer vision that segment and image into parts, match similar points in different images, and obtain responses from filters.

3.2.1 Definition of a Camera

Cameras can be viewed most simply as a pinhole projection. That is to say, given a light source and an object, each point on the object will reflect light in all directions. In order to assure that only one ray of light from each point on the object is captured in the image, it must go through a pinhole filter. Once the light is through the pinhole, it creates a flipped image projected onto a screen past the filter with only one ray from each point on the object.

To make a camera one simply has to replace the screen on the right hand side of the pinhole with a piece of film or the CCD of a digital camera. In the current work, digital cameras are used with CCDs that are sensitive to visual, near infrared, shortwave infrared, and long wave infrared, although for the LWIR a similar device called an uncooled microbolometer array is used. These CCDs react to being hit by light, creating a voltage for the camera's circuitry to read.

The camera reads the voltage from the CCD and records the intensity into an N-bit image. An 8-bit image is assumed since that is what the current work uses. Each pixel then is represented by an 8 bit number, 0 through 255, typically where zero corresponds to complete darkness and 255 is full intensity. The CCD, however, is an array of sensors creating an array of pixels with size corresponding to the resolution of the image, 320 x 240 for the SWIR camera for example.

While the previous explanation creates a monochrome image, some cameras create multiple planes within an image such as typically done with a visual spectrum camera. It has been shown that color images require at least three coordinates to accurately represent a scene for human viewing [21]. This can be accomplished by simply having three CCDs for each desired plane (typically RGB). The visual camera used in this study, however, uses a less expensive alternative called a Bayer filter. This device is a grid of red, green, and blue filters that only allow the corresponding spectrum of light through each sensor in the CCD. Instead of reducing the resolution three fold, however, the RGB values for each pixel is determined by averaging the pixels surrounding it. The resulting image, using the JAI visual camera as an example, is a 1024 x 768 x 3 matrix of values between 0 and 255 with each of the three planes corresponding to red, green, and blue.

3.2.2 Image Segmentation

Simply put, image segmentation is the process of separating the pixels in an image into groups corresponding to desired classes. In the current study, for example, each pixel in the image is determined to be plant or background. The typical result of an image segmentation is a black and white, or binary, image with white corresponding to positively classified as subject, or foreground, and black negatively classified as background or just not the subject of the current class. This resulting segmentation image is often referred to as a mask.

Thresholding

Several methods exist to classify each pixel as subject or background. Thresholds on image values are the simplest of classification algorithms allowing all pixels with a value above (or below) the threshold to be classified as the subject. This is easy and works well with a specifically constructed image and with a plain background, but often requires the hand tuning of the threshold value and is sensitive to changes in lighting.

Thresholding can become more complicated when multiple values are required. If the image is monochrome two thresholds can be used to classify the pixels within a specified range as subject, or with a multi-plane image thresholds for each plane can be used. One can imagine, however, that increasing the number of thresholds produces a more accurate segmentation, but also multiplies the problems of hand tuning and changes in lighting.

Alternatively to placing a threshold on the pixel values, a threshold can be applied to the image response of a filter. Similar to digital signal processing, filtering an image usually involves the convolution a small array of weight values over the entire image. This filters can show responses for x, y, and total gradients, shapes of varying sizes, and many other features.

Machine Learning

Another method of distinguishing subject pixels used in this study is via machine learning algorithms. Methods such as neural networks and support vector machines can be trained to classify a pixel with hand labeled examples from another image. These algorithms are generally used for specific cases since it is required to have a representative set of examples. Namely, not only are enough positive examples needed to generalize the algorithm, but negative samples as well. This means examples of every possible type of background pixel is needed, which can be difficult for an unstructured environment.

The machine learning algorithms typically output -1 to +1 or 0 to 1 with the top half of the range being a positive result. This threshold can be altered depending on the case. For example, if it were more important to correctly classify positive examples as positive than negative as negative, then the threshold can be made closer to 1 resulting in more false positive but more true positives as well.

Post-processing

Any image segmentation method generally has noise and there are many ways to reduce it. Most commonly, erosion and dilation is used. These are morphological operations that are used to smooth out a mask and examples of both are shown in Figure 3.3. These methods carve away or fill out a mask by sliding a filter over the image using an AND operation. For example in dilation with a filter centered in a circular shape, each resulting pixel is set positive if there is a positive pixel within the radius of the filter in the original image. Erosion is simply this operation on negative pixels.

Figure 3.3: Example of Erosion and Dilation

(a) Original (b) Erosion (c) Dilation

Erosion, as shown in Figure 3.3b, helps remove random pixels in the background incorrectly classified and smooths the edges of the mask. Figure 3.3c shows that dilation fills in holes in the mask (pixels incorrectly classified as negative) and also smooths the edges of the mask. Typically erosion is used with dilation following, which removes random background pixels and then rebuilds edges of the mask that were eroded while also filling in the holes in the mask. This is what is shown in Figure 3.3.

More specific methods can be used on a case by case basis. For example, every blob in a mask can be filtered by circularity, size, or elongation. Alternatively each blob could be compared to a template or its distance to another blob. As done in the current work, these blobs are labeled by a method called connected components.

Connected components labels each pixel in a mask by going left to right and row by row by checking the pixel above it and pixel to its left. If either of these pixels is labeled and the current pixel is positive, the label transfers to the current pixel. If neither are labeled, the current pixel is given a new number. Finally the pixels groups are reviewed to find groups that belong in a single label.

3.2.3 Image Registration

Image registration is the process of aligning images of a single scene from different angles. The process involves three steps, finding points of of interest, describing these points, and matching the points between images.

Finding Key Points

Points of interest, called key points, are areas of an image that are significant and unique enough to be found from a different angle. These are typically high gradient areas and are typically corners.

The method used in the current work is the Harris corner detector. Corners are significant because they can be easily localized. If a section of an image only has gradient in one direction, y for example, then there is little change in the other direction; a region could be slid left or right and no difference would be found. That means that if a point in that area was selected, it could be matched to any point along the x-axis near it. Corners, however, have gradients in both directions and also have change in gradients in both directions, and therefore can be localized easily.

This step results in a list of points within an image only described by x and y coordinates as well as a size parameter. Next, these points need to be described in a way that makes each one unique.

Describing Key Points

While there are many methods for describing key points, they all result in an array of numbers. This N-dimensional vector can contain anything from pixel values to filter responses to histograms. The Scale-Invariant Feature Transform (SIFT) keypoint descriptor implementation is used as a comparison point for future descriptors. This method is known to work well with two images of the same scene from the same camera or type of camera. Namely, this method works well matching RGB to RGB, or NIR to NIR, etc.

The Edge-Oriented-Histogram (EOH) Algorithm as described by [49] is a descriptor that ignores the pixel intensities and instead describes the orientation of the detected Canny Edges [50]. The Canny edges are found for the image using thresholds relative to the maximum gradient magnitude of the image. Then, for each key point, an N x N region of the Canny edge image is observed and is split into 16 subregions. Five 3 x 3 edge filters (0°, 90°, 45°, 135°, and no orientation) are convolved over each subregion. Then, for each subregion, a histogram is generated from the set of maximum filter responses. All 16 histograms are concatenated into an 80 feature long descriptor, which is then normalized by the Euclidean norm.

Once all the key points are described, they are compared using a fast approximation to the Euclidean nearest neighbor called FLANN.

3.2.4 Image Warping Algorithms

To register these images they must be warped so two pixels in the same location from different images will correlate a physical location in the 3D world. Three image warping techniques are observed assuming that matching points have been selected either automatically or by hand.

The Sensor Specifications (SS) Algorithm is the simplest approach to register multiple images [51]. It makes the assumption that all cameras are barrel-sighted and only takes into account

the transformation caused by the field of view and resolution changes. The instantaneous field of view is calculated by dividing the field of view dimensions for each camera by their corresponding pixel dimensions. Then by dividing the desired field of view by this number, one will obtain the pixels to crop from each image.

This is calculated for each image by using the smallest field of view (RGB/NIR) as the desired, and then resizing each image to the smallest resolution (SWIR). This algorithm does not work for most cases since it does not account for the translation or rotation of the cameras, or for the occlusion in the scene. If the scene is far enough from the camera, these matter little, but our cameras are not rotationally aligned perfectly which is detrimental to this algorithm. Additionally the field of view calculations need to be precise.

It is, however, a useful preprocessing step as it is very simple to implement (especially in that it does not require matched points) and makes the images more similar in a number of ways. Firstly, it removes portions of the images (particularly the LWIR) that are not in the other images, such as outside the door on the left of Figure 4.6. More importantly, it makes the images of the same scale, so any matched features between the images must also be the same scale.

The homography transformation is similar to the SS algorithm in that it does not account for the physical translation of the cameras or for any occlusion resulting there of. It has the advantage over the SS algorithm by accounting for the possible rotation of the cameras, and the disadvantage of that it requires calibration.

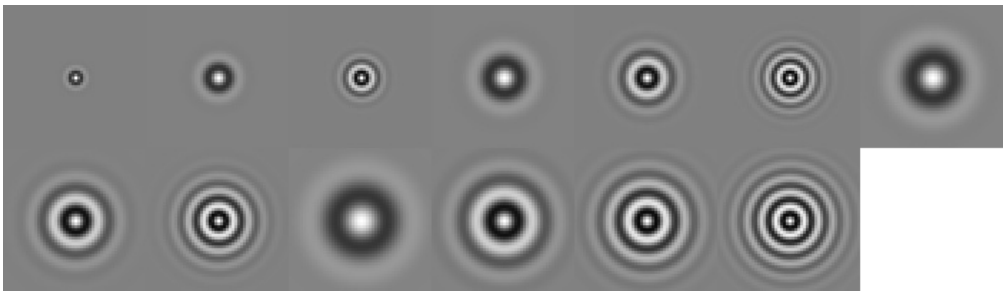
The last method attempted was warping through Thin-Plate-Splines (TPS) [52]. The TPS warping method uses the algorithm created for modeling a thin metal plate warping under the strain of forces. For each matched point between images it places a "force" at the point straining the image toward it. Since the algorithm is nonlinear, it should account for everything homography does, plus help with the problem of disparity.

3.2.5 Textons

As described by [53], textons are local features that allow humans to quickly discriminate between textures. In terms of computer vision, they are a set of filter responses that describe a texture within an image. The textons are generated using a set of filters and a group of training images. The method used in the current work was taken from [54].

The first step in generating the textons is to decide on an input filter set. The Schmid filter set was chosen for this study because it contains linearly independent and rotationally invariant filters. These Gabor-like filters are each 50 x 50 pixels and are shown in Figure 3.4.

Figure 3.4: Schmid Filter Set



The Schmid set is generated with Equation 3.3. F is the value of the pixel in question and r is the distance from the center. The pair (σ, τ) are varied to create the set in Figure 3.4. The pairs as left-to-right and row-by-row are $(2,1)$, $(4,1)$, $(4,2)$, $(6,1)$, $(6,2)$, $(6,3)$, $(8,1)$, $(8,2)$, $(8,3)$, $(10,1)$, $(10,2)$, $(10,3)$, and $(10,4)$.

$$F(r,\sigma,\tau) = F_0(\sigma,\tau) + \cos\left(\frac{\pi\tau r}{\sigma}\right)e^{-\frac{r^2}{2\sigma^2}} \tag{3.3}$$

For each class, the set of filters is convolved over a group of training images to generate a response set. K-means clustering is computed on the response set and each center found describes a texton for the class. The textons can be visualized by linearly combining the input filter set with proportional corresponding to the responses. A response from a new pixel would then be labeled by the texton closest to it.

New textures are the described by a histogram of texton labels of its pixels. A perfectly classified texture would contain only textons from a single training set, although commonly textures share textons.

Chapter 4

Experimental Design

The goal of the current work is to find which multispectral cameras and which imaging features can be used to monitor a plant. As described in Section 4.1, several cameras are in use including the spectrum for color (430-350 nm), NIR (750-900 nm), SWIR (950-1700 nm), and LWIR (7500-16000 nm). Through these cameras and machine vision, a study will be made to find the pertinent information for agricultural use.

4.0 Chapter Outline

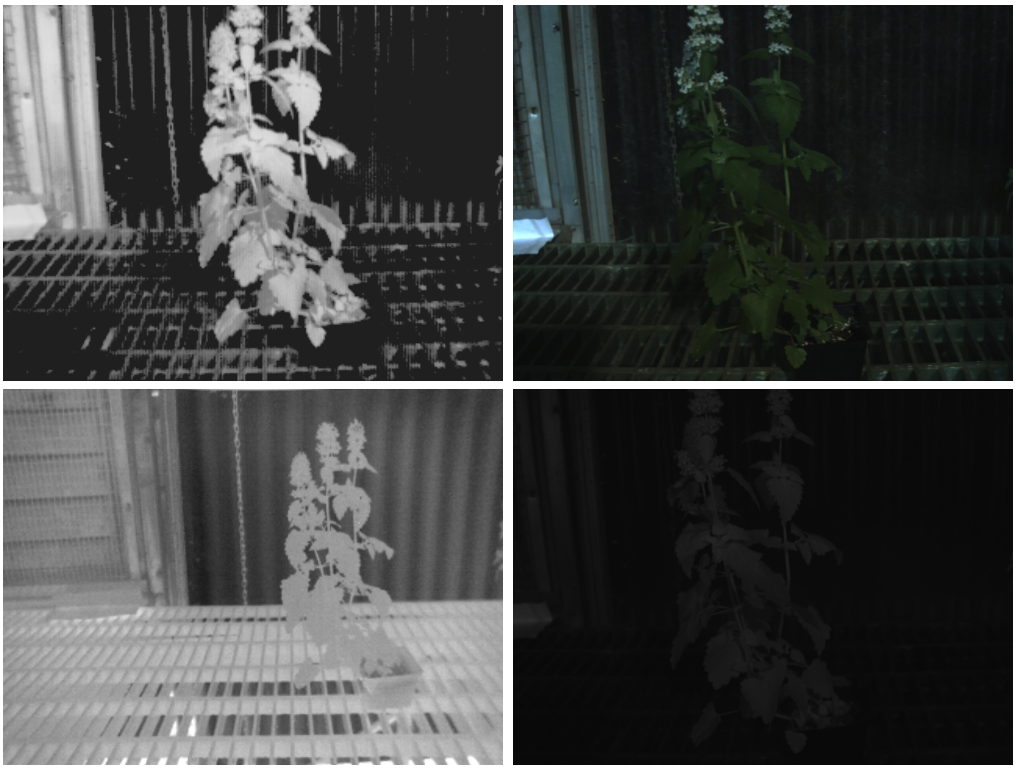
The chapter starts with a review of the cameras used in the study in Section 4.1. This camera rig was used for the plant stress experiment and the same dataset was used for image segmentation experiments. Example images from this set are shown in Figure 4.1 for perspective throughout. Note they have been scaled for convenience and should be compared to that shown in Figure 4.6.

The experiments are conducted with respect to computer vision in agriculture and can be separated into several groups. The first being image segmentation in Section 4.2, the process of labeling each pixel in an image as plant or background. Various approaches are taken to segment images collected for the water stress experiment mentioned later and each is compared.

Secondly in Section 4.3, an image registration experiment is designed to align the multispectral images. It is set up to test the feature extractors and descriptors across the multispectral images.

Lastly, an experiment is conducted to find machine vision features required to identify water stress on plants. Section 4.4 details how a group of plants are placed under water stresses and multiple computer vision methods are tested.

Figure 4.1: Examples of Plant Images Collected: RGB (top right), NIR (bottom right), SWIR (top left), and LWIR (bottom righ).



4.1 Mechatronics Multispectral Camera Rig

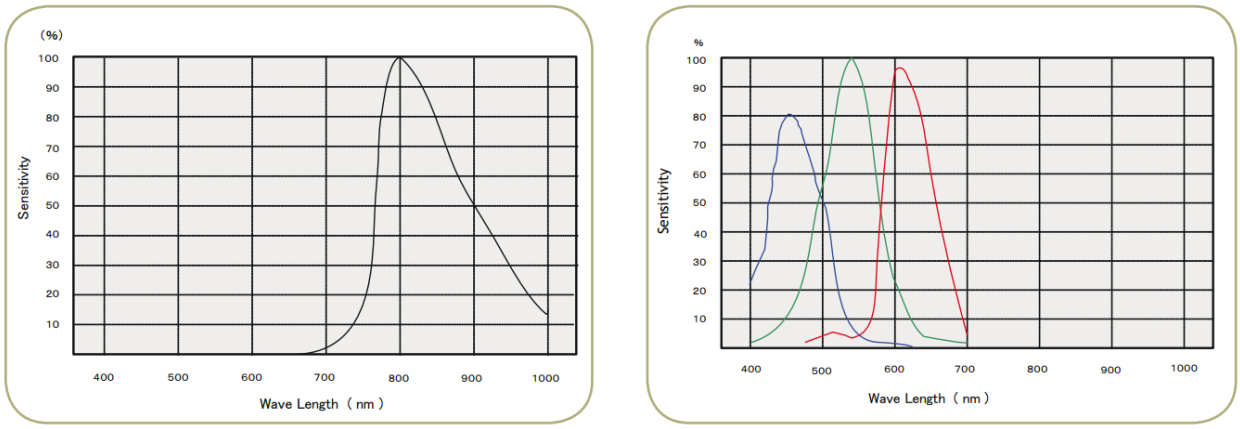
Throughout this paper the multispectral camera rig will be referenced. This setup is comprised of several types of cameras in different spectrum, each presented in detail here. An image of the camera rig can be seen in Figure 4.2. Note that only the three left-most cameras will be used in this study. For more on how a typical camera works refer to Section 3.2.1

Figure 4.2: Mechatronics Multispectral Camera Rig

JAI AD-080GE VIS/NIR Camera

The JAI camera is in actuality two cameras producing a 24-bit RGB image for the visual spectrum and an 8-bit grayscale image for the near infrared. It has two CCDs that each produce a 1024 x 768 pixel image. It is equipped with a lens that produces a field of view of 43.3° x 33.1°. The spectral responses of the two CCDs is presented in Figure 4.3. This information is within the industrial datasheet [55].

Figure 4.3: The JAI Camera Spectral Response. JAI. *AD-080GE Digital 2CCD Pogressive Scan Multi-Spectral Camera User's Manual*, 6 2012. Ver. 1.1. [Under Fair Use, 2014].

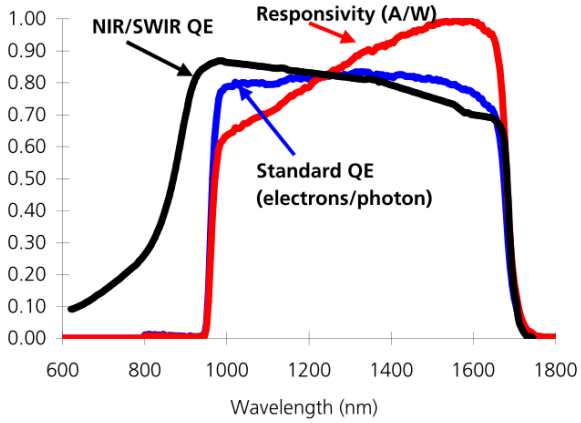


The JAI camera includes the GigE vision SDK for image acquisition and exposure control. For the length of the experiment, the internal exposure setting as well as the lens aperture were attempted to be kept constant to allow for consistent measurements throughout.

Goodrich SWIR Camera

The Goodrich camera is a high sensitivity, InGaAs shortwave infrared (SWIR) camera. It produces an 8-bit monochrome, 320 x 240 image of the SWIR spectrum with a field of view of 43.6° x 33.4°. The sensitivity of the Goodrich SWIR camera is shown in Figure 4.4.

Figure 4.4: The Goodrich Camera Spectral Response. Sensors Unlimited. *SU320HX-1.7RT Mil-Rugged High Sensitivity InGaAs SWIR Camera with Advanced Dynamic Range Enhancements.* [Under Fair Use, 2014]

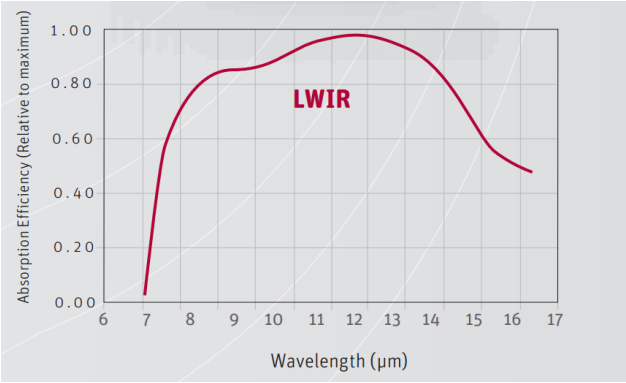


SWIR is reflected off of many surfaces such as cotton clothing, hair, and, particular to this study, plants. It is also absorbed almost completely by water.

Xenics Gobi-384 LWIR Camera

The Gobi camera is an uncooled thermal camera that produces a 8-bit monochrome image of the longwave infrared spectrum. The resulting image is 384 x 288 with a field of view of 56.1° x 43.6°. This FOV is much larger than that of the other cameras and even results in a fisheye effect in the image. This is corrected with the use of a MATLAB script for all images in the study. This information is within the industrial brochure [57], including the sensitivity of the Gobi-384 LWIR camera as seen in Figure 4.5.

Figure 4.5: The Gobi Camera Spectral Response. Xenics Infrared Solutions. *Gobi-384 Industrial Brochure.* XB-015 Issue 3. [Under Fair Use, 2014]



This camera is interfaced with Xenics software for configuration and then CameraLink SDK for data acquisition. The exposure and range settings of the camera were set to automatic.

Camera Rig Summation

For ease of use, a summation of above information has been put into Table 4.1.

Table 4.1: Mechatronics Camera Rig Summation

Camera	**Spectral Response**	**Resolution**	**Field of View**
JAI RGB	430-650 nm	1024 x 768	43.3° x 33.1°
JAI NIR	750-900 nm	1024 x 768	43.3° x 33.1°
Goodrich SWIR	950-1700 nm	320 x 240	43.6° x 33.4°
Gobi LWIR	7500-16000 nm	384 x 288	56.1° x 43.6°

For further comparison, Figure 4.6, featuring the author, has been added to show the difference between the raw images of the cameras (left-bottom LWIR, left-top SWIR, right-bottom RGB, right-top NIR). It can be seen that the JAI, the images on the right, produces much larger images, but also has a smaller field of view.

Figure 4.6: Camera Rig Raw Images Comparison

The image in Figure 4.2 presents the camera rig with cameras additional to those mentioned. The camera furthest to the right of the image is the polarimetric camera developed by Mark Umansky of the Mechatronics lab. While the data from this camera would have been interesting, it is not used in this study because its field of view is one third that of the other cameras and therefore the spatial resolution of the other cameras would to have been greatly reduced to fit the plant in all the images.

The camera next to it is another JAI camera identical to the one mentioned before. It was not used for similar reasons to the polarimetric and that it would have produced duplicate data. It would, however, have given the option of 3D data and this is discussed later in Future Work, Section 6.2.

Lastly, the device on top is the Microsoft XBox Kinect. It was temporarily placed on the rig for an attempt at calibration mentioned in Section 4.3 for image registration.

4.2 Image Segmentation

The ultimate goal of plant segmentation is to create an image mask that will label each pixel in the image as plant or not plant. This will allow the pixels representing the plant to be used in further studies without the interference of the background pixels. This experiment is essentially as follows: example images from the visual and NIR are hand labeled as a

ground truth, several segmentation methods are compared on these examples, and the most accurate method is selected for use in the stress experiment.

As detailed in Section 2.1, authors have attempted many methods for segmenting an image containing a plant. Their methods to determine which pixels represent a plant range from simple intensity thresholds to vegetation indices to machine learning algorithms, and in this study a selection of these methods will be tested. To do so, a ground truth must be established. For this, six images were hand labeled by essentially erasing the background of an image using a standard image editor (GIMP). An example of which can be seen in Figure 4.7.

Figure 4.7: Example of Hand Labeled Image Segmentation

Only six hand labeled images are needed to classify the remaining 714 because within each image there are 786,432 examples with around seventy thousand being positive and the rest negative. The six images were chosen for their diversity, namely to obtain examples from basil and catmint, dehydrated and over watered, and at different stress levels.

4.2.1 Thresholds and Indices

The most popular plant segmentation method is placing a threshold on either a pixel value or calculated vegetation index, as seen in Section 2.1 and 2.1.4. These are popular for their ease of implementation and low computational cost, however, do not generally adapt well to an uncontrolled environment.

Thresholds will be placed on the RGB and NIR values of the original images as well as converted HSV image. Namely, these values are red (R), green (G), blue (B) from the visual JAI camera, near infrared (NIR) from the NIR JAI camera, and hue (H), saturation (S),

and value (V) calculated from the RGB values as seen in Table 4.2. Note that ofr the HSV calculations, the value (V) is calculated first, then used in the other equations.

Table 4.2: HSV Calculations

Color Plane	Plane Equation
Hue (HSV)	$60(G-B)/(V-\min(R,G,B))$ if $V=R$ $120+60(B-R)/(V-\min(R,G,B))$ if $V=G$ $240+60(R-G)/(V-\min(R,G,B))$ if $V=B$
Saturation (HSV)	$\frac{V-\min(R,G,B)}{V}$
Value (HSV)	$\max(R,G,B)$

In addition to the thresholds on these pixel values, thresholds will be tested on vegetation indices created by various authors. These are explicitly state in Table 4.3, but for more information see Section 2.1 and 2.1.4.

Table 4.3: Vegetation Indices Used for Segmentation

Index Name	Index Equation
ExG	$2\times G-R-B$
NDI	$\frac{G-R}{G+R}$
Marchant	$\frac{r_m}{g_m^A}$
NDVI	$\frac{NIR-Red}{NIR+Red}$

The effectiveness of these thresholds will be determind as follows. For the set of training images, those that have been hand labeled, the above values will be calculated. Then for each value or index, upper and lower thresholds will be varied and applied. Namely, the lower threshold will vary by 100 steps between the minimum and maximum of the value, and for each lower threshold the upper threshold will vary by the same step size, but only between the lower threshold and the maximum of the value.

Finally for each set of thresholds, the f-score will be calculated and the maximum of this will be noted. The f-score weighs the predicted positive and negative accuracies (for more information see Section 3.1.4).

4.2.2 Machine Learning Algorithms

The aforementioned thresholds work well for simple backgrounds as in [40, 44], but often fail when a background is complex or a similar tone to the foreground. To find the plant in an

unstructured image it is often necessary to employ machine learning on the pixel values or values otherwise calculated such as the indices in Table 4.3 or texture features.

This section of experiments aims at finding the machine learning algorithm best suited for seeing the plant in an image. More specifically, a series of algorithms will be tested along with a variety of input features. These features are separated into relevant groups that will help suggest what information is most useful.

The algorithms that will be tested include neural networks and support vector machines. Note that the these algorithms are trained using the hand labeled images mentioned above. Approximately one million pixel examples are used to train the neural networks while about fifty thousand are used for the support vector machines. The reduction in sample size was required due to the heavy computation time needed to train the SVM. The order of the samples were randomized for every algorithm.

The Neural Network (NN), or alternatively multilayer propagation, has been widely used in the agriculture computer vision community; at least in [10, 21, 42, 44, 46, 48]. This type of algorithm helps find nonlinear relationships in the features without specific calculation. The parameters of NNs that will be varied for the experiment will include the input features, number of nodes in the first hidden layer, and number of nodes in the second hidden layer (including zero, no second hidden layer). The dlib, [58], implementation is used in this study and more information on NNs can be found in Section 3.1.2.

The Support Vector Machine (SVM), is recently the most popular of the machine learning algorithms and has been used in a couple of aforementioned papers [20, 40]. This algorithm generalizes the data well and allows the easy exchange of the kernel giving it a highly flexible nature. The SVM parameters that will be varied include the input features and the kernel type. Again, the dlib [58] implementation is used in this study and more information on SVMs can be found in Section 3.1.1.

The input features for the above mentioned algorithms will be placed into groups for study. These groups are shown in Table 4.4 and are split into convenient sets. Moreover, the sets are selected as such that would be typically used, for example the Visual Pixels subset is a likely candidate for someone to first try using if they only had an RGB camera, and similarly for JAI pixels if the near infrared channel was also available to them.

Table 4.4: Segmentation Feature Subsets

	All Features	JAI Pixels	Visual Features	Visual Pixels
Red	•	•	•	•
Green	•	•	•	•
Blue	•	•	•	•
NIR	•	•		
Hue	•		•	
Saturation	•		•	
Value	•		•	
ExG	•		•	
NDI	•		•	
NDVI	•			
F	•		•	

4.2.3 Post-processing a Segmentation

As will be seen in the Results chapter, it is common for segmentation algorithms to mislabel background pixels as foreground and visa versa. There are a variety of methods to remove these pixels including dilation and erosion or image blur, but for this work, a system using connected components was employed.

Connected components labels sections of pixels in a binary image that are grouped together. Its namesake stems from each pixel belonging to the group of any pixel directly adjacent to it, or connected to it. This was used on a segmented image to identify any sections of the image that were mislabeled. Additionally, two steps were used before connected components.

First the image was eroded. This removes any small blobs or single pixels floating around in the background. Erosion simply removes an edge layer around a blob in a binary image. Secondly, the image was dilated. This undoes the erosion on the larger blobs while doing nothing to the blobs that have been completely removed. It also connects larger blobs that are close to each other, such as two leaves connected by a stem too small for the camera to see. This method of connected components on a dilated image was also used by [40].

4.3 Image Registration

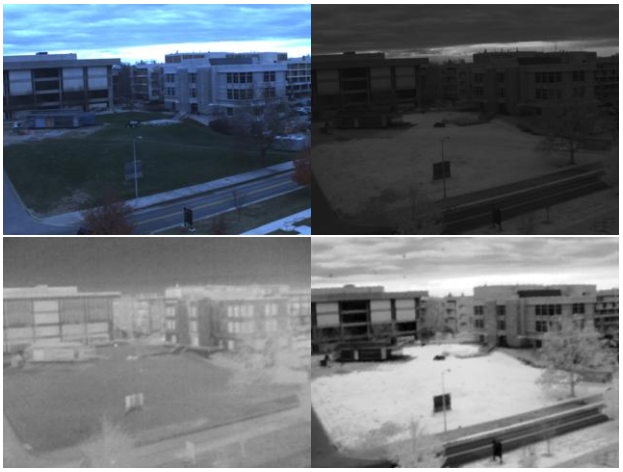
To evaluate the each approach, a ground truth is developed where the pixel location relationships are known from one image to another. The results from the point matching algorithms between each channel are then compared to the ground truth. Two methods to

create a ground truth are attempted. The first involves registering all the cameras to an X-Box Kinect. Images of a checkerboard pattern appearing in all the cameras are used with the MATLAB Camera Calibration Toolbox to generate each camera's intrinsic and extrinsic parameters. After this, the parameters are used to transform the 3D points generated by the Kinect into pixel coordinates for each camera.

Several approaches were taken to create a checkerboard to show up in every camera. The most successful was a heated, spread, white shirt and placing a cooled, cut out, black checkerboard pattern over the shirt. This worked since cotton has high NIR reflectance, but did not give the resolution needed for calibration.

The second method for ground truth generation is to take images of a scene with no objects closer than a certain threshold and then compute the homographic transformation. This threshold can be found using trigonometry and the two constraints of the LWIR is five inches from the RNIR (the furthest distance between cameras) and a maximum registration error of one pixel (of a 320 x 240 resolution). This gives a minimum distance of 156 ft. Images of nine different scenes were taken from the top of a parking garage and an example is shown in Figure 4.8.

Figure 4.8: Example of Registration Ground Truth



A few approaches at image registration are attempted with the same goal in mind: register all images to the NIR image of the left-most JAI camera. Namely, this work aims at accomplishing this task for the SWIR and LWIR images, since the NIR is already aligned due to the barrel-sighted setup of the camera. Also note that before each algorithm the LWIR lens distortion was corrected through a MATLAB script by Jaap de Vries found on the MATLAB Central File Exchange. Additionally, the OpenCV implementation is used for the Harris corner detector, SIFT, Simple blob detector, Canny edges, and FLANN. The EoH descriptors were written in C++ after the design of [49].

First for the experiment, for each image in the dataset, key points are found in three ways:

Harris corner detector, SIFT feature detector, and a Simple Blob detector. Each of these has its own parameters, mainly the general scale of features, and are tuned to return the greatest number of points.

Next, EOH and SIFT are used to extract descriptors for each key point. EOH contains parameters of region size, which was set to 120 x 120 as recommended by [49], and the Canny edge parameters which were set low for the maximum amount of information. The SIFT parameters are left at default.

Finally, these descriptors are matched using FLANN which is a fast implementation of Euclidean nearest neighbor. This returns a match for nearly every point, so they are filtered as mention earlier. The match is then judged correct if is within five pixels of the ground truth. The accuracy of the matching is determined as the percent of filtered matches that were correct.

Image warping methods are also investigated by visual inspection. Hand labeled point matches are used for homography and TPS warping, and the camera parameters of the rig are used for SS warping.

4.4 Plant Health Classification

Given that images can be registered across spectrum and the plants can be segmented from these images, an experiment is designed to derive plant health from the collected information. A group of seedlings are planted in a greenhouse and multispectral images are taken daily. The plants are stressed in various ways until the resulting physiological changes ceased thirteen days later.

4.4.1 The Stressed Plant

Of the collection of plants there are three stress categories: over-watered, under-watered, and normally watered for a control group. Each plant grew in an individual one-pint plastic seedling pot.

Water stress specifically can be seen in plants as a change in leaf color and temperature due to the closing of stomata and the impedance of photosynthesis and transpiration [59]. This is what most studies in the area have looked for and why a change in the hyperspectral reflectance has been investigated. With basil and catmint in particular, drought causes the leaves to droop and become darker and dull in color within a couple days and eventually the leaves will brown.

Additional to these features, however, water stress can cause leaf curling and wilting because of loss of cell turgidity [11]. Wilting has been tracked using computer vision by several

authors, but requires the constant monitoring of a plant that must be indoors [11].

Over watering a plant causes hypoxia and can result in stunted growth and eventual death of the plant. In catmint, the symptoms of over watering are similar to drought but on a slower scale. In basil, the leaves begin to yellow and die starting at the bottom.

To assure the plants are over watered, they are placed in a tub of water where the waterline is above the soil level. The pots are held underwater by placing rocks on the soil. Having them completely underwater creates an environment where the roots have nowhere to grow, as they will grow towards dry soil.

The second stress, under-watered, replicates a drought environment which can greatly stunt growth and quickly kill a plant. To simulate a drought the plants were not watered. A grower can tell if the plant is dehydrated usually by the dryness of the soil or the texture of the leaves.

The last group was for control and they were watered regularly every night.

Figure 4.9: Examples of Basil and Catmint

(a) Healthy Basil (b) Healthy Catmint

Within each group, there are two types of plants: basil and catmint. The basil, *Ocimum basilicum*, grows to be between six inches and two feet and one to six inches wide. It has large green silky leaves, and those at the base of the plant may turn yellow if the plant has been stressed. Basil grows best outdoors, but can be grown indoors as in this experiment. An example of a basil plant used is shown in Figure 4.9a.

Catmint, *Nepeta*, grows to be one to four feet tall and flower white at the top. An example is shown in Figure 4.9b.

While the production of neither of these plants is on par with a main crop such as corn or wheat, they do have uses and many plants share symptoms resulting from stress. Basil, for

example, is used as an herb in many dishes and can be found at any grocery, and every year 100 tonnes of essential oils are produced using basil. Additionally, basil has a trade value for the potted plant of around $15 million a year. Catmint it is also grown for its essential oils for use in medicine, and the plants are ground for catnip. More importantly, the way these plants react to stress is similar to the way many other crops react, such as bell peppers, cotton leaves, and tobacco.

4.4.2 Image Construction

One of the most important practices in computer vision is the construction of the image. The amount of detail in the image and the structure of the background can change the success of the algorithms to even which algorithms are used. Decisions are made for image resolution with respect to area of view, the subject background, and the environment's lighting. These are generally used to make the subject of the image stand out from the background for ease of processing.

As seen in Chapter 2, other authors have varied between structured and unstructured environments for the plant images. Authors such as [9, 10, 15, 38] have restricted images to having a single leaf be the majority of the image and the rest being a plain, either white or black matte background. The camera setup from [9] specifically can be seen in Figure 2.5 and is a good example of a highly structured environment.

Other authors, such as in [16, 22], took images of their full plants in the field from the top down. This is a slightly less controlled environment, however, the plants still stood out from the plain background of the dirt. Another less controlled environment is in [20] where an image was taken of a plant from its side, however, the background of the image still consisted of a plain wall and whiteboard.

In the current work, the images are in a less structured environment then the papers previously mentioned. As seen in Figures 4.9a and 4.9b, the images are taken from the side of the plants with a background that has an intensity comparable to the subject. This angle of capture was chosen largely for convenience because of the configuration of the equipment (it was also used by other people at the time) and it had to be moved every day, however, it provides an angle similar to what a small ground vehicle would produce. This is of interest since the ultimate goal of a plant monitoring system would be the automatization through use of unmanned ground vehicles.

Due to the greenhouse being a space commonly used for other groups, the cameras used are transported and set up daily as mentioned before. This created a dataset with backgrounds that were similar but not identical, therefore methods of background removal such as median subtraction could not be used. The backgrounds are also not a consistent color making simple thresholds less likely to work. They are, however, similar enough throughout the days to where a machine learning algorithm could be taught to recognize the patterns.

Many authors reflect on the importance and impact of lighting on a computer vision study. A change in lighting will cause a change in intensity between images and will result in static thresholds not working. Some cameras must have a fine tuned exposure as well and a change in lighting can cause an image to be washed out or too dark resulting in a low signal to noise ratio. The lighting in the current work was held consistent by taking images strictly after sundown in the artificial light of the greenhouse. This is considered more later in this section.

Camera Selection and Calibration

The selection in cameras for the current work is made by what is available, however, it is important to review what other authors have success with and how it compares. Camera features considered include the resolution of the resulting image, distance from the subject, and subject area. A summary of the cameras used in this experiment is shown in Table 4.5, but a more detailed description is in Section 4.1. Note that the area per pixel was calculated assuming that the plant was about half a meter from the cameras.

Table 4.5: Mechatronics Camera Rig Summation

Camera	**Spectral Response**	**Resolution**	**Field of View**	**Area Per Pixel**
JAI RGB	430-650 nm	1024 x 768	43.3° x 33.1°	0.6 mm^2
JAI NIR	750-900 nm	1024 x 768	43.3° x 33.1°	0.6 mm^2
Goodrich SWIR	950-1700 nm	320 x 240	43.6° x 33.4°	5.8 mm^2
Gobi LWIR	7500-16000 nm	384 x 288	56.1° x 43.6°	6.7 mm^2

Authors have used a variety of resolutions for their studies ranging from a low resolution image of an entire plant to a high resolution image of a single leaf. It may be obvious that the higher the resolution the better, however, it is really the spatial resolution that is important, the physical area represented by a single pixel. One author considers the trade-offs of high spatial resolution and the resulting sacrifice of field of view (FOV) [38]. Having a high spatial resolution, usually on the order of microns, allows more detail in the resulting image of color gradients, textures, and shape, however, if the field of view is too constricted the subject may not entirely fit and with the case of plant studies, the subject may grow out of frame.

Not all authors list all camera features, but a general idea of image quality can still be gathered. Authors who were doing studies on single leaves often had high spatial resolutions. Most were on the order of 10^{-3} mm^2 ([10, 40, 41, 48]), [15] was on the order of 10^{-4} mm^2, and [9] was on the order of 10^{-2} mm^2. These authors used the large resolutions of their

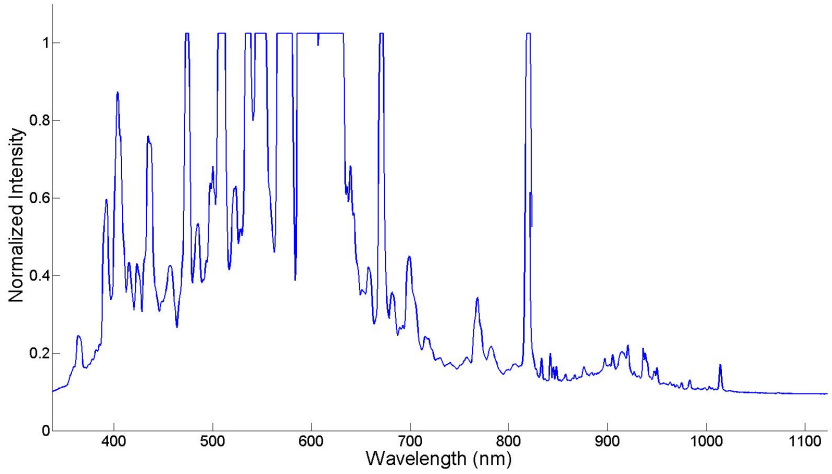
cameras to obtain detailed images of individual leaves filling at least half the image. This produces images with finer details on the outside shape of the leaves, the diseased blotches, or the insects residing on them. This level of resolution would certainly be useful for obtaining water stress information, however, is unpractical for several reasons. Namely, each leaf was severed from the plant before study, it would be impractical in future field use of the current work, and the cameras chosen do not have the resolution and focal length needed for this high of spatial resolution.

Closer related to the current work, several authors used a top-down view of the entire plant for their study. These authors generally had a spacial resolution on the order of 10^{-1} mm^2 and the images typically contained small plants filling under a third of each image [16, 17, 19, 21]. Another author, [22], had a spacial resolution on the order of 10^2 mm^2. The typical goal of these papers were to measure crop growth or simply image segmentation, so a finely detailed image was not typically needed.

A single author, [20], collected images with a profile view of the plant and was similar in construction to the current work. They reported each image to be approximately 400 kB. This is three times the size of the near infrared image created by the JAI and therefore must have a spatial resolution greater than the current study.

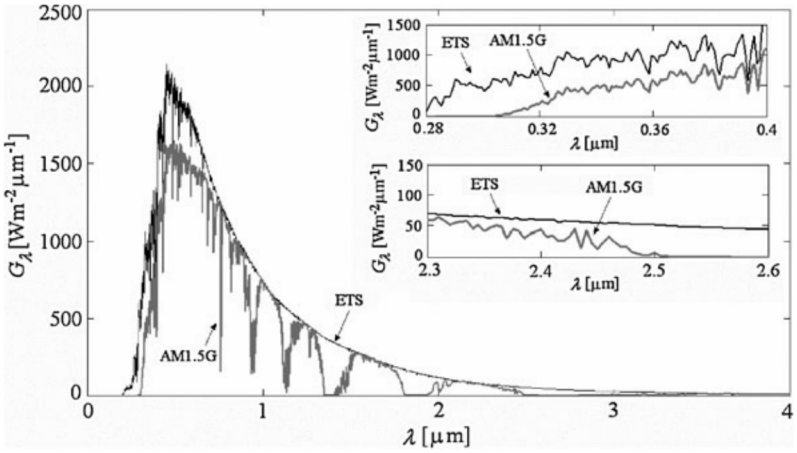
Many authors chose to have calibration boards in all their images for intensity and spatial calibration [9, 15, 16]. This allows consistent relation between pixel values and the physical variables between all the images. This is more difficult or even impossible for a multispectral study however. A calibration card would be needed for each spectrum, including a consistent heat sources for the longwave infrared, additionally the GOBI camera cannot be calibrated. In place of this, the images were collected in a consistent lighting and time of day. It was assumed the air in the greenhouse was temperature controlled and the images were collected at night so the artificial lighting of the greenhouse could be used for consistency. A spectrometer was used to record the emission of the greenhouse light and the result is shown in Figure 4.10.

Figure 4.10: Ambient Light For Experiments



This figure shows that most of the light produced by the greenhouse is in the visible spectrum but also is spread into the near and shortwave infrared, although not near the amount from sunlight. For comparison, the spectrum provided by the sun is shown in Figure 4.11.

Figure 4.11: Ambient Light From the Sun. M Paulescu, E Tulcan-Paulescu, P Gravila, and V Badescu. *Solar Radiation Measurements*. Green Energy and Technology. Springer, London, xviii edition, 2013. [Under Fair Use, 2014].



4.4.3 Stress Analysis

From the data collected there are 72 images for each of the ten days collected. From these 720 images, a variety of features will be tracked over the days and compared against the

control group. The goal is to reveal trends in the feature set that could be used to determine the stress state of the plant.

For the entire Stress Analysis section only the pixels representing the plant are considered. The masks generated by the best neural network approach are used to segment the images.

Tracked Features

The image features tracked over the ten days will largely be similar to those used for segmentation with a few exceptions. Namely, the values of the RGB, HSV, near infrared, and shortwave infrared will be tracked as well as the indices including ExG, NDI, NDVI, and Marchant (a summary of these indices is shown in Table 4.3). In addition to these, textons will be used to observe any change in texture (see Section 3.2.5).

The pixel values and indices will be observed through the mean for each image. More specifically, all the values will be determined for every pixel in the image and then sorted by plant and not plant. The values corresponding to the plant will be averaged and this mean will be the tracked value to represent the image.

Textons, as described by an author in 1991, are how humans see texture and are able to distinguish patterns within a surface [53]. Texture information was sought after in this study because it was the most clear change for the naked eye to see as the plant wilts. Textons were put into use as a computer vision algorithm in the late nineties [61], and the idea behind them is to develop a filter bank describing the texture information of different classes.

This starts with choosing a filter bank. The resulting textons are essentially linear combinations of the filters in the original bank. For the current work the Schmid set was chosen. It offers Gabor-like filters that are linearly independent and rotationally invariant.

Textons are then generated using k-means clustering for each day in the set. Namely, for each day, stress type, and plant type 20 textons are generated. Since the goal of the study is to obtain days of stress, the textons resulting from all the days are combined to make a total texton dictionary containing 200 textons for each plant and stress type. The number 200 was selected based on the results found by [54].

More specifically, a random selection of pixels from all the images in the subset in question (of day, plant, and stress) are filtered and their responses noted. In total, 2% of relevant pixels in each image (around one thousand) are used and they are only collected from plant pixels. K-means clustering is computed with K equal to 20 on these responses and the centers are recorded as textons.

The resulting texton dictionary is used to classify all the plant images of the subset. For an example image, this is done by convolving the Schmid set over the images and for each pixel finding the k-means center closest to the new result. A tally of each center is counted and the resulting histogram is used to represent the image.

The above method is as done by [54], however the current work takes it further. The experiment has the prior knowledge that as a plant wilts (or dies due to hypoxia) the histogram describing the image should shift more to the right; towards the classes associated with more days of stress. Thus the statistics of mean, median, standard deviation, and skew describing the histogram are compared across the images.

Additional to each class being described by the texton dictionary generated by the same class, the control classes are described by the dictionaries of their corresponding stressed classes. This, along with the control classes being described by their texton dictionaries, is used to the resulting trends are due to the plant dying.

Chapter 5

Results and Analysis

The experiments in designed in Chapter 4 are carried out and their results are detailed in the current chapter. Namely, segmentation accuracies, registration accuracies, and stressed plant image feature trends are presented.

5.0 Chapter Outline

The sections of this chapter follow the three main thrusts of the current work. The accuracies of several methods of plant image segmentation are presented and analyzed in Section 5.1. The failure modes of example methods are presented as well.

Next, the results of the experiments dealing with multispectral image registration are discussed in Section 5.2, such as point matching and image warping.

Lastly in Section 5.3, the trends of image features generated from the stressed plants are shown, including pixel intensities, vegetation indices, and texton descriptors.

5.1 Segmentation Methods

Two branches of segmentation methods are shown: thresholds and machine learning. The first deals with the simple band passing of pixel intensities and vegetation indices, and the second involves results of algorithms trained from image examples.

5.1.1 Thresholds and Indices

Beginning with the most basic approach, the six example images are segmented using previously mentioned vegetation indices and pixel values. A large range of lower and upper thresholds are applied to the images and the results were compared to the hand labeled examples. The results of these segmentation methods are shown in Table 5.1 and in order of their resulting f-score.

Table 5.1: F-Scores of Thresholds Tested for Segmentation

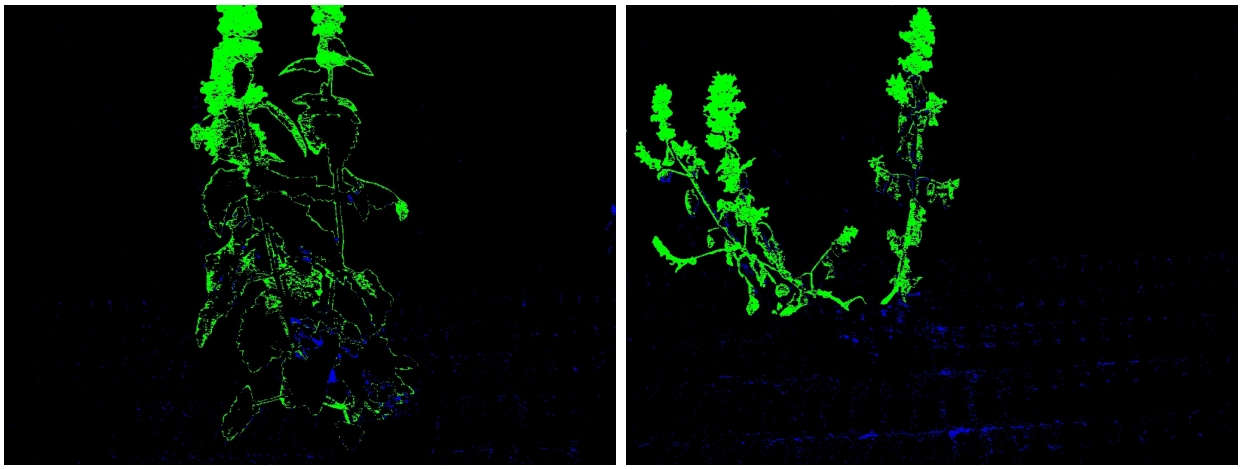
Feature	f-score
Marchant	**0.741**
Exg	**0.611**
Saturation (HSV)	**0.574**
RGB	**0.389**
Hue (HSV)	**0.435**
Value (HSV)	**0.326**
Green (RGB)	**0.320**
NDI	**0.278**
Red (RGB)	**0.277**
NDVI	**0.241**
Blue (RGB)	**0.146**

These results show that the Marchant index clearly outperforms the others with the dataset of the current study. It is important to note, however, that this does not generalize well to other datasets. As mentioned by other authors, change in lighting or cameras can change the thresholding values. This is particularly clear with the results of NDVI, a well known and commonly used method of plant segmentation, being at the bottom of the list. This strange result emphasizes the importance of tuning computer vision algorithms to a case-by-case basis.

Moreover, the poor performance of NDI and NDVI is likely due to the structure of the background of this dataset. Namely, the background has a red plane intensity on par with that of the plant as well as similar green and near infrared reflectivity. If the plants were the foreground of another material such as soil these would have performed better.

The segmentation generated using the Marchant index and, in Figure 5.1, is compared to the hand labeled version by presenting the difference between the two. Namely, any pixel in disagreement is presented as white while the black pixels correspond to the same labeling.

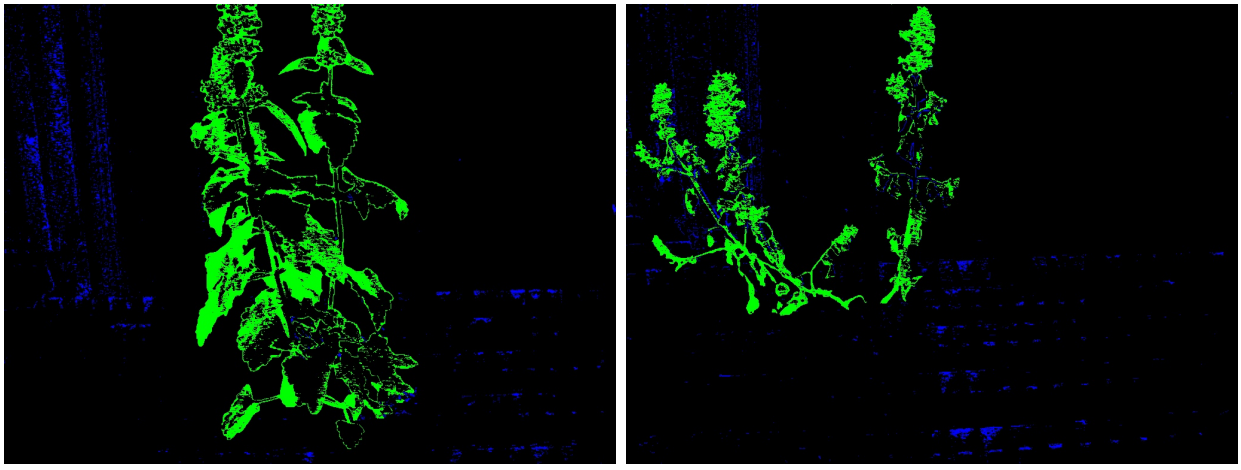
Figure 5.1: Example Marchant Threshold Segmentation Failure



This shows that the Marchant index largely fails on the white flowers of the catmint plant. This is not a surprise since, as most of the vegetation indices, the Marchant index is designed for green plants. Otherwise, the segmentation fails around the edges of the plant. It is possible this is due to incorrect hand labeling. The last notable failure is speckled around the gardening table the plant is sitting on. It is also green, so this is not surprising, but this failure could be removed through erosion in post-processing.

Figure 5.2 is the ExG segmentation failure and similarly constructed as Figure 5.1 but with the ExG as the index used. This segmentation mostly fails on the darker regions on the plant. Similar to the Marchant, it also fails on the white flowers and around the table, but it also incorrectly identifies pixels of the window pane.

Figure 5.2: Example ExG Threshold Segmentation Failure



Example segmentation failures of the remainder of the thresholds are shown in Appendix A.

5.1.2 Machine Learning

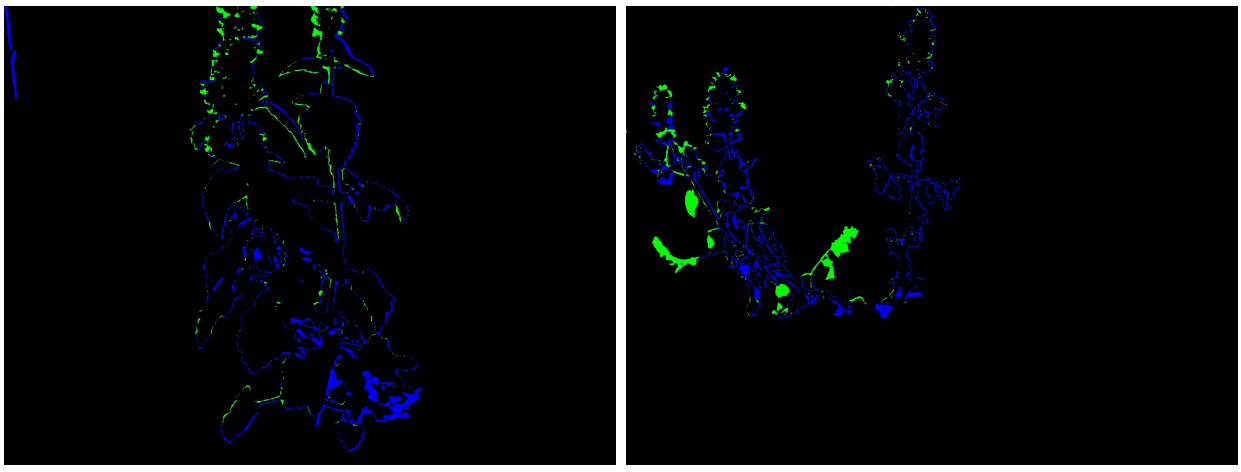
The methods tested here are varied as well as the parameters within. The first results listed, in Table 5.2, show how a neural network behave with different input feature groups and an example result is shown in Figure 5.3. Several NN structures are tested and only the best f-score is listed.

Table 5.2: F-Scores of Neural Network Parameters Tested for Segmentation

Feature Selection	Input Layer	First Hidden Layer	Second Hidden Layer	f-score
All Features	**11**	**11**,15,20,25	**0**,5,10,15,20,25	**0.972**
JAI Pixels	**4**	4,15,**20**,25	0,5,10,**15**,20,25	**0.969**
Visual Features	**9**	9,15,20,**25**	**0**,5,10,15,20,25	**0.893**
Visual Pixels	**3**	3,**15**,20,25	0,5,**10**,15,20,25	**0.888**

The powerful attribute of neural networks is their ability to create nonlinear relationships between features. This is evident in that the networks using only the raw pixels of the two cameras were close to the performance of that will all the features. The same can be said about the networks using the visual pixels and those with all visual features.

Figure 5.3: Example NN Segmentation Failure



The better performance of the networks given the features can be explained by the features giving the networks a step in the right direction before training begins. Neural networks are trained one example at a time, and it is likely that with more training the pixel-only networks could have performed just as well as their feature driven relatives. The same can be seen in Table 5.3 containing the SVM segmentation results.

Table 5.3: F-Scores of Support Vector Machine Parameters Tested for Segmentation

Feature Selection	Input Features	Kernel Type	f-score
All Features	**11**	**RBF**	**0.972**
JAI Pixels	**4**	**RBF**	**0.968**
Visual Features	**9**	**RBF**	**0.790**
Visual Pixels	**3**	**RBF**	**0.722**

The ν and γ parameters of the SVM was varied as well as the input features. The best f-score for each set of features is shown in Table 5.3. Overall, the SVM performed nearly as well as the NN given the same input features and typically outperformed the thresholding methods.

5.2 Multispectral Image Registration

5.2.1 Image Warping

The image warping methods of homography and TPS are tested using hand labeled points. The resulting images are visually investigated.

Figure 5.4: Homographic Transformation Example

The homography warping algorithm works very well as long as everything in the scene is at least a certain distance away, as discussed later, however, if there is too much disparity between the two images the results are not satisfactory. A failure case of this method is shown in Figure 5.4.

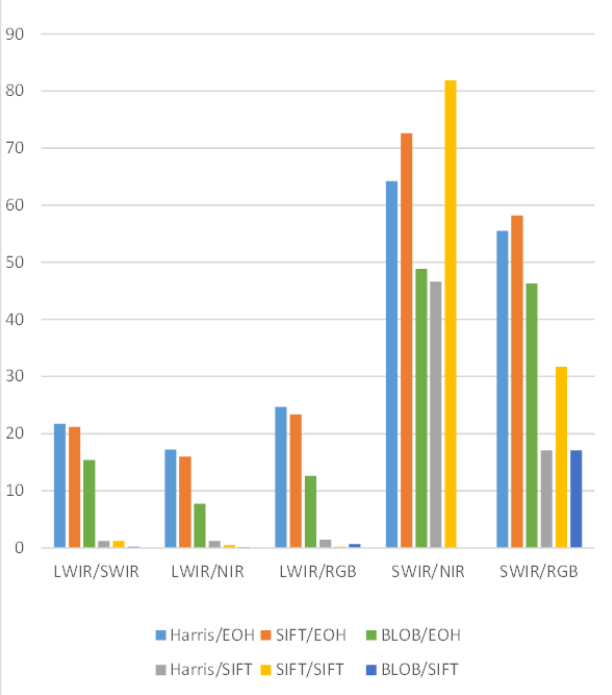
The TPS warping method needs an immense number of matched points for reasonable use. Using the same points and images as for Figure 5.4, TPS is used and the results can be seen in Figure 5.5. It can be seen that where many features are available, such as the man inside the barn, the warping works OK, but where there is not, such as the sky, there is a large amount of error.

Figure 5.5: Thin Plate Splines Transformation Example

5.2.2 Point Matching Results

As seen in Figure 5.6, the EOH method outperformed SIFT in most pairs of images. Most notably, SIFT produces around 1% accuracy for matching LWIR to the other spectral bands, while EOH is about 20% accurate. This 20% accuracy would allow for an algorithm such as RANSAC to be used to find the true matches.

Figure 5.6: Accuracies (%) of Registration Methods



With only one exception, SWIR/NIR, the EOH descriptors always outperformed that of SIFT in feature matching. Ultimately, however, the algorithm is not accurate enough to properly align the LWIR images. They are not used in the rest of the study.

5.2.3 Current Study Method

For matching SWIR to NIR for the current work, even though SIFT had a matching accuracy of 80%, not enough points could be obtained for a proper warping. Another method was developed that is specific to the current work. Specifically, the images are resized and cropped using the described SS method, and then the plant segmentation is matched to the SWIR image using template matching. This works well since the plant is much brighter in the SWIR than the background.

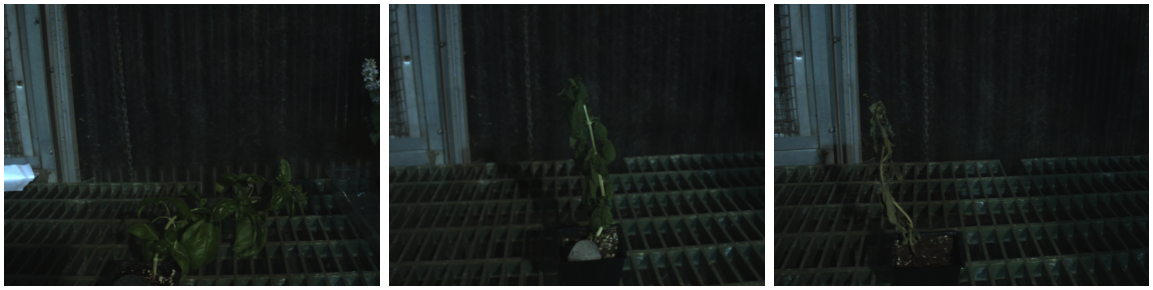
5.3 Plant Health Classification

A dataset of images are collected of several plants under varying stress conditions. The reaction of the plants to the stresses is discussed followed by a tracking of image features during this reaction.

5.3.1 The Stressed Plants

In total, there are six groups of plants: half are basil and half are catmint, and each half is split into three groups, under watered (drought), over watered (hypoxia), and normally watered (control). With the exception of control, a few example images were chosen from each subset to show the change in the plant over time. The first images shown, in Figure 5.7, represent the basil plant under drought conditions.

Figure 5.7: Example Dehydrated Basil Images



(a) Early (b) Middle (c) Late

The basil plant starts with full, bright green leaves in Figure 5.7a and is a healthy plant within 24 hours of being watered. The second image, in Figure 5.7b, was taken several days into the experiment and the plant now has smaller leaves with a less saturated color. Lastly, Figure 5.7c shows the plant near the end of the experiment with shriveled, more brownish leaves.

From these example images the human eye can discern how unhealthy the plant becomes. This is especially true with the size of the leaves, their color, and their texture. Note, however, that the overall direction of the stem of the plant is not a good indicator of plant health since the latter two stand erect while the image from the first day is toppled over.

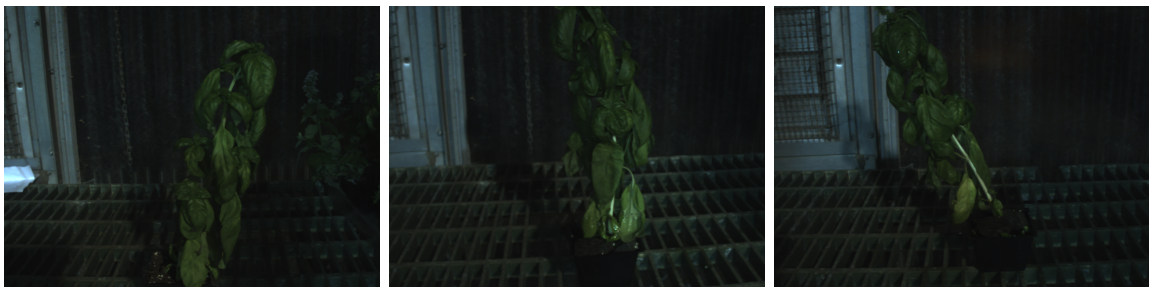
The next group, shown in Figure 5.8, are examples of a catmint plant under drought conditions. Many of the same things as said about the basil images can be said about that of the catmint. The plant begins with healthy, green leaves and over time they become shriveled and dull in color.

Figure 5.8: Example Dehydrated Catmint Images

(a) Early (b) Middle (c) Late

The next set contains example images from basil in over watered conditions and are shown in Figure 5.9. The difference in these images are difficult to see with the naked eye. The next set of plants, in Figure 5.10, however, do show a visible reaction to too much water. They show results similar to the their drought cousins, although it takes more days for the effect to set in.

Figure 5.9: Example Hypoxia Basil Images



(a) Early (b) Middle (c) Late

Figure 5.10: Example Hypoxia Catmint Images

(a) Early (b) Middle (c) Late

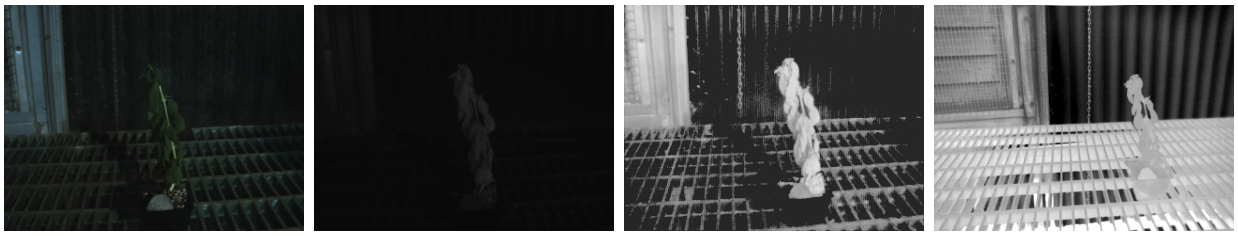
The control group, of course, all look similar to the first days of the figures above. The plants in this group do not noticeably change over the thirteen days.

5.3.2 Image Construction

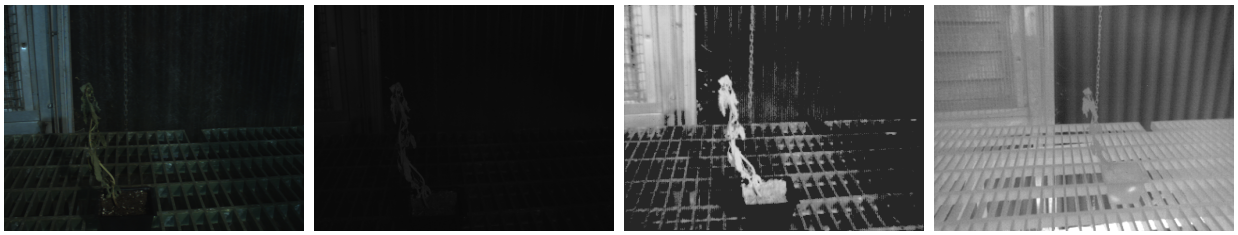
Example images resulting from the data collection are shown in Figure 5.11. The order of the camera spectrum from the left are visual, near infrared, shortwave infrared, and longwave infrared. Images from three different days are shown to provide an idea how the plant changes in the additional spectrum in the underwatered group. More images are placed in the Appendix to show more specifically how each plant and watering group behaves in all the images.

Figure 5.11: Example Multispectral Images

(a) Early



(b) Middle



(c) Late

Starting from the left, the visual images are darker than expected and the background contains very similar hues and tones. The plant is often either centered in the image or on the left side since the visual camera is the rightmost on the rig. The same can be said about the near infrared. Occasionally, another plant slid into frame and this was accounted for with manual editing of the images.

The near infrared images have similar construction to visual. They are especially dark due to the light source shown in Figure 4.10, however there is enough contrast for a computer vision algorithm. One can see that the plant is brighter than the background in NIR wit the exception of the window pane on the right side of the images.

The exposure of the shortwave camera was tuned to the point where texture on the plant could be seen. This results in a very low exposure that caused a grainy quality of the image. One can see again that the plant is bright compared to the background, but again the window pane is the exception.

The longwave images show an abundance of thermal information about the scene, but not very much about the plant temporally. The main background is a thin wall that leads to

the outside, winter air and so is normally dark in the thermal image. In the other direction, the pipes underneath the table contain hot water and are bright in every image. These two items in the background causes the intensity width, a parameter not manually controlled, to increase resulting in little intensity resolution of the plant. Additionally, the shown images are post fisheye correction.

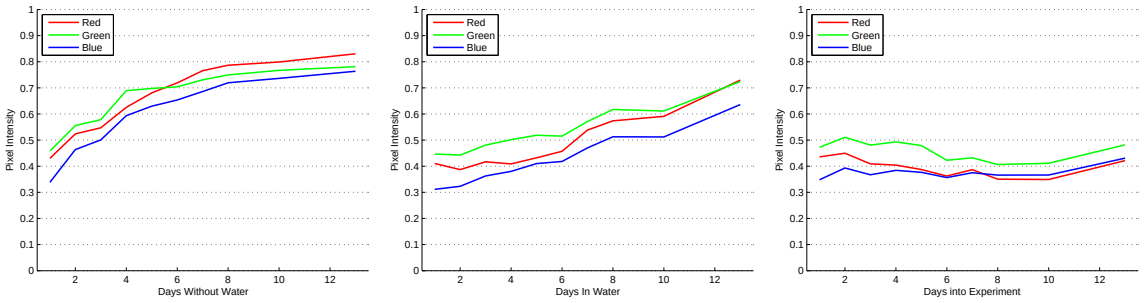
5.3.3 Stress Analysis

For each subset of plants (droughted catmint, control basil, etc.) there is a resulting 12 images a day for 10 days. The features are tracked over time in order to reveal trends corresponding to the health of the plants.

Simple Features

Beginning with the simplest of features trends can already be seen. The red, green, and blue (RGB) pixel intensities are plotted for all three catmint groups in Figure 5.12. From first glance, for both stresses, all three intensities trend upward while that of the control group remains level.

Figure 5.12: Mean of RGB Values with Catmint

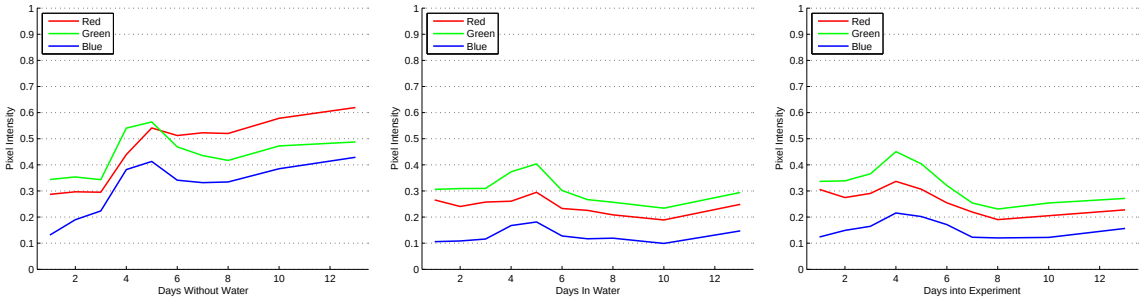


(a) Drought Group (b) Hypoxia Group (c) Control Group

For the drought catmint all three intensities sharply rise and then begin to level off. This agrees with the state of the plant since it too quickly wilted due to dehydration and then slowly became more brittle and dead. The hypoxia condition catmint, however, slowly wilted over the days and this can also be seen in the graphs.

The equivalent plots for basil are presented in Figure 5.13. The same trend of quickly rising and tapering off can be seen in the drought condition and this again agrees with the physical state of the plants. The same can be said about the hypoxia group since this group of plants never showed any sign of stress as exampled in Figure 5.9.

Figure 5.13: Mean of RGB Values with Basil



(a) Drought Group (b) Hypoxia Group (c) Control Group

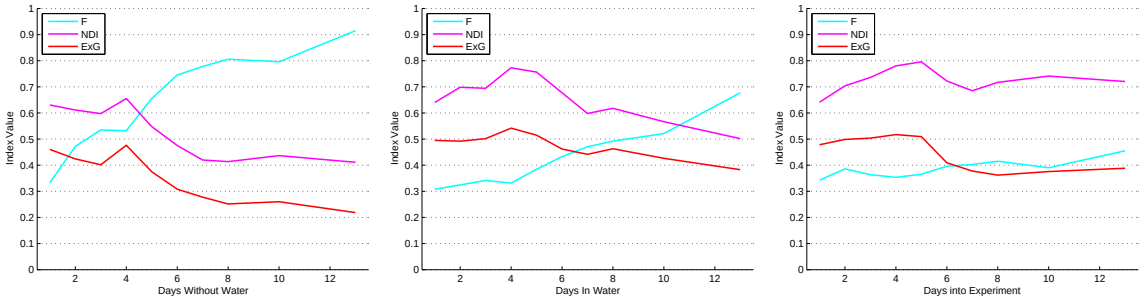
A phenomenon exists in Figure 5.13, however, that cannot be explained. The pixel intensities for days 4 and 5 are much higher than the trend. This is true for all three stress conditions of the plant, which would lead one to question the camera parameters, but these were unchanged through each day and even through the catmint images that resulted in days 4 and 5 for Figure 5.12.

The next group of plots, tracking catmint in Figure 5.14, represent the indices that can be derived from visual cameras, namely Excess Green (ExG), the Normalized Difference Index (NDI), and the Marchant index.

The ExG and NDI indices tend to follow each other in that for the drought condition there is a sharp downward trend and little trend in hypoxia and control conditions.

The Marchant index seems to be a good predictor of an under watered plant with the average going nearly from its minimum to its maximum over the thirteen days. The same can be said about the hypoxia condition graph, and, although a slower ascent, this closely matches the wilting pattern of the plant.

Figure 5.14: Mean of Visual Index Values with Catmint

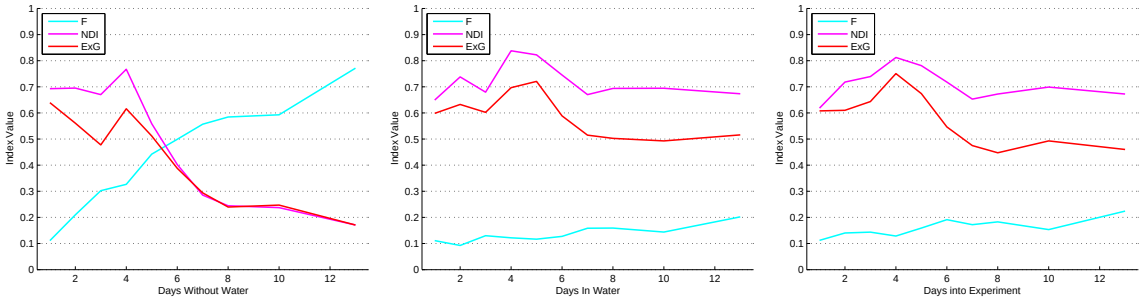


(a) Drought Group (b) Hypoxia Group (c) Control Group

The basil version of these graphs are shown in Figure 5.15. The indices of ExG and NDI are

very similar to those for the catmint cases and the Marchant for the drought condition as well. Again, since the basil plant did not react to over watering there is very little to see in the hypoxia graph.

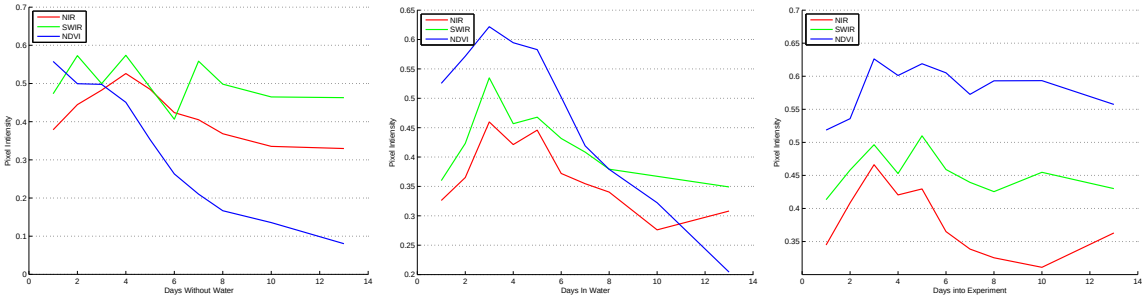
Figure 5.15: Mean of Visual Index Values with Basil



(a) Drought Group (b) Hypoxia Group (c) Control Group

The next two figures, 5.16 and 5.17, show pixel intensities of the NIR and SWIR cameras, as well as the Normalized Difference Vegetation Index (NDVI), tracked over the thirteen days. The multispectral intensities alone do not show any trends different from that of the control group. The NDVI value, however, has a very steep downward slope for the catmint drought and hypoxia as well as the drought basil.

Figure 5.16: Mean of Multispectral Values and Indices with Catmint

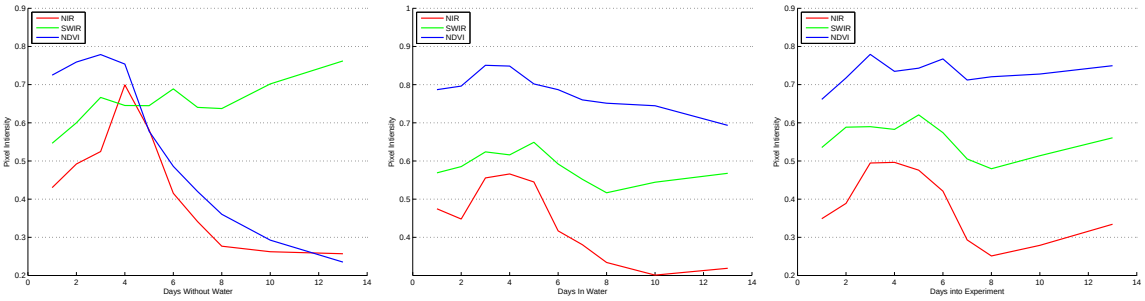


(a) Drought Group (b) Hypoxia Group (c) Control Group

The NDVI result is especially interesting for the hypoxia conditions. Since the stress in this subset is difficult to see visually and responses have been relatively weak in other tracked features, it could be considered a valuable tool for measuring the over watered stress in plants.

While 12 images a day is not enough to train a robot to recognize the stress of a plant, these graphs do reveal the potential of even these simple measurements.

Figure 5.17: Mean of Multispectral Values and Indices with Basil

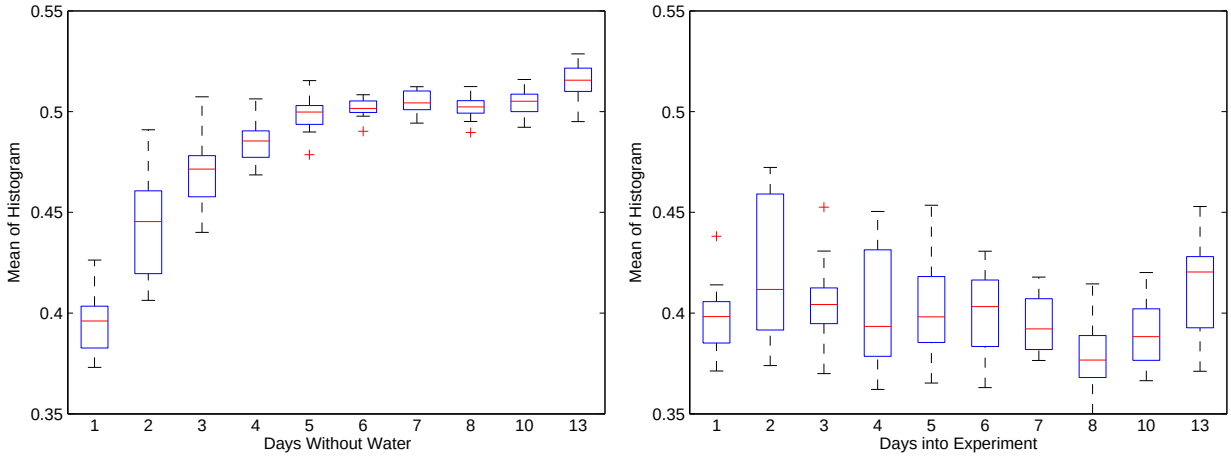


(a) Drought Group (b) Hypoxia Group (c) Control Group

Texton Analysis

From each class of stress, a texton dictionary is generated. Using each dictionary, every related class is described in order to find trends in the resulting histograms. The first shown, in Figure 5.18 for example, is the under watered and control groups of catmint being described by the dictionary generated using the former class.

Figure 5.18: Mean of Drought Texton Histograms with Catmint



(a) Drought Group (b) Control Group

The plots in Figure 5.18 show how the mean value in the histograms describing each day's plants shift along with the days. Note that the mean has been normalized by dividing by the total number of textons, 200. This means that a value near zero is more associated with a recently watered plant, while a value closer to one has been stressed.

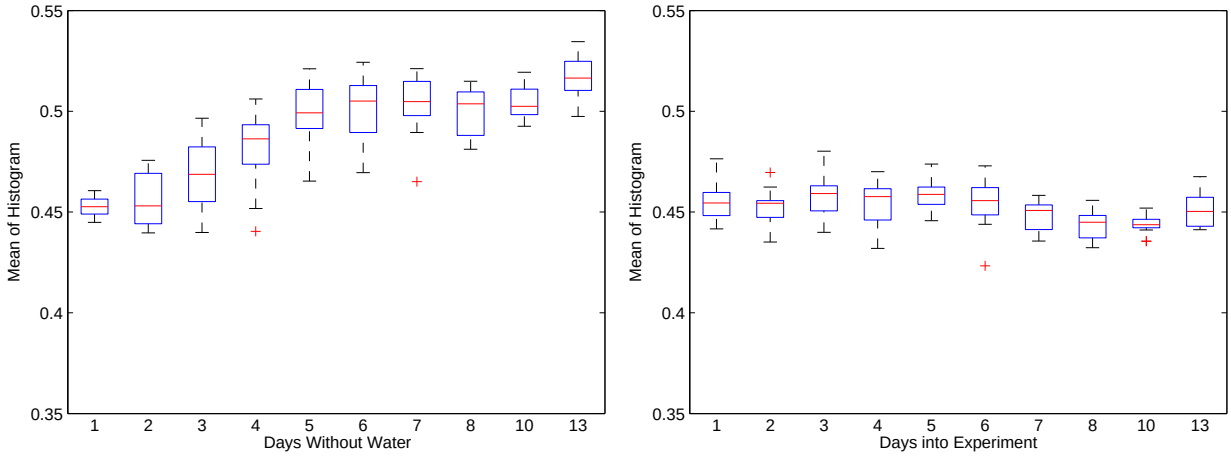
The mean values of the early catmint plants are near 0.4, quickly level out around 0.5, and has a greater variance around the first few days. This data agrees with what happened with

the plants, as they started healthy and then wilted at different rates, but all finally being dead after about a week without being watered. Furthermore, the same dictionary is used on the control group and it shows the means of plants that have been recently watered tend to stay near a 0.4 value.

The same information can be obtained from the skew of the histograms. If a plant has been recently watered, it should be skewed more to the left side of the histogram while a wilted plant should tend to the right. These can be seen in Appendix B.

The same experiments are done with the basil classes and the results are shown in Figure 5.19. The data again agrees with logic, however, not as much as with the catmint plants. The basil plants are more resilient to under watering than catmint which is shown in the plot as the slow upward slope. The test on the control group also agrees in that there is no upward slope in the data. It can be concluded, though, that there is simply not as much textural information in the wilting of basil as in that of catmint.

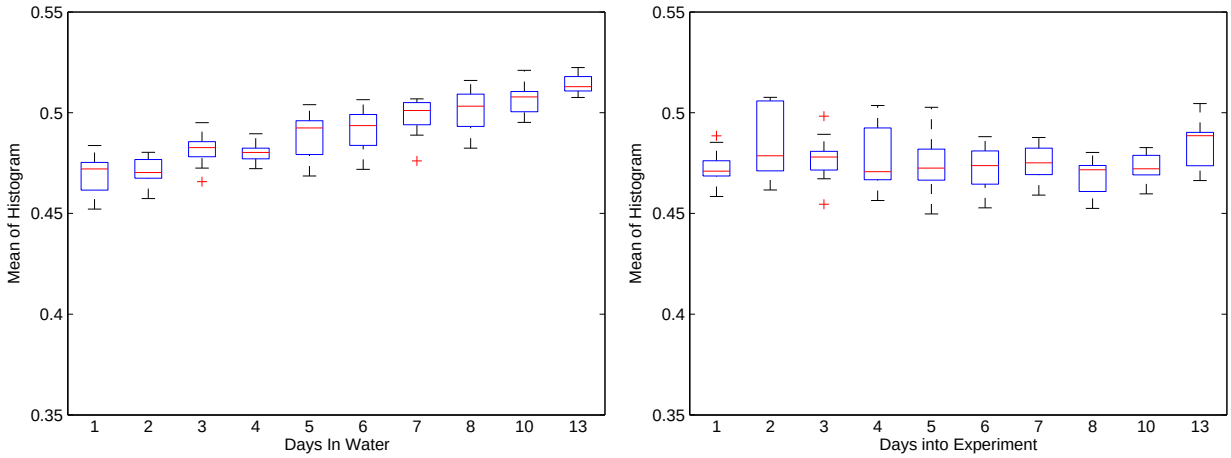
Figure 5.19: Mean of Drought Texton Histograms with Basil



(a) Drought Group

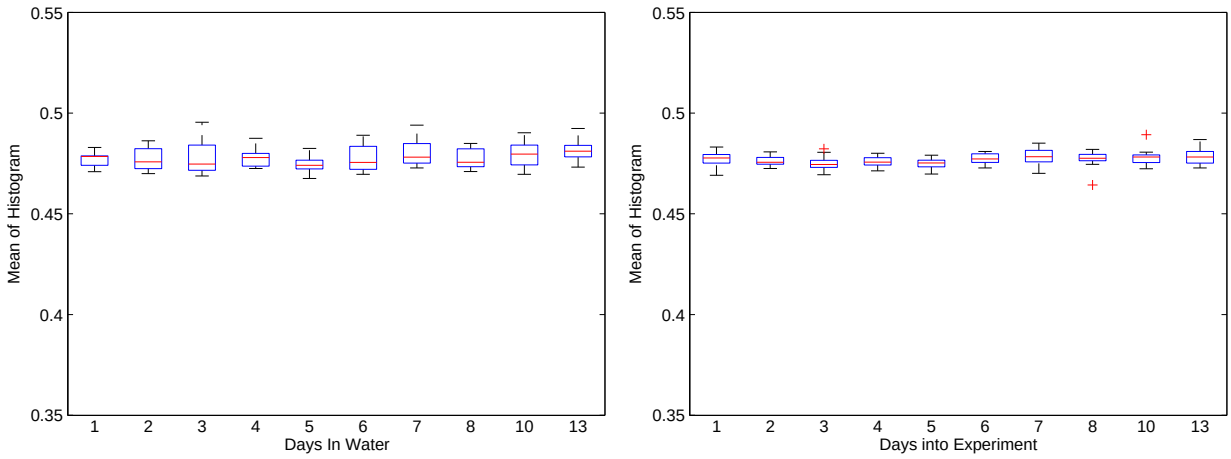
(b) Control Group

Figure 5.20: Mean of Hypoxia Texton Histograms with Catmint



(a) Hypoxia Group (b) Control Group

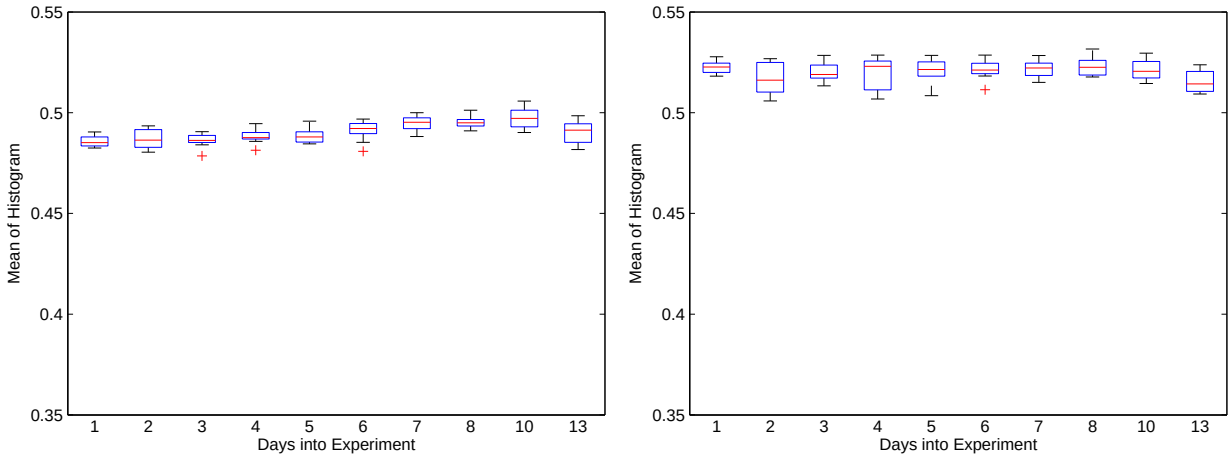
Figure 5.21: Mean of Hypoxia Texton Histograms with Basil



(a) Hypoxia Group (b) Control Group

Ideally, a collection of random textures or textures that do not change between classes would generate a texton dictionary that would result in no trends. More specifically, training a dictionary from the control group and using that dictionary on the same group should result in a random distribution histogram. This experiment is conducted and the trends of the mean and skew values are shown in Figures 5.22 and B.5, respectively.

Figure 5.22: Mean of Control Texton Histograms



(a) Catmint (b) Basil

The figures show that there are very little trends in the data resulting in similar histograms for all images in the set. Any trend that does exist, however, could be eliminated simply by collecting more data.

Chapter 6

Conclusions

The current work has the ultimate goal of reviewing the potential of multispectral computer vision in an agricultural setting. Within this, there were three main thrusts: segmenting plant images, registering the multispectral images, and finding image features that could reveal the stress state of a plant.

6.1 Key Conclusions

Segmenting Plant Images

Many authors have attempted to find the pixels of an image that correspond to a plant within it. Most authors have used simple thresholds on pixel intensities or vegetation indices, and some have ventured into machine learning. The current work seeks to validate their findings and find the method best fit for the current scenario.

An experiment was set up to test the accuracy of these methods by using a hand labeled ground truth. Six images were chosen for their representative nature of the dataset. Namely, these six images contained both types of plants and at various health levels, including very much wilted. This is significant as it has not been done by other authors and reveals the weakness of simple thresholding algorithms, all resulting in f-scores below 0.8. Again attributing to their weakness, this low scoring could be mitigated with a more controlled, structured environment.

The machine learning algorithms of neural networks and support vector machines, on the other hand, greatly outperformed the simple thresholds. Both methods were able to find the nonlinear relationships in the intensities that correspond to plant pixels and ultimately resulting in the highest score of 0.972. Their ability to find the relationships within the indices was proven by the small difference in score when adding the indices as features to

the training. Additionally, the value of the near infrared channel was shown by the increase of accuracy when its intensities were used.

Multispectral Image Registration

For the current study, the homography warping outperformed that of TPS, which required too many points to be matched between images. The EoH descriptors ultimately gives better matching between multispectral images than SIFT, but still does not result in enough points for the LWIR to be used. Additionally, a method specific to this study was developed and gave better results for matching the SWIR to the NIR.

Plant Stress Classification

The goal for this section was to find what information in visual pixel intensities, vegetation indices, and multispectral vision could be used to determine the stress of a plant. To this end, an experiment is conducted where images were collected over a span of thirteen days of plants under the stress conditions of drought and hypoxia as well as a normally watered control group.

One author has done a similar study on sunagoke moss and was able to find the level of water content based on textural information. The plants and resulting images of the current work, however, are very different and many of the tracked features were found to be useful.

The pixel intensities and vegetation indices reveals even the RGB values to be useful with clear upward trends resulting from the wilting plants. The Marchant index especially emphasized these trends since it is the linear combination of the former. The intensities from the NIR and SWIR cameras do not show great promise, however, this could be due to the quality of the cameras and the ambient lighting of the greenhouse. NDVI, though, does respond to a wilting plant.

Additional to these tracked features, textons are used to obtain textural information of the wilting plant. The resulting texton histograms are reviewed through the shifting of their means over the dataset and are found to be useful in determining the stress.

A wealth of information is retained within images of the plants. The current work shows that multispectral information can be helpful in tracking the stress of a plant, but it could be easily determined by visual information alone.

6.2 Future Work

Computer vision in agriculture is still a young field with boundless opportunities for future work. The current work could be expanded and improved in a variety of ways, but it also presents the reader with the ideas for many other related projects.

Improvement of the Current Work

The first and most basic improvement to the current work is more data. The collected images for this study were few in number and all similar in nature. More specifically, the stress classification section of the work could be expanded to ML algorithms if more examples existed per class (more than twelve samples for a single class in other words). More importantly, the dataset should be generalized to different lighting conditions, backgrounds, plant ages, camera angles, and depths. A well rounded set would assure the trends found could be associated with the stress state of the plant.

Many algorithms could also be added to the study. This could be as simply as adding textons to the image segmentation feature set or as complex as using bio-inspired feature selection algorithms as in [21] to more scientifically determine which features are important. Co-occurrence matrices as used in [9], for example, could be employed to find additional textural information in all areas of the study.

The segmentation study specifically could be improved by adding features, such as gradients, area means, energy, or something more complex. Additionally, an attempt could be made towards creating a vegetation index from the results found. The plant health study, on the other hand, could be expanded to include morphological features of the plant. The resolutions of the incoming images could be varied as well to find the what resolutions are needed, the spatial resolutions are needed, and ultimately how close the cameras need to be to the plants.

More specifically to the current work, many changes could be made to improve the results. Foremost, the image construction could be better designed with better lighting (particularly in the NIR and SWIR), backgrounds more natural to farming such as soil, and additional frames captured from angles including top-down. Calibration in intensity could have been employed for more accurate readings as well as calibration in the spatial sense or easier image registration. As a small note, the measurements could be extended for the hypoxia condition basil plants since they did not wilt, and largely the image registration requires more work.

Better results could have been obtained also through better cameras. A calibrated LWIR camera would vastly improve the readings. The resolutions of the LWIR and SWIR cameras are small in comparison to that of the JAI, which inhibits proper registration and fine textural detail within the multispectral range. Even the JAI could be improved with a smaller focal length that would allow more texture information into the image.

Related Projects

While the experiment could be expanded through different stresses such as diseases, fungi, and pests, or through having a variety of plant species, the more interesting expansion is obtained by more cameras of different types.

Of course the work could be augmented by adding cameras of polarimeteric or hyperspectral nature similar to [11]. With this data, however, specific wavelengths could be targeted by the employment of feature selection methods on the hyperspectral bands as used in [21]. This would allow a scientist of agriculture to chose camera filters with pass-bands designed for their needs.

Another avenue worth exploring is the world of three dimensional analysis. Using stereo vision, LIDAR, an XBox Kinect, or even trinocular vision as in [39] could open up a vast amount of information to be obtained about a plant subject. Having a 3D model of a plant would allow the *absolute* measurements of leaf area, plant height, leaf count, and biomass. Image segmentation would be greatly improved and multiple plants within a scene could even be separated.

Alternatively, a stationary field camera could be constructed for constant monitoring. This would allow the close watching of individual plants and methods such as wilt tracking. This camera could easily also be used to measure the growth of the plants, watching for larger pests, and the localization of autonomous field equipment.

In relation to field autonomy, the data of the current work could be collected from an unmanned ground vehicle, resulting in a more relevant study. The dataset could then be expanded to use in studies involving fruit detection, localization, and vehicle row navigation. Additionally, if the field were equipped as in [3], the dataset could even be augmented with labeling in nutrients.

Bibliography

[1] Juan Ignacio Arribas, Gonzalo V. Sánchez-Ferrero, Gonzalo Ruiz-Ruiz, and Jaime Gómez-Gil. Leaf classification in sunflower crops by computer vision and neural networks. *Computers and Electronics in Agriculture*, 78(1):9–18, August 2011.

[2] Hermann Auernhammer. Precision farming the environmental challenge. *Computers and Electronics in Agriculture*, 30(1-3):31–43, February 2001.

[3] Jonathan Weekley, Joseph Gabbard, and Jerzy Nowak. Micro-Level Management of Agricultural Inputs: Emerging Approaches. *Agronomy*, 2(4):321–357, December 2012.

[4] Eva Boegh, H. Soegaard, N. Broge, C.B. Hasager, N.O. Jensen, K. Schelde, and A. Thomsen. Airborne multispectral data for quantifying leaf area index, nitrogen concentration, and photosynthetic efficiency in agriculture. *Remote Sensing of Environment*, 81(2):179–193, 2002.

[5] D.W. Lamb. The use of qualitative airborne multispectral imaging for managing agricultural crops a case study in south-eastern Australia. *Animal Production Science*, 40, 2000.

[6] N Aleixos, J Blasco, F Navarrón, and E Moltó. Multispectral inspection of citrus in real-time using machine vision and digital signal processors. *Computers and electronics in ...*, 33:121–137, 2002.

[7] C. L. McCarthy, N. H. Hancock, and S. R. Raine. Applied machine vision of plants: a review with implications for field deployment in automated farming operations. *Intelligent Service Robotics*, 3(4):209–217, August 2010.

[8] M. Guijarro, G. Pajares, I. Riomoros, P.J. Herrera, X.P. Burgos-Artizzu, and a. Ribeiro. Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, 75(1):75–83, January 2011.

[9] R. Pydipati, T.F. Burks, and W.S. Lee. Identification of citrus disease using color texture features and discriminant analysis. *Computers and Electronics in Agriculture*, 52(1-2):49–59, June 2006.

[10] M El-Helly, AA Rafea, and S El-Gammal. An Integrated Image Processing System for Leaf Disease Detection and Diagnosis. *IICAI*, 2003.

[11] Yunseop Kim, David M. Glenn, Johnny Park, Henry K. Ngugi, and Brian L. Lehman. Hyperspectral image analysis for water stress detection of apple trees. *Computers and Electronics in Agriculture*, 77(2):155–160, July 2011.

[12] J. Hemming and T. Rath. Computer-Vision-based Weed Identifcation under Field Conditions using Controlled Lighting. *Journal of Agricultural Engineering Research*, 78(3):233–243, March 2001.

[13] Tijmen Bakker, Kees Asselt, Jan Bontsema, Joachim Müller, and Gerrit Straten. Systematic design of an autonomous platform for robotic weeding. *Journal of Terramechanics*, 47(2):63–73, April 2010.

[14] F Tsai, EK Lin, and K Yoshino. Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species. *International Journal of Remote ...*, (June), 2007.

[15] Paul Boissard, Vincent Martin, and Sabine Moisan. A cognitive vision approach to early pest detection in greenhouse crops. *Computers and Electronics in Agriculture*, 62(2):81–93, July 2008.

[16] Guanghui Teng. Development of Non-Contact Measurement on Plant Growth in Greenhouses Using Machine Vision. *2003, Las Vegas, NV July 27-30, 2003*, 2003.

[17] Chuanyu Wang, Xinyu Guo, and Chunjiang Zhao. Detection of Corn Plant Population and Row Spacing Using Computer Vision. *2011 Second International Conference on Digital Manufacturing & Automation*, pages 405–408, August 2011.

[18] D. Wang and J. Gartung. Infrared canopy temperature of early-ripening peach trees under postharvest deficit irrigation. *Agricultural Water Management*, 97(11):1787–1794, November 2010.

[19] DS Shrestha and BL Steward. Automatic corn plant population measurement using machine vision. *Transactions of the ASAE*, 46(2):559–565, 2003.

[20] Jie Liang, Ali Zia, Jun Zhou, and Xavier Sirault. 3D Plant Modelling via Hyperspectral Imaging. *2013 IEEE International Conference on Computer Vision Workshops*, pages 172–177, December 2013.

[21] Yusuf Hendrawan and Haruhiko Murase. Bio-inspired feature selection to select informative image features for determining water content of cultured Sunagoke moss. *Expert Systems with Applications*, 38(11):14321–14335, May 2011.

[22] T. Kataoka, T. Kaneko, H. Okamoto, and S. Hata. Crop growth estimation system using machine vision. *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, 2(Aim):b1079–b1083, 2003.

[23] E.a. Moya, L.R. Barrales, and G.E. Apablaza. Assessment of the disease severity of squash powdery mildew through visual analysis, digital image analysis and validation of these methodologies. *Crop Protection*, 24(9):785–789, September 2005.

[24] D.M. Bulanon, T. Kataoka, Y. Ota, and T. Hiroma. A Segmentation Algorithm for the Automatic Recognition of Fuji Apples at Harvest. *Biosystems Engineering*, 83(4):405–412, December 2002.

[25] D M Bulanon, T F Burks, and V Alchanatis. Fruit Visibility Analysis for Robotic Citrus Harvesting. *American Society of Agricultural and Biological Engineers*, 52(1):277–283, 2009.

[26] R. C. Harrell, D. C. Slaughter, and P. D. Adsit. A Fruit-Tracking System for Robotic Harvesting. *Machine Vision and Applications*, 2(2):69–80, March 1989.

[27] D L Peterson. Harvest Mechanization Progress and Prospects for Fresh Market Quality Deciduous Tree Fruits. *HortTechnology*, 15(5):72–75, 2005.

[28] M W Hannan, T F Burks, and D M Bulanon. A Real-time Machine Vision Algorithm for Robotic Citrus. In *American Society of Agricultural and Biological Engineers Meeting*, volume 0300, 2007.

[29] R C Harrell, P D Adsit, T A Pool, and R Hoffman. The Florida Robotic Grove-Lab. *American Society of Agricultural and Biological Engineers*, 22(2):391–399, 1990.

[30] R.D. Tillett and B.G. Batchelor. An algorithm for locating mushrooms in a growing bed. *Computers and Electronics in Agriculture*, 6(3):191–200, December 1991.

[31] Jinlin Xue and Liming Xu. Autonomous Agricultural Robot and its Row Guidance. *2010 International Conference on Measuring Technology and Mechatronics Automation*, pages 725–729, March 2010.

[32] CE Schertz and GK Brown. Basic considerations in mechanizing citrus harvest. *TRANSACTIONS of the ASAE*, 11(2):343–384, 2002.

[33] EA Jr Parrish and AK Goksel. *TRANSACTIONS of the ASAE*, 20(5):822–827, 1997.

[34] RJ Haggar, CJ Stent, and S Isaac. *Journal of Agricultural Engineering Research*, 28:349–358, 1983.

[35] DE Guyer, GE Miles, MM Schreiber, OR Mitchell, and VC Vanderbilt. *Transactions of the ASAE*, 29(6).

[36] GE Meyer, A Stepanek, DP Shelton, and EC Dickey. *Transactions of the ASAE*, 31(3).

[37] GR Hack. The use of image processing under greenhouse conditions for growth and climate control. *Symposium on High Technology in Protected ...*, 1988.

[38] Edgar P Spalding. *Plant Systems Biology*, volume 553 of *Methods in Molecular Biology*. Humana Press, Totowa, NJ, 2009.

[39] Michael Nielsen. *Automatic Plant Annotation Using 3D Computer Vision*. PhD thesis, Aalborg University, 2011.

[40] Neeraj Kumar, PN Belhumeur, and Arijit Biswas. Leafsnap: A computer vision system for automatic plant species identification. *Computer VisionECCV*, pages 1–14, 2012.

[41] Shitala Prasad, Sateesh K. Peddoju, and Debashis Ghosh. Unsupervised resolution independent based natural plant leaf disease segmentation approach for mobile devices. *Proceedings of the 5th IBM Collaborative Academia Research Exchange Workshop on - I-CARE '13*, pages 1–4, 2013.

[42] DL Hernández-Rabadán. Method for Segmenting Tomato Plants in Uncontrolled Environments. *Scientific Research*, pages 599–606, 2012.

[43] Peter N Belhumeur, Daozheng Chen, Steven Feiner, David W Jacobs, and W John Kress. Searching the world's herbaria: A system for visual identification of plant species. *Computer VisionECCV ...*, 3:1–11, 2008.

[44] Floris De Smedt, I Billiauws, and Toon Goedemé. Neural networks and low-cost optical filters for plant segmentation. *International Journal of Computer Information Systems and Industrial Management Applications*, 3:804–811, 2011.

[45] DM Woebbecke and GE Meyer. Shape features for identifying young weeds using image analysis. *Transactions of the ...*, 38(l):271–281, 1995.

[46] DS Shrestha, BL Steward, and E Bartlett. Segmentation of Plant from Background Using Neural Network Approach. *Proc. of the Artificial Neural Networks in Engineering Conference*, (4-7):903–908, 2001.

[47] J Pastrana and T Rath. Detection of overlapping plants using Active Shape Models. *Herrenhauser*, 2012.

[48] Jyotismita Chaki and R Parekh. Plant Leaf Recognition using Shape based Features and Neural Network classifiers. *International Journal of Advanced ...*, 2(10):41–47, 2011.

[49] Cristhian Aguilera, Fernando Barrera, Felipe Lumbreras, Angel D Sappa, and Ricardo Toledo. Multispectral image feature points. *Sensors (Basel, Switzerland)*, 12(9):12661–72, January 2012.

[50] J Canny. A computational approach to edge detection. *IEEE transactions on pattern analysis and machine intelligence*, 8(6):679–98, June 1986.

[51] Glenn D. Hines, Zia-ur Rahman, Daniel J. Jobson, and Glenn a. Woodell. Multi-image registration for an enhanced vision system. In Zia-ur Rahman, Robert A. Schowengerdt, and Stephen E. Reichenbach, editors, *Visual Information Processing XII*, volume 5108, pages 231–241, August 2003.

[52] A. Ardeshir Goshtasby. *2-D and 3-D Image Registration: For medical, remote sensing, and industrial applications*.

[53] B Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290:91=97, 1981.

[54] Manik Varma and Andrew Zisserman. A Statistical Approach to Texture Classification from Single Images. 2004.

[55] JAI. *AD-080GE Digital 2CCD Pogressive Scan Multi-Spectral Camera User's Manual*, 6 2012. Ver. 1.1.

[56] Sensors Unlimited. *SU320HX-1.7RT Mil-Rugged High Sensitivity InGaAs SWIR Camera with Advanced Dynamic Range Enhancements*.

[57] Xenics Infrared Solutions. *Gobi-384 Industrial Brochure*. XB-015 Issue 3.

[58] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[59] HE Nilsson. Remote Sensing and Image Analysis in Plant Pathology. *Canadian Journal of Plant Pathology*, (15):489–527, 1995.

[60] M Paulescu, E Tulcan-Paulescu, P Gravila, and V Badescu. *Solar Radiation Measurements*. Green Energy and Technology. Springer, London, xviii edition, 2013.

[61] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and Texture Analysis for Image Segmentation. 43(1):7–27, 2001.

Appendix A

Segmentation Failures

Figure A.1: Example RGB Threshold Segmentation Failure

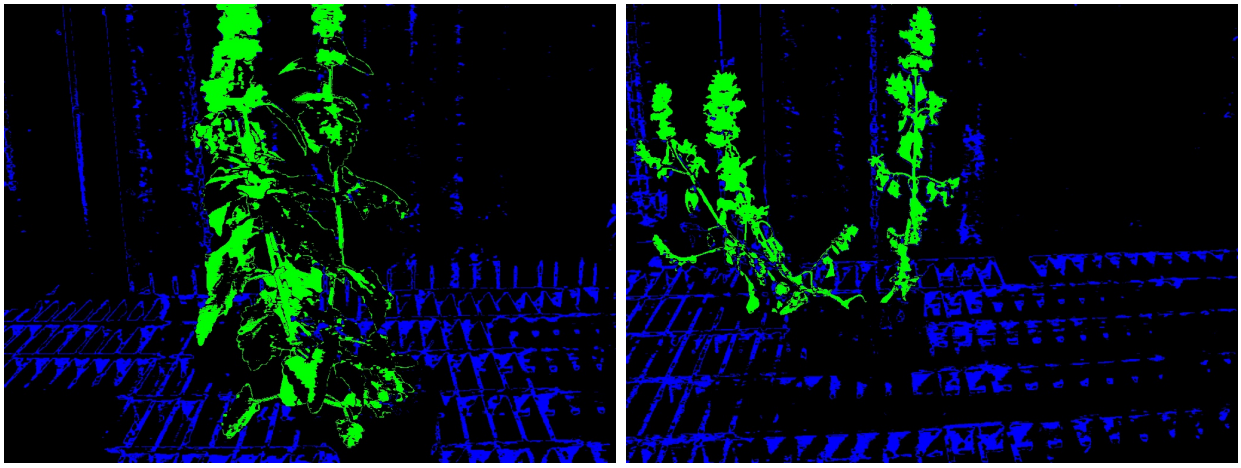


Figure A.2: Example Red Threshold Segmentation Failure

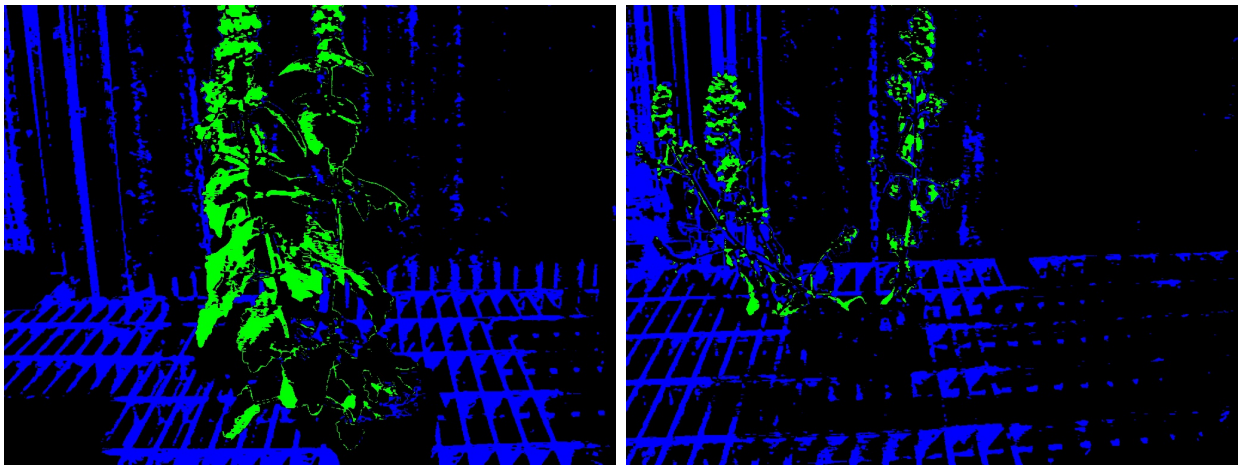


Figure A.3: Example Green Threshold Segmentation Failure

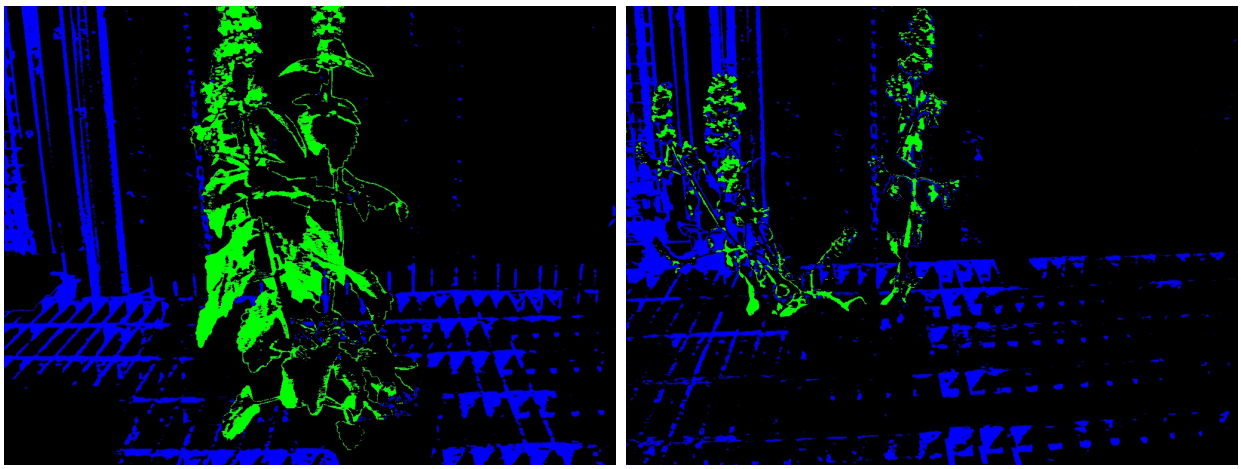


Figure A.4: Example Blue Threshold Segmentation Failure

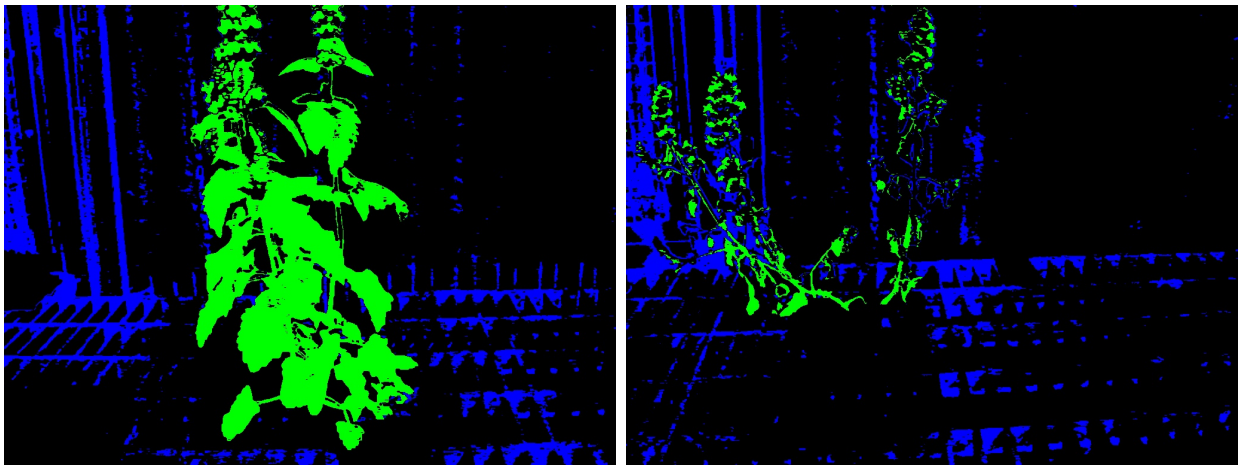


Figure A.5: Example Hue Threshold Segmentation Failure

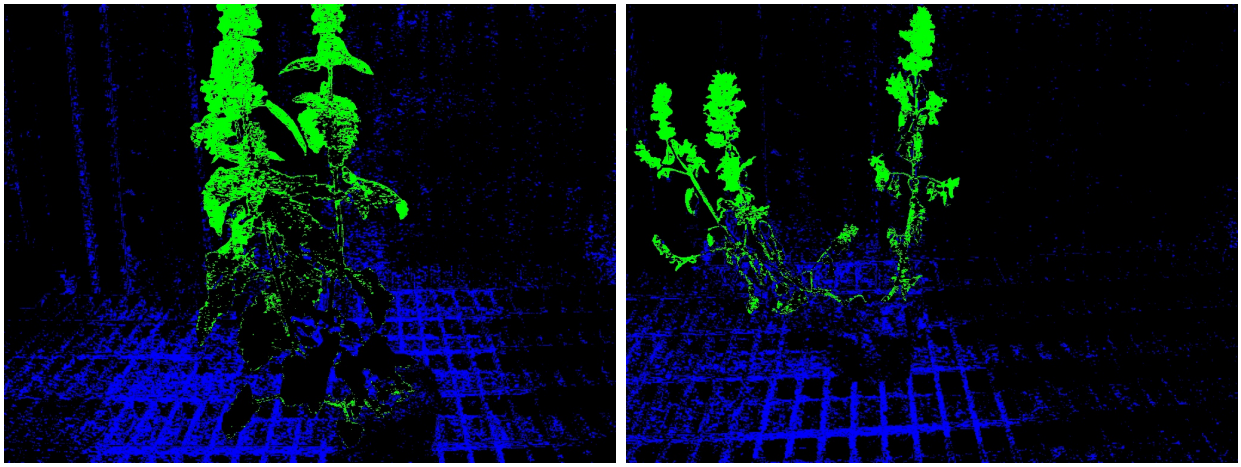


Figure A.6: Example Saturation Threshold Segmentation Failure

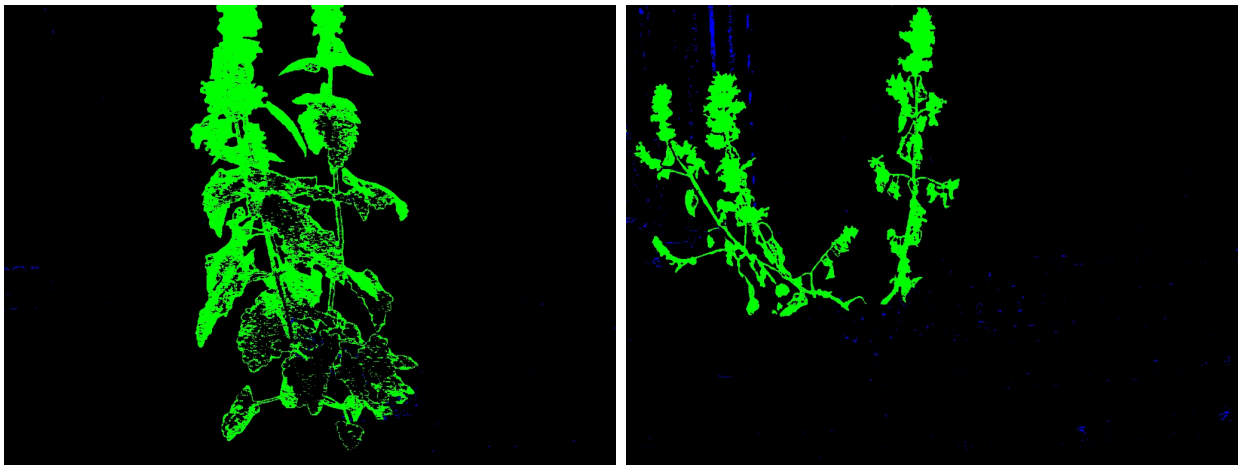


Figure A.7: Example Value Threshold Segmentation Failure

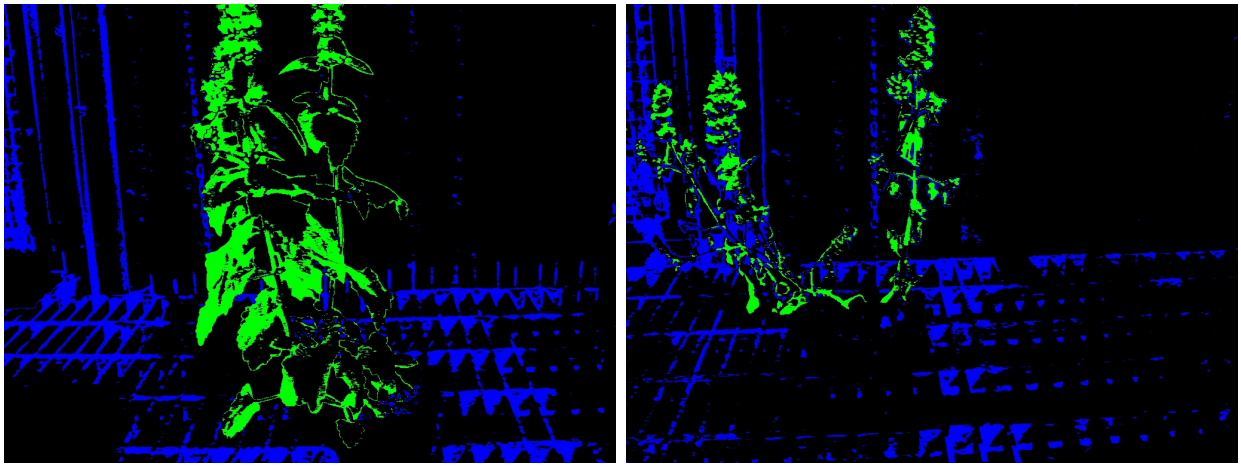


Figure A.8: Example NDI Threshold Segmentation Failure

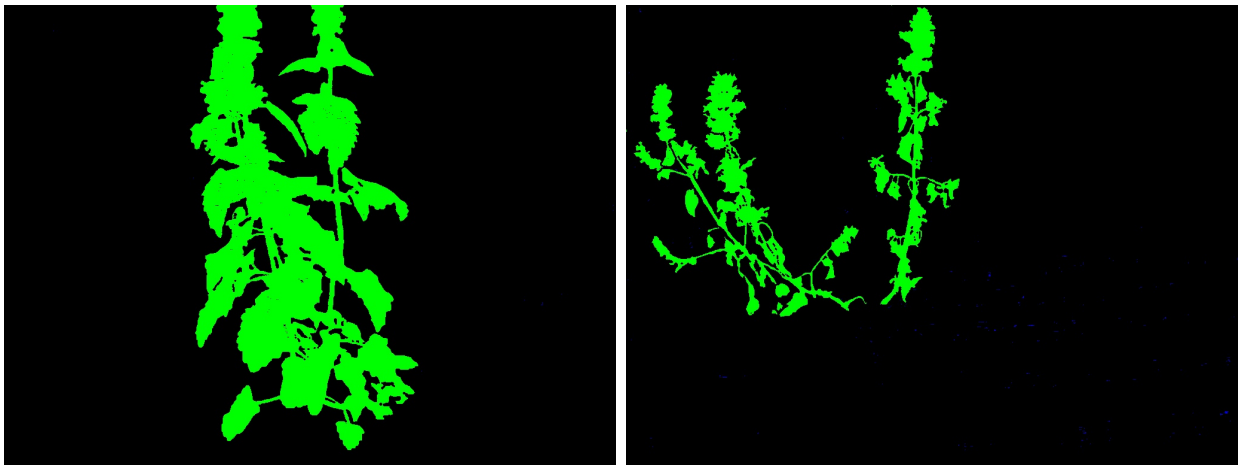
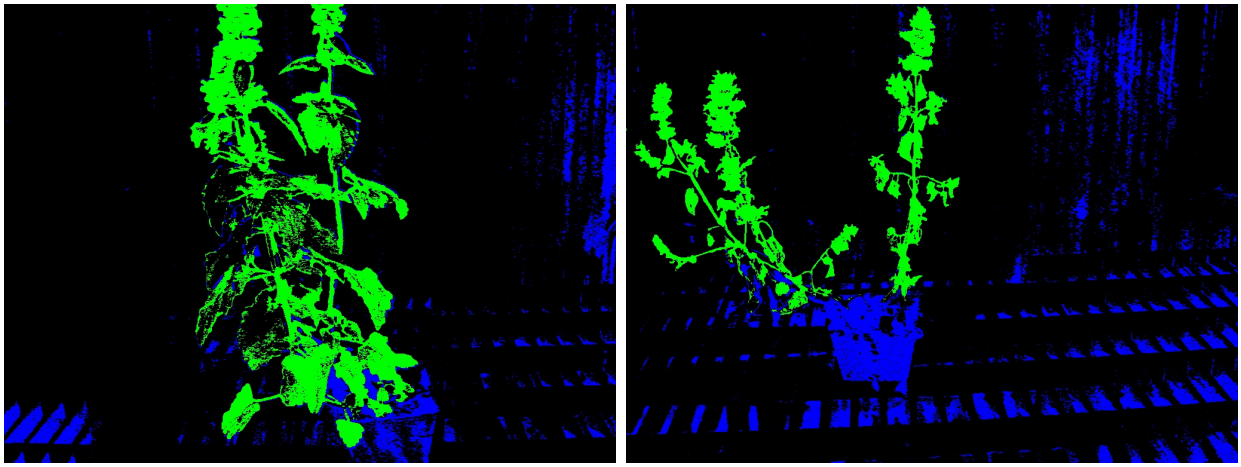


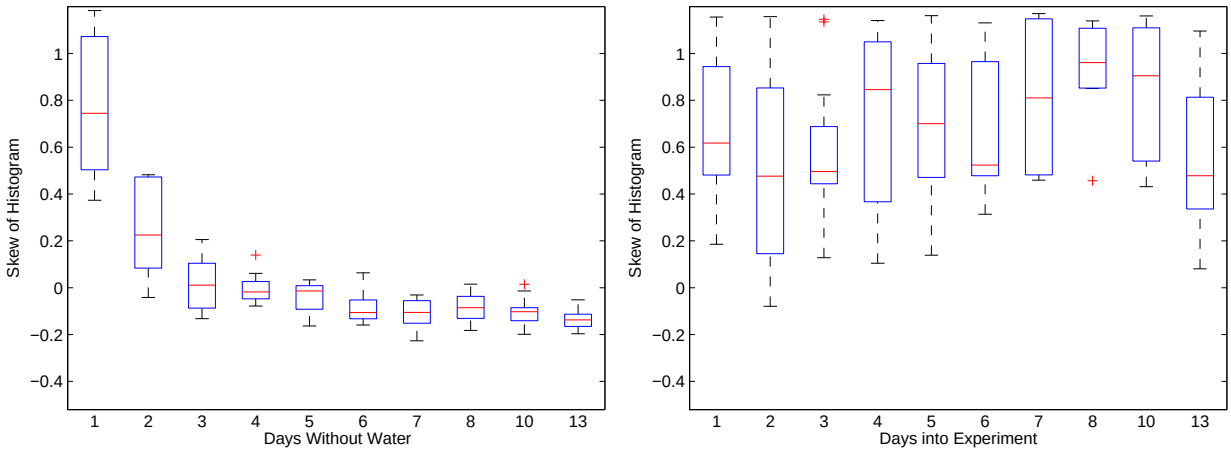
Figure A.9: Example NDVI Threshold Segmentation Failure



Appendix B

Texton Histogram Skew Plots

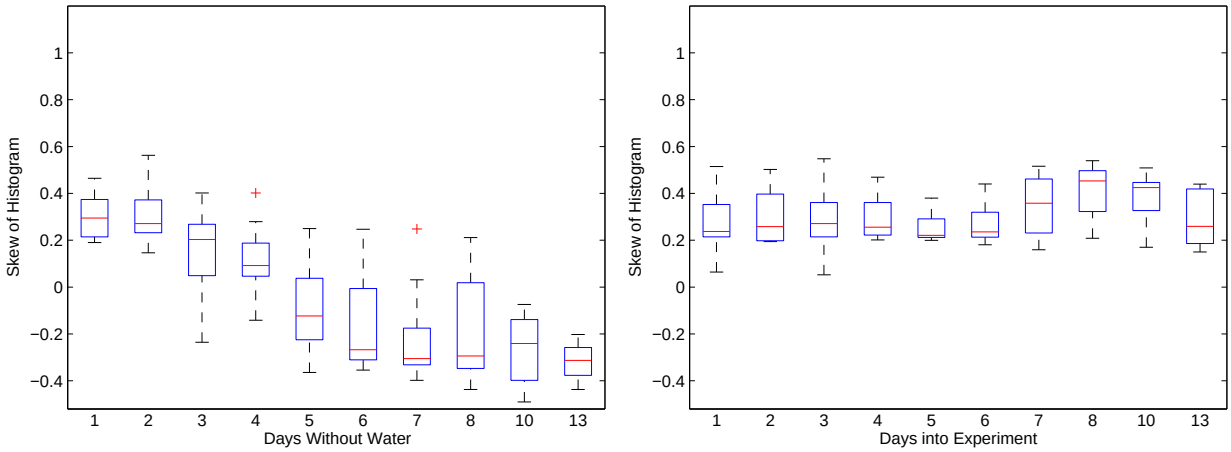
Figure B.1: Skew of Drought Texton Histograms with Catmint



(a) Drought Group

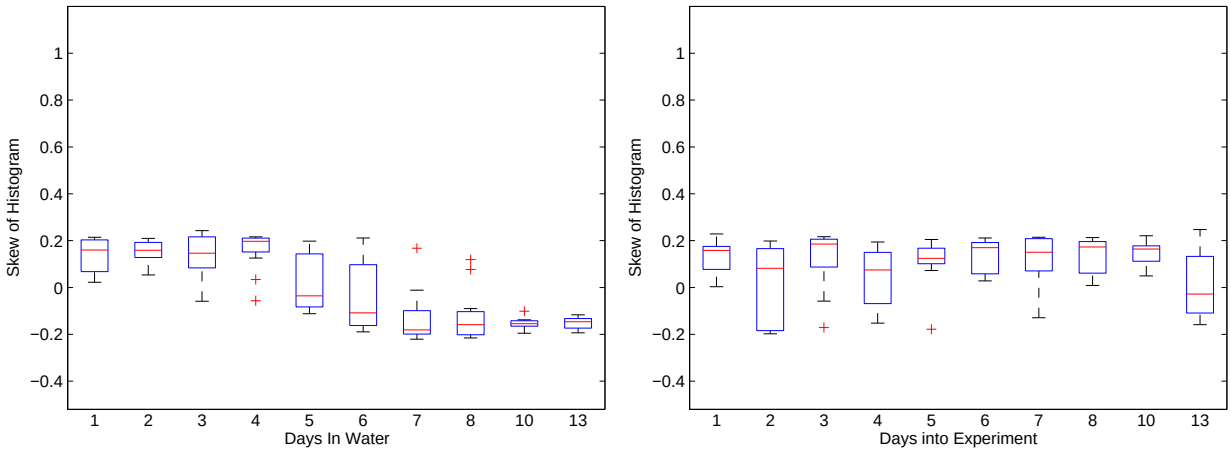
(b) Control Group

Figure B.2: Skew of Drought Texton Histograms with Basil



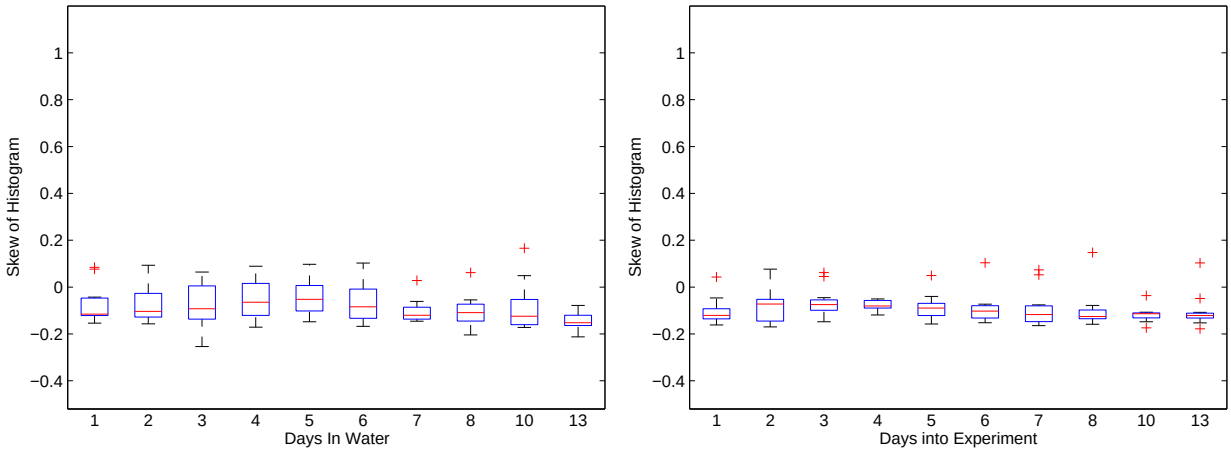
(a) Drought Group (b) Control Group

Figure B.3: Skew of Hypoxia Texton Histograms with Catmint



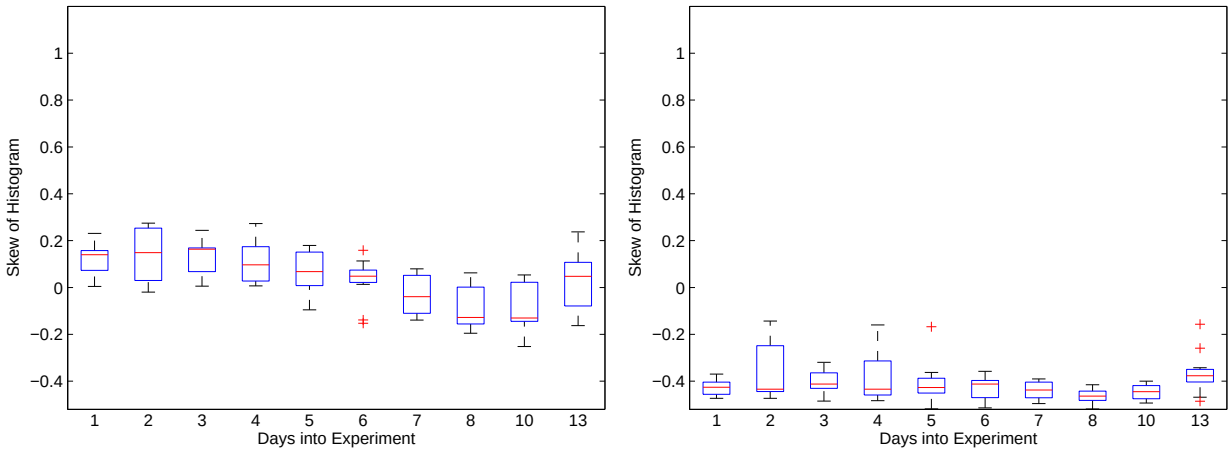
(a) Hypoxia Group (b) Control Group

Figure B.4: Skew of Hypoxia Texton Histograms with Basil



(a) Hypoxia Group (b) Control Group

Figure B.5: Skew of Control Texton Histograms



(a) Catmint (b) Basil