<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Signal and image processing technology for smart agriculture applications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Hamuda, Esmael Ali</td>
</tr>
<tr>
<td><strong>Publication Date</strong></td>
<td>2019-04-25</td>
</tr>
<tr>
<td><strong>Publisher</strong></td>
<td>NUI Galway</td>
</tr>
<tr>
<td><strong>Item record</strong></td>
<td><a href="http://hdl.handle.net/10379/15137">http://hdl.handle.net/10379/15137</a></td>
</tr>
</tbody>
</table>

Some rights reserved. For more information, please see the item record link above.
Signal and Image Processing Technology for Smart Agriculture Applications

A thesis presented by Esmael Hamuda to The College of Engineering & Informatics in fulfillment of the requirements for the degree of Doctor of Philosophy in the subject of Electrical & Electronic Engineering National University of Ireland, Galway, Ireland April 2019

Supervisor: Dr. Edward Jones Co-Supervisor: Dr. Martin Glavin
Declaration

I declare that this thesis titled, "Signal and Image Processing Technology for Smart Agriculture Applications" and the work presented in it are my own. I confirm that:

• This work was done wholly or mainly while in candidature for a research degree at this University.

• This work has not been submitted for a degree of any kind at this institution or any other institution.

• Where I have consulted the published work of others, this is always clearly attributed.

• Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

• I have acknowledged all main sources of help.

• Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Esmael Hamuda
April 2019
Acknowledgments

I would like to express my deepest appreciation to my supervisors, Dr. Edward Jones and Dr. Martin Glavin for their invaluable guidance, financial support, and encouragement. Without their dedication, this work would have not been done. I also would like to thank Dr. Brian McGinley for his support and assistance. Many thanks to my friends: Atif Shahzad, Damien Dooley, Darragh Mullins, Declan O’Loughlin, and Ashkan Parsi for their friendship and support.

Nobody has been more important to me in the pursuit of this project than the members of my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are the ultimate role models. Many thanks also to my brothers and sisters for their support. Most importantly, I wish to thank my loving and supportive wife, Hana, and my two wonderful children, Taleen and Eyad, who provide unending inspiration.
Sponsor Acknowledgment

This research was supported by funding from the Ministry of Higher Education, Libya and the National University of Ireland, Galway, Ireland.
Abstract

This thesis is concerned with development of signal and image processing technology for smart agriculture applications, with a particular focus on applications in automatic weeding systems. Developing an automatic weeding system requires robust detection of the exact location of the crop to be protected. Computer vision techniques can be an effective means of distinguishing crops from weeds and determining their locations. To achieve this, several practical issues need to be addressed such as weather variability, presence of shadows in sunny conditions, natural similarities between the target object (weed or crop) and the background, occlusion of objects of interest, and unexpected changes in camera parameters. This thesis addresses a number of these issues.

Firstly, a comprehensive study was conducted on image-based plant segmentation techniques (identifying plant from a background of soil and other residues). Three primary approaches, namely, (i) colour index-based segmentation, (ii) threshold-based segmentation, (iii) learning-based segmentation are discussed. The challenges and some opportunities for future developments in this space are identified. Secondly, a novel algorithm based on colour features and shape analysis is proposed to detect cauliflowers on a frame-by-frame basis from video acquired under various weather conditions (cloudy, partially cloudy, and sunny). The algorithm was tested under different weather conditions and achieved a detection performance of 98.91% and precision of 99.04%.

Then, in order to increase system robustness, the detection algorithm was extended through the addition of object tracking. A multi-object tracking algorithm based on Kalman filtering and the Hungarian algorithm was applied. With the help of the tracking algorithm,
detection failures were reduced, especially in sunny conditions, such that overall detection performance was raised from 97.28 to 99.34%.

Overlapping between plants (depending on the plant growth phase) is one of the most challenging problems that face computer vision techniques in real conditions, especially for plant segmentation and classification. In order to solve this issue, a novel approach based on main stem feature detection is proposed. Results of evaluation of the proposed algorithm show that the majority of plants were correctly detected with distance error of less than one centimetre, even in occluded conditions.

Finally, a comparative study of plant classification using deep learning approaches and traditional approaches was conducted. Two well-known deep learning architectures (AlexNet and GoogleNet), and three based on Support Vector Machine (SVM) with different feature sets (Bag of Words in L*a*b colour space feature, Bag of Words in HSV colour space, Bag of Words of Speeded-up Robust Features (SURF)) were applied. Results show that the best overall performance was achieved by deep learning-based approaches (AlexNet and GoogleNet), while the SVM-based approaches achieved close to the same performance.
# Contents

List of Figures ........................................... xvii

List of Tables ........................................... xxiii

Nomenclature .............................................. xxv

1 Introduction ........................................... 1

1.1 Background .......................................... 1

1.2 Weed control methods ................................ 2

1.2.1 Manual and mechanical weed control .............. 2

1.2.2 Chemical weeding .................................. 3

1.2.3 Motivation for this thesis .......................... 5

1.3 Contributions of this Thesis .......................... 6

1.3.1 Contributions ....................................... 6

1.3.2 Publications ........................................ 6

1.4 Thesis Structure ...................................... 7

2 Background .............................................. 9

2.1 Introduction .......................................... 9

2.2 Image processing in smart agriculture .............. 9

2.2.1 Pre-processing ..................................... 10

2.2.2 Segmentation ....................................... 10

2.3 Crop detection ......................................... 12
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>Use of tracking algorithms</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Dealing with occlusions</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>Plant classification using machine learning approaches</td>
<td>17</td>
</tr>
<tr>
<td>2.7</td>
<td>Concluding remarks</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>Plant Segmentation</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>Colour Index-Based Approaches</td>
<td>23</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Normalised Difference Index (NDI)</td>
<td>23</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Excess Green Index (ExG)</td>
<td>24</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Excess Red Index (ExR)</td>
<td>25</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Colour Index of Vegetation Extraction (CIVE)</td>
<td>25</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Excess Green minus Excess Red Index (ExGR)</td>
<td>26</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Normalised Green-Red Difference Index (NGRDI)</td>
<td>26</td>
</tr>
<tr>
<td>3.2.7</td>
<td>Vegetative Index (VEG)</td>
<td>27</td>
</tr>
<tr>
<td>3.2.8</td>
<td>Combined Indices 1 (COM1)</td>
<td>27</td>
</tr>
<tr>
<td>3.2.9</td>
<td>Modified Excess Green Index (MExG)</td>
<td>27</td>
</tr>
<tr>
<td>3.2.10</td>
<td>Combined Indices2 (COM2)</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>Evaluation of plant extraction based on colour indices</td>
<td>28</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Verification of individual colour indices</td>
<td>29</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Machine learning approaches</td>
<td>31</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Combination of machine learning approaches and colour indices</td>
<td>34</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Verification of combinations of colour indices</td>
<td>35</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Colour spaces approaches compared to colour indices and machine</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>learning</td>
<td></td>
</tr>
<tr>
<td>3.3.6</td>
<td>Use of colour indices with images taken from the air</td>
<td>46</td>
</tr>
<tr>
<td>3.3.7</td>
<td>Performance in the presence of strong illumination</td>
<td>46</td>
</tr>
<tr>
<td>3.4</td>
<td>Other Segmentation Approaches</td>
<td>50</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Threshold-Based Approaches</td>
<td>50</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Learning-Based Approaches</td>
<td>52</td>
</tr>
</tbody>
</table>
## 4 Development of a Novel Crop Detection Algorithm for Varying Illumination Conditions

4.1 Introduction ............................ 61
4.2 Background and Related Work .................. 62
4.3 Proposed Detection Algorithm .................. 64
4.4 Algorithm Steps ............................ 67
    4.4.1 Algorithm Outline .................. 67
    4.4.2 Threshold parameter selection .......... 72
4.5 Testing framework and performance metrics ............ 75
    4.5.1 Test Data .................. 75
    4.5.2 Performance evaluation .......... 76
4.6 Results and discussion ........................ 77
    4.6.1 Performance Evaluation .......... 77
    4.6.2 Comparison with HSV decision tree method .... 81
4.7 Conclusions and future work .................. 81

## 5 Enhanced Crop Detection System using Kalman Filter-based Tracking

5.1 Introduction ............................ 85
5.2 Proposed method ............................ 86
    5.2.1 Kalman filter .................. 87
    5.2.2 Hungarian data association algorithm ........ 94
    5.2.3 Additional post-processing .......... 94
5.3 Testing framework and performance metrics .......... 96
    5.3.1 Performance evaluation .......... 96
    5.3.2 Bounding box and Tracking evaluation ........ 96
5.4 Results and discussion ........................ 97
    5.4.1 Performance of tracking algorithm .......... 97
5.4.2 Comparison with Optical Flow based on the Lucas-Kanade algorithm (OF-LK) .................................................. 101
5.5 Conclusion ......................................................... 103

6 Detection of plants in conditions of high overlapping .................................................. 105
6.1 Introduction ......................................................... 105
6.2 Proposed Algorithm .............................................. 106
   6.2.1 Algorithm steps ............................................. 106
   6.2.2 Performance evaluation ................................... 112
6.3 Results and Discussion .......................................... 113
6.4 Conclusions ......................................................... 119

7 Plant Classification using Machine Learning ............................................................... 121
7.1 Introduction ......................................................... 121
7.2 Machine Learning Algorithms and Features ......................................................... 122
   7.2.1 SVM model ................................................. 122
   7.2.2 Bag of Words (BoWs) ..................................... 122
   7.2.3 Deep learning approaches ................................ 123
7.3 Testing framework and performance metrics ...................................................... 124
   7.3.1 Training Set ................................................ 125
   7.3.2 Data Augmentation (DA) ............................... 125
   7.3.3 Challenge set ............................................. 127
   7.3.4 Performance Evaluation .................................. 128
7.4 Results and Discussion ........................................... 128
   7.4.1 Parameter Selection ...................................... 128
   7.4.2 Results from test Set .................................... 129
   7.4.3 Results from challenge Set ......................... 130
7.5 Conclusions ......................................................... 132
8 Conclusions and Future Work 135

8.1 Project Summary and Conclusions .................................. 135

8.1.1 Primary Contributions ............................................. 137

8.2 Suggestions for Future Work ....................................... 137

References 141
List of Figures

3.1 General scheme for segmentation and its evaluation .......................... 22
3.2 Comparison of the performance of selected colour indices: ExGR, ExG+Otsu, and NDI-Otsu under greenhouse conditions. SD is indicted by error bar in the plot ................................................. 30
3.3 Comparison of the performance of selected colour indices: ExGR, ExG+Otsu, and NDI+Otsu under actual field conditions. SD is indicted by error bar in the plot ................................................. 31
3.4 Comparison of mis-classification of CIVE and ExG for GV and GVS image types .......................................................... 32
3.5 Comparison of mis-classification of CIVE and ExG for both image types (NGV & NGVS) .......................................................... 33
3.6 The average segmentation quality for CIVE, ExGR, and NDI .................. 35
3.7 Comparison of average error of greenness segmentation for COM1, CIVE, ExGR, ExG, and VEG .................................................. 36
3.8 Comparison of the mean and standard deviation of vegetation extraction for ExG, ExGR, VEG, and CIVE. SD is indicted by error bar in the plot . . . . 38
3.9 Comparison of the segmentation quality ($Q_{seg}$ & $S_r$) of plant extraction for ExGR, MExG, and ExG for two different data sets under non-sunny conditions 40
3.10 Comparison of the segmentation quality ($Q_{seg}$ & $S_r$) of plant extraction for ExGR, MExG, and ExG for two different data sets under sunny conditions . 40
3.11 Comparison of the mean and standard deviation of segmentation for ExGR, ExG+Otsu, and CIVE based on ATRWG metric. SD is indicted by error bar in the plot. ................................................. 42

3.12 Comparison of the mean plant extraction for ExGR, ExG+Otsu threshold, and CIVE under cloudy, overcast, and sunny conditions based on ATRWG metric ................................................. 43

3.13 Comparison of the mean and standard deviation of segmentation for ExGR, ExG+Otsu, and CIVE based on evaluated method which is defined Xiao et al., SD is indicted by error bar in the plot ................................................. 44

3.14 Comparison of mean and standard deviation of plant extraction for ExGR and ExG+Otsu. SD is indicted by error bar in the plot ................................................. 45

3.15 Comparison of the mean and standard deviation of plant extraction for ExG, VEG, COM1, COM2, ExGR, NGRD, WI, and CIVE of images were captured at 30m flight altitude ................................................. 47

3.16 Comparison of mean and standard deviation of crop extraction for ExG, NDI, VEG, and CIVE. SD is indicted by error bar in the plot ................................................. 48

4.1 General scheme for crop detection algorithm ................................................. 66

4.2 Pre-processing results. (a) Original image. (b) Blurred image. (c) HSV image. (d) Filtered image using HSV ranges. (e) Image after morphological erosion and dilation ................................................. 69

4.3 Example image (RGB) and its HSV channels; Hue image (H), Saturation image (S), and Value image (V) ................................................. 73

4.4 The histogram of Hue, Saturation, and Luminosity Value of cropped image. The red arrows indicate the approximate boundaries of different types of material in the image, including:: cauliflower (item of primary interest), weeds, soil, and other residues ................................................. 73
4.5 An example of results from the proposed method for a set of images were captured in various light conditions. (a) Cloudy/overcast day; (b) Partially cloudy day; (c) Sunny day. Cauliflower plant regions are highlighted with red rectangles. Green dots indicate the location of the centre of mass of each crop. 79

4.6 An example of false detection results from the detection algorithm for a test image, frame numbers 226 captured in sunny condition. A single weed was wrongly detected and highlighted with a red rectangle as a cauliflower. Each cauliflower plant in the row is independently detected by the algorithm and highlighted with red rectangle; ground truth is indicated by blue rectangles; yellow rectangles indicate the overlap between detections and annotations. Green dots indicate the location of the centroid of each plant. 80

4.7 Comparison between proposed method and Yang’s method. First column shows test images. The second column includes the segmentation results from Yang’s method. The third column includes the segmentation results of the proposed method, and the fourth column includes the detection results from the proposed methods (Cauliflower plant regions are highlighted with red rectangles). 82

5.1 General scheme for the tracking algorithm. 87
5.2 An example of results from the detection algorithm Chapter 4 for a pair of test images, frame numbers 508 (a) and 457 (b) captured in sunny condition. Each cauliflower plant in the row is independently detected by the algorithm and highlighted with a red rectangle; ground truth is indicated by blue rectangles; yellow rectangles indicate the overlap between detections and annotations. Red dots indicate the location of the centroid of each plant. The results from the proposed algorithm of test images (frame number 508 (c) and 457 (d). Each cauliflower plant in the row is independently tracked by the proposed algorithm and highlighted with red rectangle; ground truth is indicated by blue rectangles; yellow rectangles indicate the overlap between detections and annotations. Red dots indicate the location of the centroid of each crop. ................................. 98

5.3 The plotting results of detection and tracking algorithms. ................. 100

5.4 Example image and output of stages of the OF-LK algorithm. (a) Original image with blue annotated rectangle; (b) Detection results; (c) Harris corners results with corner points indicated by red stars; (D) Tracking results of the OF-LK algorithm (red box), and the overlap between tracked plant and ground truth (yellow box). ........................................ 102

5.5 The plotting results of tracking algorithm using Optical Flow based on the Lucas-Kanade algorithm. ........................................ 102

6.1 The flow chart of the proposed algorithm ............................... 107

6.2 The results of each step of the proposed algorithm ...................... 110

6.3 An example of two overlapping objects (A and B) ...................... 111
6.4  An example of results from the proposed algorithm. (a) Two overlapping cauliflowers in normal conditions (low weed coverage). (b) Cauliflower occluded by heavy weeds. (c) Two overlapping cabbages with leaves of flat surfaces in normal conditions. (d) Two overlapping cabbages with leaves of flat surfaces in sunny conditions. (e) Two overlapping cabbages with leaves with textured surfaces in sunny conditions. (f) Two overlapping cabbages with leaves of textured surfaces in sunny conditions. The blue rectangle represents the first plant, and the green one represents the second plant. These rectangles are automatically drawn by the proposed algorithm. Red dots indicate the location of the centre of each crop automatically determined by the algorithm. ................................................................. 114

6.5  The distance error of the proposed method for each test image. ............... 115

6.6  The overall performance of the proposed algorithm ............................. 116

6.7  An example of different camera attitudes and their corresponded appearance of the main veins of the same cauliflower. The green lines represent the main vein ............................................................. 118

7.1  An illustration of the architecture of AlexNet ................................. 124

7.2  The inception module of GoogleNet with dimension reduction .......... 125

7.3  Sample images from training set. Figure (a) represent the positive set (cauliflowers), while figure (b) represent the negative set (weeds) ........... 126

7.4  Sample images of Evaluation set. Figure (a) represent the positive set (cauliflowers), while figure (b) represent the negative set (weeds) ......... 127
# List of Tables

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Comparison of plant segmentation methods based colour indices</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of threshold based segmentation methods</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison of learning based segmentation methods</td>
</tr>
<tr>
<td>3.4</td>
<td>Suggested segmentation algorithms for use in different conditions.</td>
</tr>
<tr>
<td>4.1</td>
<td>The ranges of HSV channels for cauliflower pixels under the corresponding weather conditions</td>
</tr>
<tr>
<td>4.2</td>
<td>The selected average range of S channel for each weather condition</td>
</tr>
<tr>
<td>4.3</td>
<td>Results produced by average metric</td>
</tr>
<tr>
<td>4.4</td>
<td>Overall performance of proposed algorithm.</td>
</tr>
<tr>
<td>5.1</td>
<td>Hungarian algorithm steps</td>
</tr>
<tr>
<td>5.2</td>
<td>Comparison of performance for video recorded in sunny conditions</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of performance for video recorded in sunny conditions</td>
</tr>
<tr>
<td>7.1</td>
<td>Average test accuracy on Test Set</td>
</tr>
<tr>
<td>7.2</td>
<td>Overall results on the challenge Set</td>
</tr>
</tbody>
</table>
Nomenclature

Acronyms / Abbreviations

A  Transition matrix

$\alpha_\theta$  Accuracy factor

APHI  Affinity Propagation-Hue Intensity

ATRWG  Automatic Target Recognition Working Group

$B_{det}$  detected bounding box

$B_{gt}$  ground truth (annotated) bounding box

BoW  Bag of Word

$B_{rk}$  Tracked bounding box

CGHMMKF  Combined Gaussian Hidden Markov Model and Kalman Filter

CIVE  Colour Index of Vegetation Extraction

CNN  Convolutional Neural Network

COM1  Combined Indices 1

COM2  Combined Indices 2

CPWC  Critical Period of Weed Control
DA Data Augmentation

$D_N$ vector of distances between all intersection points

EASA Environmentally Adaptive Segmentation Algorithm

ExG Excess Green Index

$ExGR$ Excess Green minus Excess Red Index

ExR Excess Red Index

FLD Fisher Linear Discriminant

$F_N$ False Negative

$F_P$ False Positive

$\lambda$ performance measure for plant segmentation

$\sigma$ The standard deviation of Gaussian distribution

$\theta$ angle-radius parameterization

$M_{00}$ zero order moment

$M_{01}$ First order moment for y-axis

$M_{02}$ Second order moment for y-axis

$M_{10}$ First order moment for x-axis

$M_{20}$ Second order moment for x-axis

GMM Gaussian Mixture Modeling

GST Generalized Search Tree

GV Green Vegetation without Shadow
GVS  Green Vegetation with Shadow

$G_x$  Gradient component of $x$ of the Sobel operation

$G_y$  Gradient component of $y$ of the Sobel operation

$H_{\text{max}}$  The maximum value of Hue channel

$H_{\text{min}}$  The minimum value of Hue channel

$HSI&B – Spline$  Hue–Saturation–Intensity and B-Spline curve fitting method

HSI  Hue Saturation and Intensity values

HSV  Hue Saturation Value

$K$  Kalman Gain

LBP  Local Binary Pattern

$MExG$  Modified Excess Green Index

MHT  Multiple Hypothesis Tracking

MSBPNN  Mean-shift algorithm with Back Propagation Neural Network

MSFLD  Mean-shift algorithm with Fisher Linear Discriminant

MS  Mean Shift

$NDI$  Normalised Difference Index

$NGRDI$  Normalised Green-Red Difference Index

NGV  Non-Green Vegetation without Shadow

NGVS  Non-Green Vegetation with Shadow

$OP – LK$  Optical Flow and Lucas-Kanade algorithm
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>Prediction error covariance</td>
</tr>
<tr>
<td>( P )</td>
<td>the perimeter of the contour</td>
</tr>
<tr>
<td>( \hat{P}_k )</td>
<td>Covariance error</td>
</tr>
<tr>
<td>( Pre )</td>
<td>Precision</td>
</tr>
<tr>
<td>PSMRF</td>
<td>Probabilistic Superpixel Markov Random Field</td>
</tr>
<tr>
<td>PSOMM</td>
<td>Particle Swarm Optimisation clustering and Morphology Modelling</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>( Q )</td>
<td>Covariance matrix</td>
</tr>
<tr>
<td>( Q_{seg} )</td>
<td>Segmentation quality based on both plants and background regions</td>
</tr>
<tr>
<td>( R )</td>
<td>Measurement error covariance</td>
</tr>
<tr>
<td>( R )</td>
<td>Measurement noise matrix</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>( Rec )</td>
<td>Recall</td>
</tr>
<tr>
<td>RGB</td>
<td>Red Green Blue</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>SD</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>( S_{\text{maxavg}} )</td>
<td>The maximum average value of Saturation channel</td>
</tr>
<tr>
<td>Smax</td>
<td>The maximum value of Saturation channel</td>
</tr>
<tr>
<td>( S_{\text{minavg}} )</td>
<td>The minimum average value of Saturation channel</td>
</tr>
<tr>
<td>Smin</td>
<td>The minimum value of Saturation channel</td>
</tr>
</tbody>
</table>
Nomenclature

\( S_r \)  Segmentation quality based on both plants and background regions

\( S_r \)  green segmentation quality

SURF  Speeded-up Robust Features

SVM  Support Vector Machines

\( T_P \)  True Positive

\( VEG \)  Vegetative Index

\( VF \)  Vegetation Fraction

\( V_{\text{max}} \)  The maximum value of Value channel

\( V_{\text{min}} \)  The minimum value of Value channel

VOC  Visual Object Classes

\( \hat{x}_k^- \)  Priori estimate of state
Chapter 1

Introduction

1.1 Background

Weeds are one of the biggest challenges facing crop cultivation in agriculture because they appear everywhere randomly, and compete with the plant for resources. As a result of this competition for resources, crop yields suffer. Yield losses depend on factors such as weed species, population density, and relative time of emergence and distribution, as well as on the soil type, soil moisture levels, pH and fertility [1]. Numerous researchers have identified a strong link between weed competition and crop yield loss for a wide range of crop varieties. For example, according to the study by [2], an annual loss of 146 million pounds of fresh market sweet corn and 18.5 million pounds of sweet corn for processing occurred in the United States from 1975 to 1979 due to weed competition, which corresponds to revenue losses of $13,165,000 and $9,155,000 respectively. Furthermore, the dry and head weight of crop yield is measured to evaluate losses. Based on a study carried out in 1996/1997 and repeated in 1997/1998 in central Jordan [3], it was found that the average reduction in shoot dry weight and head yield due to weeds were 81% and 89% respectively. An effective and efficient weed management system is therefore necessary to minimise yield losses in valuable crops. The critical period for weed control must be taken into account to enhance
weed management strategies [4], as the duration of co-existence of weed and crop is an important indicator of yield losses due to weed competition [5].

In [6, 7] Zimdahl et al. defined the critical period of weed control (CPWC) as "a span of time between that period after seeding or emergence when weed competition does not reduce crop yield and the time after which weed competition will no longer reduce crop yield". A more quantitative definition is the number of weeks after crop emergence during which a crop must be weed free in order to prevent yield losses greater than 5% [8–10].

A number of studies have been carried out in many different locations, under different environmental conditions in an attempt to establish the CPWC. The studies are generally conducted by keeping the crop free from weeds for a fixed period of time, and then allowing the weeds to infest the crop. Another approach used is growing weeds with the crop for certain predetermined durations, after which all weeds are removed until the growing season ends [11]. Some studies have reported that weeds that emerge at the same time as the crop, or slightly after, cause greater yield loss than weeds emerging later in the growth cycle of the crop [12–14]. Most studies recommend that crops should keep weed-free within the CPWC in order to minimise yield loss (e.g. [4]).

1.2 Weed control methods

1.2.1 Manual and mechanical weed control

Manual methods for weed control include hand weeding and use of simple hand tools. Hand weeding is a conventional weed removal method that has been successfully used to control weeds for many centuries, before any other methods existed. It is an environmentally friendly solution but is not practical for large scale commercial farms because it is extremely labour intensive, costly, tedious, and time consuming. The high wage rate of labour is also an issue. According to a report in 1996 by the United States Department of Agriculture (USDA),
1.2 Weed control methods

growers who used to pay $0.10/h were faced with paying $0.50/h in the early 1950s and $1.00/h in the 1960s [15]. Nowadays, manual labour for weed control is expensive and is not often readily available in Europe, the United States, and many other countries [16].

Mechanical methods for weed control (by tillage or cultivation of the soil) are mostly applied for inter-row weed control in large areas for row crops such as sugar beet, wheat, and corn. A number of studies have been carried out to evaluate the efficacy of mechanical weed control methods. In [17] Forcella et al. reported that rotary hoeing yielded approximately 50% weed control alone without using other weed control methods such as herbicides and manual labour. In [18] Donald et al. found that inter-row mowing systems for controlling both winter annual and summer annual weeds may reduce the use of herbicides by approximately 50%.

Mechanical weeding is particularly suited for weed control in organic fields (where the use of herbicides is not appropriate) and can also be helpful in conventional fields. On the other hand, it may also have negative effects on crops, in that the cash crop plants may be damaged by the use of machinery, and on the environment through increasing soil erosion and nutrient loss. Cultivation can be quite harmful for crops such as potatoes and peanuts, reducing their yields because of root pruning caused by the machine tools. In [19] Nelson and Giles found that due to root pruning, potato yield reduced by 3 to 31%. A further potential drawback of mechanical weeding systems is that instead of eliminating weeds, cultivation may contribute to spreading weed species [20, 21]. Moreover, it can be practically difficult to apply the tillage process in wet soil (particularly during a rainy season) to control weeds [22].

1.2.2 Chemical weeding

Chemical weeding is the most widely used method for weed control in agriculture since the introduction of synthetic organic chemicals in the late 1940s, and farmers now rely heavily
on herbicides for effective weed control in crops [23]. Several researchers have found that using herbicides was a primary factor in controlling weeds and increasing crop yield [24–26], particularly on large scale commercial farms. It has been reported that in 1998 and 1999, 87 million ha of cropland were sprayed with herbicide, and herbicides accounted for approximately 60% of the total volume and 65% of the expenditure for all pesticides used by U.S farmers [27]. One of the motivations for the adoption of herbicides in U.S. farms is the desire to reduce weed control costs because of the shortage of labour. Many studies have documented that the use of herbicides is a more economical method for controlling weeds compared to hand and mechanical weeding. For example, with the help of herbicides, farmers in Mississippi were estimated to have saved $10 million per year in labour costs [23]. The adoption of chemical sprays for weed control is credited as the primary factor in the reduction in cost of crop production, saving time on cultivation and hand weeding. For example, herbicides reduced the number of tillage trips in almond fields by 16 and saved $52/ha [28]. On the other hand, increasing demand for chemicals by farmers has increased the market size. According to [29], the biopesticide and synthetic pesticide market are expected to reach up to $83.7 billion by 2019.

Although herbicides are very effective at controlling weeds, they have clear negative impacts on both the environment (through pollution) and plant biology (development of resistance). Groundwater and surface water pollution has been reported in many cases in recent decades, and excessive use of herbicide has often been found to be the cause [30, 31]. To counteract these catastrophic environmental effects, most European countries have introduced legislative directives to restrict the use of herbicides in agriculture [32]. The need to reduce chemical usage and find alternative methods of weed management has increased in urgency in recent years. Some researchers have reported that it is possible to control weeds by applying lower rates of herbicides in combination with mechanical weeding [33, 34]. If there are means to accurately detect and identify weed spatial distribution (weed
patches), it is possible to limit herbicide quantities by applying them only where weeds are located [35–39]. For example, in [40] Heisel et al. demonstrated a potential herbicide saving of 30 to 75% through the use of appropriate spraying technology and a decision support system for precision application of herbicides.

1.2.3 Motivation for this thesis

Given the challenges associated with traditional means of weed control based on herbicides in particular, and the high labour costs associated with mechanical weeding, there is increasing interest in more sustainable and environmentally-friendly approaches. Automated weeding technology is of increasing interest as a means of reducing labour costs and increasing productivity, giving rise to developments in “smart agriculture”. An important component of such systems is a means to distinguish between plants and weeds, so that only weeds are actually removed by automated processing [41–43]. Image processing technology presents a promising approach to carrying out this automated detection of weeds and plants and is currently used in numerous projects in smart agriculture, resulting in increased yields, reduced spraying, more efficient growth methods and increased profit. Besides distinguishing between weeds and plants to permit precision weeding, it could be also useful for other purposes, and applied in several applications such as plant species recognition [44], growing phase determination [45], and plant disease detection [46]. While weeding remains the most important motivator at present, these other applications are growing in importance with increasing interest in smart agriculture. Despite developments in this field, important challenges remain to be solved; this thesis addresses some of these challenges.
1.3 Contributions of this Thesis

1.3.1 Contributions

This thesis is concerned with development of signal and image processing technology for smart agriculture applications, and in particular, a range of techniques is developed to improve performance in detection and classification of plants and weeds. For the purposes of this thesis, the term “crop” will be used to refer to the cash crop of interest, e.g. cauliflower is of primary interest in this work. Several issues of importance in the development of image processing technology in these applications are addressed in this thesis. The primary contributions of this thesis are as follows:

1. A detailed review of plant segmentation-based approaches from the literature, including analysis of the suitability of different algorithms for particular growing conditions.

2. Development of an image processing algorithm to recognize cauliflower from weeds in video sequences, at different growing stages, and against different weather conditions, by applying HSV colour space and shape feature analysis.

3. An improved detection algorithm incorporating tracking in order to overcome problems with the basic detection algorithm particularly in sunshine.

4. Development of an image processing algorithm to detect plant locations in conditions of high occlusion.

5. Investigation of the use of deep learning approaches for plant classification (cauliflower and cabbage versus weeds) and comparison with more well-established approaches such as Support Vector Machines (SVM).

1.3.2 Publications

The publications that have resulted from this research are as follows:
1.4 Thesis Structure

The remainder of this thesis is organised as follows:

Chapter 2 provides background information for the work described in the thesis, including a review of image-based plant segmentation (identifying plant from a background of soil and other residues), plant detection, and classification (identifying the plant as either crop or


weed). In addition, multi-object tracking algorithms that are used in agricultural applications and other computer vision applications are also discussed. A brief discussion of deep neural network machine learning approaches is also given.

Chapter 3 presents a detailed review of plant-segmentation techniques, as well as discussing the advantages and disadvantages of different approaches, and discussing the suitability of different algorithms for different conditions.

Chapter 4 presents the development of an algorithm for crop detection to recognise cauliflower from weeds in different growing stages and presents performance results of the algorithm from a varying database of images created for this work.

The following chapter addresses the limitations of the system presented in Chapter 4, and presents a system to increase the robustness of crop detection system, particularly in sunny conditions through a multi-target tracking algorithm based on the Kalman filter and the Hungarian algorithm.

Chapter 6 addresses the problems of detecting plants in occluded conditions and presents a method for plant detection in such conditions. The algorithm is tested using a database of cauliflower plants in highly occluded conditions.

Chapter 7 investigates the use of deep learning approaches for plant classification (cauliflower and cabbage against weeds) in smart agriculture applications. Five approaches are considered, two based on well-known deep learning architectures (AlexNet and GoogleNet), and three based on Support Vector Machine (SVM) with different feature sets (Bag of Words in L*a*b colour space feature, Bag of Words in HSV colour space, Bag of words of Speeded-up Robust Features (SURF)).

Finally, Chapter 8 gives a summary and conclusions, and suggested future work.
Chapter 2

Background

2.1 Introduction

Chapter 1 presented an overview of the application domain presented in this thesis, and discussed the rationale for the development of technology to facilitate more effective crop management and in particular, techniques for weed management. This in turn motivates the use of image processing and machine vision technology to support detection and classification of plants and weeds. This chapter reviews the state of the art relating to some key elements of the work addressed in this thesis, including segmentation, detection, multi-object tracking, and classification techniques.

2.2 Image processing in smart agriculture

Machine vision technology has already been widely used and studied in agriculture to identify and detect plants (crops and weeds). It has shown potential for success in a number of case studies in robotic weed control systems despite some serious challenges that will be discussed in later chapters [47–50]. Machine vision technology has also been applied in other agricultural applications such as grading and harvesting fruits [51–53]. As summarised
in [54], many researchers have developed image processing methods as guidance for machine vision, working in different fields and environments (under controlled and uncontrolled conditions). Image-based segmentation techniques mostly involve two main stages: pre-processing and pixel classification. This section discusses some of the steps involved in image processing for crop and weed detection in agriculture. A particular challenge in outdoor applications is variation in illumination and much research has been targeted at addressing this problem.

### 2.2.1 Pre-processing

Pre-processing involves some important initial processing on the original image from the camera such as enhancement and removing noise. Image enhancement has played a significant role in various applications such as medical imaging, industrial inspection, remote sensing, and plant disease detection. Image enhancement is a process used for enhancing and adjusting the contrast of the acquired image to address issues such as the variability of luminance due to sunlight and shadow [55]. Colour conversion is used to address lighting problems in the scene of an image. For example, in [56] Perez et al. applied the normalised difference index (using only green and the red channel) to reduce the illumination effect and discriminate between plants and background. Filtering is also used in image enhancement; in agricultural application, colour conversion and histogram equalization are used for plant leaf disease detection [57]. For instance, homomorphic filtering is a technique that has the ability to minimise illumination issues and has been successfully applied in outdoor images under various environmental conditions [58].

### 2.2.2 Segmentation

The initial goal in almost all image processing plant detection approaches is to segment the different pixels which appear in images into two classes: plant (crops and weeds) and
background (soil and residues). In [54] Slaughter et al. considered image processing techniques for detection and discrimination of plants and weeds in some detail. The plant has to be segmented from background soil, considering all field conditions, because mis-segmentation could seriously affect the accuracy of plant/weed detection. Among other things, Slaughter et al. concluded that natural illumination plays a crucial role in effective plant segmentation, and poor illumination contributes to poor plant segmentation. They also found that most of the available machine vision techniques are not sufficiently robust for real time conditions. High segmentation performance is required for precision chemical application, and with good performance, the volume of herbicides that are applied to the fields can be minimised.

A large number of studies employed colour index-based methods to segment plants (crops and weeds) from the background (soil and residues) under various image conditions [59, 60, 45, 61–64]. These methods are most commonly based on the RGB colour space, and they generally rely on the principle that the green channel contains more useful information than the red one. However, these methods by themselves were unable to discriminate between crops and weeds. Often they could not even completely separate plant pixels from the background without threshold adjustments. Furthermore, in such algorithms, the colour-based methods failed to segment plant pixels from the background in low or bright lighting conditions. To overcome this problem, several researchers proposed learning-based methods such as the Environmentally Adaptive Segmentation Algorithm (EASA) proposed by [65, 66], the Mean-shift algorithm with Back Propagation Neural Network (MS-BPNN) [67], the Mean-shift algorithm with Fisher Linear Discriminant (MS-FLD) [68], a Decision Tree based Segmentation Model (DTSM) [69], Affinity Propagation-Hue Intensity (AP-HI) [70], and Particle Swarm Optimisation clustering and Morphology Modelling (PSO-MM) [71]. Although these methods demonstrate good vegetation segmentation results under various lighting conditions, they are computationally expensive and may not suit real time
application. In addition, they were applied only for segmentation of all vegetation from the background, but not for distinguishing between crop and weeds. A comprehensive and critical survey on image-based plant segmentation techniques is described in Chapter 3 of the thesis.

### 2.3 Crop detection

Numerous researchers have applied image processing techniques based on different features including colour, texture, and shape to distinguish between crops and weeds. Some have demonstrated good performance especially under controlled conditions, however, in uncontrolled conditions, accurate detection remains a challenging problem.

Several research studies have applied texture feature analysis to identify plant species [72–75], while other studies have used textural analysis approaches to classify between crops and weeds [76–78].

The principal advantage of using colour-based techniques is that computational requirements are reduced compared to other techniques [54]. Colour features are most effective for crops with a distinguishable and unique colour. For instance, in [79] Lamm et al. found that the 'Maxxa' cotton plant has red pigment in the leaf tissue (natural red spots where the petiole attaches to leaf). This feature distinguishes the cotton plant from weeds, and mitigates partial occlusion because the locations of red spots are close to the centre of the cotton plant. In [80] Lee et al. also demonstrated that colour features can be robust and accurate in distinguishing tomato seedlings from weeds even when partially occluded. This is because tomato seedlings are deep purple in colour. The disadvantage of these algorithms is their narrow range of applications, i.e. they are only effective where the target crop has a unique colour (different from weeds).

In several cases where the crop colour is close to that of the surrounding weeds, colour alone is likely to be insufficient to accurately distinguish crop pixels from weed pixels. Thus,
several research studies have applied shape features with colour simultaneously to improve accuracy. In [81] Astrand and Baerveldt demonstrated that using colour features to distinguish between sugar beet and weed was feasible and achieved a success rate of 92%. Furthermore, when two shape features (compactness, elongation) were combined together with colour features, the accuracy increased to 96%. In [56] Perez et al. also demonstrated that using shape features combined with colour increased the weed detection accuracy rate from 75% up to 85%. In [82] Cho et al. applied colour features to discriminate plants (radishes and weeds) from the background, and extracted several shape to improve classification. The detection rate was 92% for radish and 98% for weeds. In [83] Kiani and Jafari applied the Excess Green Index (ExG) which was proposed in [60] to separate plant vegetation from the soil background. Several shape features were fed to an artificial neural network to classify between corn and weed. The algorithm demonstrated good classification results with a detection rate of 100% for corn plants while only 4% of weeds were misclassified as corn (false positives).

In summary, some of these approaches have demonstrated good performance especially under controlled conditions, however, for uncontrolled conditions, accurate detection remains a challenging problem. Moreover, one of the challenges that most affects accuracy is sunny conditions because this leads to similarities in colour between crops and weeds. For example, when the sunshine is strong, the surface of some leaf types (such as cauliflower or corn leaf), acts as a mirror (specular reflection); as a result, it may be segmented into the wrong category. To address these issues, a novel automatic crop detection algorithm robust against various weather conditions is proposed in Chapter 4 of this thesis.

### 2.4 Use of tracking algorithms

A moving camera may result in issues such as vibration, speed variations, and crops inadvertently covered by soil caused by moving machinery. Moreover, weather conditions,
particularly sunny days, results in detection errors compared to other conditions. These problems translate into loss of detection of moving objects between frames, and may result in e.g. inaccurate operation of automatic weeding or spraying machinery that relies on accurate detection of a plant across multiple frames. These issues can be addressed by applying tracking algorithms. Where multiple objects may be present in a scene, multiple-object tracking algorithms may be used.

Tracking multiple objects in a dynamic scene adds a further level of complexity in computer vision applications. This problem increases in complexity when there are similarities between the different target objects [84, 85], as might be expected in e.g. crop detection applications. As with single object tracking, numerous approaches have been proposed to address these problems and have demonstrated good performance. According to the comprehensive survey presented in [86], there are three common approaches that are used for tracking multiple objects: point tracking, kernel tracking and silhouette tracking.

The Kernel-based tracking approach is robust to occlusion, clutter, and distraction, however, some spatial information in the target is lost, and the method cannot perform well where the target object and its background have similar color. The Silhouette-based Tracking approach is less sensitive to appearance variations, however, it requires training, particularly in shape matching. With respect to point-based tracking, this approach is simple and very useful to track small objects, which can be applied to track objects like a plant that occupies a relatively small region in video sequences. The primary limitations of using point tracking occur in two scenarios: the occurrence of occluded plants where the growth stage is quite late, and where the false detection rate is high. Within the class of point tracking methods, there are three common approaches in the literature: Kalman filtering, particle filtering, and Multiple Hypothesis Tracking (MHT) [87]. According to [88] the Kalman filter is a good option for environments where the number of objects is reasonably small, e.g. on the order of 5 objects. The Kalman filter is widely used because it has excellent properties
2.4 Use of tracking algorithms

including low computation, and the use of a sophisticated recursive and optimal estimator for one-dimensional linear systems with Gaussian error statistics [89]. Other point based tracking algorithms (Particle filter and Multiple Hypothesis Tracking) are computationally more expensive than Kalman filtering [90, 91], which makes the Kalman Filter more attractive for real time processing.

In general, Kalman filtering used for multiple object tracking consists of two stages: the first using the Kalman filter itself and the second using a data association method to create robust object tracks. In [92] Chang et al. used a Bayesian network with Kalman filtering for people tracking in an indoor environment to address the matching issue between multiple people being tracked. With the same conditions that Chang used, in [93] Nguyen et al. applied a Kalman filter in a distributed monitoring system to track moving people. In [94] Vasuhi et al. proposed a Combined Gaussian Hidden Markov Model and Kalman Filter (CGHMM-KF) to detect and track multiple people in different environments. Other researchers have used the Kalman filter with a single camera and the Hungarian algorithm [95] as an assignment method to track multiple objects. For example, in [96] Lütteke et al. track vehicles using this approach. In [97] Nandashri et al. proposed an algorithm to track multiple people by using the Kalman filter, and using the Hungarian algorithm to address the occlusion problem that may occur between individuals. In [98] Yussiff et al. proposed a human tracking algorithm with a Kalman filter and the Hungarian algorithm to link data in the previous frame to the current frame. In [99] Vasuhi et al. used a similar approach to track multiple people in an outdoor environment.

To address the above-mentioned problems and achieve the goal of a robust crop detection system, the detection algorithm proposed in Chapter 4 of this thesis is extended through the addition of object tracking. A multi-object tracking algorithm based on the Kalman filter and Hungarian algorithm is proposed in Chapter 5.
2.5 Dealing with occlusions

Segmentation of overlapping or occluded objects is an extremely difficult task to perform because the overlapping plants appear as one object. This overlapping problem exists in various applications of computer vision such as: biomedical, industrial, agricultural, object search, and navigation. Although many previous studies using image processing and computer vision have demonstrated good plant segmentation performance, the problem of plant overlapping or occlusion has been largely overlooked. In addition, the extensive survey conducted by Slaughter in [54] reported that most studies that they included in their survey have not considered the occlusion between plants. Slaughter also reported that the main source error of plant segmentation (crop/weed) was caused by occlusion between crops and weeds.

An image processing technique that has been used to deal with occlusion in computer vision is the Watershed algorithm [100]. The Watershed algorithm is a segmentation process that aims to separate touching, overlapping, and occluded objects in order to make it possible to recognize and distinguish those objects from each other. In agricultural practice, a few research studies have tackled the occlusion issue between plants and applied the Watershed algorithm for that purpose. For example, In [100] Lee et al. has applied five modified watershed algorithms and compared with the original watershed algorithm to separate tomato leaves and to find out which is the most effective method in terms of less object fragmentation. They discovered that the original watershed algorithm was the least effective, while two of the modified algorithms showed improved performance. However, on the whole, the study concluded that the watershed algorithm may not be a suitable approach for occluded objects, especially where the shapes to be separated are long and thin.

To prevent the over segmentation produced by the original watershed algorithm, in [101] Tang et al. applied a marker-controlled watershed segmentation to extract a single soybean leaf from an image with a background of a cluster of soybean leaves (leaves overlapping
or interfering with each other). The authors applied the watershed algorithm to the three channels of HSI (Hue, Saturation, and Intensity values) images individually. Thus, three segmentation results are obtained as follows: (i) watershed segmentation based on Intensity (ii) watershed segmentation based on Intensity and Hue (iii) watershed segmentation based on Intensity and Saturation. In order to investigate which method performs better leaf extraction, three different images were selected for training and the solidity matrix as defined in [102] was used for evaluation; the segmentation result with the highest solidity was used as final result. The proposed algorithm resulted in 84.87% correct target leaf extraction. The proposed watershed algorithm in [102] was used to address the overlapping between only for leaves and not for whole plants. Although the marker-controlled watershed algorithm performs better segmentation than the original Watershed algorithm, it still requires an initial marker for foreground and background objects.

The algorithms described above suffer from a problem of over-segmentation (over-segmentation is a process by which the objects being segmented from the background are themselves segmented or fractured into subcomponents) and on top of that they require several manipulations and modifications in order to produce good segmentations results. In this project, a novel algorithm based on feature detection from the main stem (main veins of cauliflower and cabbage leaves) and the crop detection algorithm (as given in Chapter 4) is proposed to address partially and highly occluded plants. This algorithm is presented in Chapter 6.

### 2.6 Plant classification using machine learning approaches

While machine vision technology has made great inroads in smart agriculture, according to [54, 103] it is still not fully capable of handling certain real-world issues such as weather variability, presence of shadows in sunny conditions, natural similarities between the target object (weed or crop) and the background, and unexpected changes in camera parameters.
Recent research has attempted to increase performance by applying deep learning technology, with promising results. In [104] McCool et al. applied deep convolutional neural networks (pre-trained model) to perform crop and weed segmentation and reported an accuracy of nearly 94%. In [105] Milioto et al. applied the NDVI color index for vegetation detection and then used a Convolutional Neural Network (CNN) classifier to classify the detected plants into crops and weeds. The algorithm was tested on early and late (two weeks later) growth stages and achieved 99.42 and 99.66% precision on weeds, respectively. In [106] Yalcin et al. used a pre-trained CNN to classify sixteen kinds of plant species. Their approach was tested on acquired images under natural outdoor illumination and compared with an SVM model with different features such as Local Binary Pattern (LAP) and Generalized Search Tree (GIST), and with different kernels such as Radial Basis function (RBF) and polynomial. The CNN achieved good performance (a classification accuracy of 97.47%) compared to SVM model (RBF kernel with LBP and GIST features: 74.92 and 83.88%, respectively; polynomial kernel with LBP and GIST features: 69.81 and 82.29%, respectively).

Other researchers have implemented CNNs for plant disease detection and achieved excellent results. In [107] Sladojevic et al. conducted experiments to detect plant diseases based on leaf image classification using deep neural networks. Their model showed precision between 91% and 98%, for separate class tests, while the overall accuracy of the trained model was 96.3%. In [108] Mohanty et al. investigated the feasibility of using a deep convolutional neural network for detection of plant disease. They used a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions with 14 crop species and 26 diseases. The trained model showed an accuracy of 99.35%. They also used the well-known AlexNet and GoogleNet architectures on different image types (RGB color, Gray scale, leaf segmented), with different training approaches (transfer learning and training from scratch), and different choice of training-testing set distribution.
(train: 80%, test: 20%; train: 60%, test: 40%; train: 50%, test: 50%). Overall, Googlenet with RGB color images, transfer learning, and 80% of the dataset for training and 20% for testing achieved accuracy of 99.3% while Alexnet with the same conditions achieved accuracy of 99.2%.

For plant classification, in [109] Pawara et al. investigated the use of AlexNet and GoogleNet trained from scratch or using pre-trained weights. They used different datasets in their experiments, including an original datasets and data-augmented image datasets for three plant classification problems: Folio [110], AgrilPlant [111], and the Swedish leaf dataset [112]. They also used six different Data-Augmentation (DA) techniques such as Rotation, Blur, Scaling, Contrast, Illumination, and Projective to investigate the classification performance on both pre-trained and fully trained CNNs. The results show that data-augmentation methods are important to obtain higher accuracies for CNN models trained from scratch. For AlexNet and GoogleNet architectures, the combined effects of rotation and illumination, or rotation and contrast are very beneficial, whereas the blur operation does not help to obtain higher accuracies. For AlexNet architecture refined by transfer learning, the scaling DA technique was somewhat helpful, whereas the transfer learning GoogleNet benefits from DA with illumination, but most other DA techniques are not helpful to obtain higher accuracies with the pre-trained CNN architectures.

In most previous work, the dataset used generally contains a single class such as a fruit (apple, banana, grape, jack fruit, orange, papaya, persimmon, pineapple, sunflower, and tulip) or a single plant leaf. In this work, more challenging problems with more than one class, and with images under different illuminations, are investigated. In addition, two well know deep neural networks (AlexNet [113] and GoogleNet [114]) are compared with an SVM model with Bag of Words feature sets based on different approaches: L*a*b colour space, HSV color space, and Speeded-up Robust Features (SURF) were applied and investigated their impact on the classification results as well as compared with AlexNet and GoogleNet.
2.7 Concluding remarks

This chapter has presented an overview of computer vision technologies applied in smart agriculture applications, particular to distinguish crops from weeds. Many approaches have been used for plant segmentation (plant or non-plant), crop detection, and others used for plant classification (crop or weed). Approaches used to distinguish partially occluded plants from weeds or background were also described, and limitations highlighted. In addition, issues that face these approaches such as light changes, shadows, and the similarity in colour between crops and weeds have been highlighted.

This thesis aims to address the above-mentioned problems in order to build a system that is more robust and capable of handling real world conditions. To achieve this goal, the dataset that is used in all experiments contains video clips captured under various conditions. Since a moving camera was used to capture video, issues such as vibration and speed variations are present in the data set, which prompt the use of tracking algorithms. Furthermore, this thesis addresses the problems of detecting occluded plants, examples of which are also contained in the data set acquired for this work. Finally, Chapter 7 investigates the use of deep learning approaches for plant/weed classification.

The next chapter presents a detailed survey and analysis of plant segmentation approaches and discusses their limitations.
Chapter 3

Plant Segmentation

3.1 Introduction

In this chapter, a comprehensive and critical survey of image-based plant segmentation techniques is presented. In this context, “segmentation” refers to the process of classifying an image into plant and non-plant pixels. The segmentation stage requires the segmentation of plant from the background (identifying plant from a background of soil and other residues). Three primary classes of plant extraction algorithms are discussed, namely, (i) colour index-based segmentation, (ii) threshold–based segmentation, (iii) learning-based segmentation. Based on their prevalence in the literature and their good performance in a variety of studies, this review focuses in particular on colour index-based approaches. Therefore, a detailed discussion of the segmentation performance of colour index-based approaches is presented, based on studies from the literature, particularly more recent papers. Finally, some challenges and some opportunities for future developments in this space are considered. The work presented in this chapter has been published in Hamuda et al., "A Survey of Image Processing Techniques for Plant Extraction and Segmentation in the Field", Computers and Electronics in Agriculture, 125, 2016.
Fig. 3.1 shows a block diagram of a general scheme for segmentation, including a broad framework for evaluation of segmentation algorithms. This typically includes a pre-processing stage, followed by the core segmentation stage, which can be done using a variety of approaches (indicated by the “Algorithms” box on the left hand side of Fig. 3.1). Evaluation is typically carried out by comparing the output of the segmentation algorithm with a reference image that is treated as a “gold standard”, by using a suitable performance or quality metric.

Figure 3.1. General scheme for segmentation and its evaluation
3.2 Colour Index-Based Approaches

Several methods have been developed for segmenting crop canopy images. The common segmentation technologies used for this purpose are: colour index-based segmentation, threshold-based segmentation, and learning-based segmentation.

3.2 Colour Index-Based Approaches

Colour is one of the most common methods used to discriminate plants from background clutter in computer vision. Several researchers have used colour to separate plant from soil e.g. colour characteristics were used to distinguish green plants from soil and estimate the leaf area [115–117].

The colour of a region of interest can be accentuated, so the undesired region (soil background region) will be attenuated. For the majority of conventional visible spectrum cameras, the images are output in the conventional RGB colour space. According to [65], converting the RGB values into grayscale did not result in good segmentation because plant and soil background pixels had similar grayscale values. Therefore, in order to demonstrate good segmentation, the RGB space is often converted to alternative colour spaces. Several common green indexes (listed according to date of publication) are as follows:

3.2.1 Normalised Difference Index (NDI)

The normalised difference index was proposed in [59]. They tested three methods to distinguish plant material from soil background in an RGB image. A range of difference indices based on the R, G and B channels was evaluated e.g. G-R, G-B, and G-R/G+R, with the third one demonstrating the best separation of plant from background. This index is applied to all pixels in the image, providing values ranging between -1 and +1, but to display the image, these values must range between 0 and 255. Therefore, the index was further processed by
adding 1 to it and then multiplied by a factor of 128 to provide a grayscale image (0-255). Thus, the final formula for NDI is as follows:

\[
NDI = \left( \frac{G - R}{G + R} + 1 \right) \times 128
\]  

(3.1)

The NDI index produces a near-binary image.

### 3.2.2 Excess Green Index (ExG)

In [60] Woebbecke et al. examined several colour vegetation indices that were derived using chromatic coordinates and modified hue in separating green plant from bare soil (corn residue and wheat straw residue). The colour vegetation indices that were used include:

\[
r - g
\]  

(3.2)

\[
g - b
\]  

(3.3)

\[
g - \frac{b}{r - g}
\]  

(3.4)

\[
2g - r - b
\]  

(3.5)

where \( r, g, \) and \( b \) are the chromatic coordinates:

\[
r = \frac{R^*}{R^* + G^* + B^*} \quad g = \frac{G^*}{R^* + G^* + B^*} \quad b = \frac{B^*}{R^* + G^* + B^*}
\]  

(3.6)

where \( R^*, G^* \) and \( B^* \) are the normalised RGB values ranging from 0 to 1, and are computed as follows:

\[
R^* = \frac{R}{R_{\text{max}}} \quad G^* = \frac{G}{G_{\text{max}}} \quad B^* = \frac{B}{B_{\text{max}}}
\]  

(3.7)

where \( R, G \) and \( B \) are the actual pixel values from the images based on each channel and \( R_{\text{max}} = G_{\text{max}} = B_{\text{max}} = 255 \) for a 24 bit colour image (3*8-bit channels).
Among selected colour vegetation methods, Woebbecke found that the modified hue \((2g - r - b)\), called Excess Green Index (ExG) was the best choice for separating plants from bare soil. This is because ExG provided a clear contrast between plants and soil, and produced near binary images. The ExG index has been widely used and has performed very well in separating plants from non-plants [47, 118, 119, 64].

### 3.2.3 Excess Red Index (ExR)

In [47] Meyer et al. inspired by the fact that there are 4% blue, and 32% green, compared with 64% red cones in the retina of the human eye, introduced ExR method and compared with ExG in the experiment to segment leaf regions from the background. Excess red index was able to separate the plant pixels from background pixels, but it was not as accurate as ExG. The formula for ExR is defined as follows:

\[
ExR = 1.3R - G
\]  

### 3.2.4 Colour Index of Vegetation Extraction (CIVE)

Colour index of vegetation extraction (CIVE) was proposed in [45] based on a study carried out in soya bean and sugar beet fields. This method was proposed to separate green plants from soil background in order to evaluate the crop growing status. The formula for CIVE is as follows:

\[
CIVE = 0.441R - 0.811G + 0.385B + 18.78745
\]

They found that the CIVE has better plant segmentation than Near-infrared (NIR) method because it provides greater emphasis of the green areas.
3.2.5 Excess Green minus Excess Red Index (ExGR)

This method was introduced in [120], and combines two colour indices, namely, Excess Green Index (ExG) and Excess Red Index (ExR). These methods were applied simultaneously to separate plants from the soil and residue, with ExG used to extract the plant region and ExR used to eliminate the background noise (soil and residue) where green-red material (stems, branches, or petioles) may exist. The ExGR is defined as follows:

$$ExGR = ExG - ExR$$  \hspace{1cm} (3.10)

where ExG and ExR are as previously defined.

3.2.6 Normalised Green-Red Difference Index (NGRDI)

The normalised green-red difference index (NGRDI) was proposed in [121] and tested on digital photograph of crops such as corn, alfalfa, and soybeans, which were captured by a digital camera mounted on the bottom of an aircraft fuselage. The method of NGRDI was used to overcome the differences in exposure settings selected by the digital camera when acquiring aerial photography of the field. These differences may cause large differences in colour bands that have the same reflectance. The formula for NGRDI is as follows:

$$NGRDI = \frac{G - R}{G + R}$$  \hspace{1cm} (3.11)

The $G - R$ component is used to discriminate between green plants and soil, and $G + R$ is used to normalise for variations in light intensity between different images.
3.2.7 Vegetative Index (VEG)

This was proposed in [62] to separate plant (cereal and weeds) pixels from soil pixels. The study was conducted under field conditions, and the image was captured by a CCD camera. To achieve segmentation, an RGB image was converted to grayscale by using the following formula:

\[
VEG = \frac{G}{R^aB^b(1 - a)}
\] (3.12)

where \(a\) is a constant value equal to 0.667. Hague found that this transformation demonstrated good contrast between plant and soil. In addition, the VEG has a significant advantage because it is robust to lighting change.

3.2.8 Combined Indices 1 (COM1)

In [62] Guijarro et al. selected four greens indices, \(ExG, CIVE, ExGR,\) and \(VEG\). These methods were applied simultaneously rather than individually to improve segmentation quality, and combined in the following way:

\[
COM1 = ExG + CIVE + ExGR + VEG
\] (3.13)

Guijarro showed that the combined method demonstrated better results than when the approaches were applied separately. The method has been tested in barley and corn fields and demonstrated high reliability under various illumination conditions in outdoor environments.

3.2.9 Modified Excess Green Index (MExG)

Modified Excess Green (MExG) Index was developed in [63] and is defined as follows:

\[
MExG = 1.262G - 0.884R - 0.311B
\] (3.14)
The authors conducted experiments under uncontrolled lighting in real time. The proposed method successfully converted the colour image into grayscale image, which was very easy to binarise with a fast automatic threshold method. The discrimination between plant and soil region was effective because the MExG method was very robust to the changing illumination conditions. They found that MExG method demonstrated better segmentation results than ExG.

### 3.2.10 Combined Indices2 (COM2)

This was introduced in [64] for analysis of maize plants, and is quite similar to COM1 in the combination of three colour Indices: $ExG$, $CIVE$, and $VEG$; $ExGR$ was excluded because it classified the shadow of the maize plant as part of the plant. Normalised difference index was also excluded because it may segment soil regions as plant. The contribution of each selected method is controlled by a weighting factor, with the weights summing to 1. The combined method is defined as follows:

$$COM2 = 0.36ExG + 0.47CIVE + 0.17VEG$$

### 3.3 Evaluation of plant extraction based on colour indices

Segmentation-based colour indices have been widely used as a benchmark by other researchers to evaluate the performance of their proposal methods for improving plant segmentation quality against various lighting conditions and complex background. This section draws together various recent studies that have used colour-index based methods, and in particular, focuses on the comparative performance of these methods. The discussion is constrained by the selection of colour indices that have been selected by the authors of the
different studies considered; however, all of the colour index methods have been evaluated in at least one study.

In previous studies, evaluation has generally been done by calculating the mean and standard deviation of an appropriately defined segmentation quality factor. A high mean value and low standard deviation of segmentation quality factor corresponds to high demonstrated segmentation performance; a value of 1 for the mean, and 0 for standard deviation represents perfect plant segmentation.

### 3.3.1 Verification of individual colour indices

In [116] Meyer and Camargo-Neto compared three green indices, namely, Excess Green minus Excess Red Index (ExGR), Excess Green Index (ExG), and Normalised Difference Index (NDI). The segmentation quality has been tested and compared for both greenhouse and actual field of soybean images. In addition, various backgrounds (bare soil, corn stalks, and wheat residue) were considered. The segmentation quality for each applied method was evaluated according to the approach described in [122]. The input parameters of the evaluation method are two different binary images: one is extracted manually using Photoshop as the annotated "gold standard", and other extracted by the colour index-based method under evaluation. Thus, the evaluation method measures the segmentation accuracy based on how similar the segmented image is to the annotated image and gives the result as ratio of correct classification of pixels. The greenhouse sets were analysed using ExGR with a fixed threshold of zero, NDI with a threshold of zero, and EXG with threshold calculated according to Otsu’s method [123]. The field images were examined with ExGR with a threshold of zero, NDI with an Otsu threshold, and ExG with an Otsu threshold.

A quality factor was used in the experiment as defined by the Automatic Target Recognition Working Group (ATRWG) and [122]. The quality factor is a ratio of correct green plant pixels against the background ones. The mean and standard deviation of the quality factor
for the colour indices considered for green house and field sets are shown in Fig. 3.2 and Fig. 3.3 respectively.

Figure 3.2. Comparison of the performance of selected colour indices: ExGR, ExG+Otsu, and NDI-Otsu under greenhouse conditions. SD is indicted by error bar in the plot [116]

From Fig. 3.2, it can be seen that ExGR with zero threshold presented the best segmentation performance, with mean quality factor of almost 90% with low standard deviation, while the segmentation quality for ExG+Otsu and NDI-Otsu were quite similar, around 50%. According to the results in Fig. 3.3, the performance for ExGR and ExG+Otsu were similar at approximately 90%, while NDI+Otsu has the lowest performance. Overall, ExGR demonstrated very good segmentation quality for the images that were taken under different environments (green house and field conditions), with various backgrounds. In addition, ExGR was superior in plant separation over the ExG and NDI. ExG also performed well in segmentation of plants in field conditions. NDI gave the lowest accuracy.
3.3 Evaluation of plant extraction based on colour indices

Figure 3.3. Comparison of the performance of selected colour indices: ExGR, ExG+Otsu, and NDI+Otsu under actual field conditions. SD is indicted by error bar in the plot [116]

3.3.2 Machine learning approaches

In [67] Zheng et al. proposed an algorithm using a Mean-Shift method and Back Propagation Neural Network (MS-BPNN) to improve the segmentation quality of plant. To assess the performance of the algorithm, two index-based methods (ExG and CIVE) were used as a benchmark. The study was conducted in outdoor environments, including variety of plant species, different illuminations, and different soil types. Before evaluating the segmentation performance, each region in the test image of size 2x2 pixels was labelled by hand with ‘1’ for green region and ‘0’ for background and then compared that with segmented image. The segmentation performance was assessed based on the mis-segmentation rate for green and background region, in particular the minimum (Min), median (Med), and maximum (Max) values were evaluated. Moreover, the mean running time per image (T) was used to evaluate the speed of MS-BPNN method. Four different image types were tested, including: green
vegetation with shadow (GVS), green vegetation without shadow (GV), non-green vegetation with shadow (NGVS), and non-green vegetation without shadow (NGV).

For GV and GVS images, the MS-BPNN method gave the lowest Min, Med, and Max of mis-segmentation for GV: 0.19%, 2.53 %, and 5.34%, respectively, and for GVS: 15.72%, 18.23%, and 34.13% respectively. Of interest here are the performance figures for CIVE and ExG shown in Fig. 3.4.

Figure 3.4. Comparison of mis-classification of CIVE and ExG for GV and GVS image types [67]

CIVE has lower mis-segmentation rate for GV images than ExG. Both ExG and CIVE based have very high mis-segmentation rate for GVS images. In general, ExG is more accurate for extracting plants with shadow than CIVE. For NGV images, the MS-BPNN method gives values of Min, Med and Max of mis-segmentation of 2.12%, 3.81%, and 5.26% respectively. These values are higher than those for NGVS, where the algorithm gives values of 0.12%, 1.28%, and 4.93%. Again, what is of interest here is the performance for the
3.3 Evaluation of plant extraction based on colour indices

colour indices considered; the mis-classification performance for CIVE and ExG for NGV and NGVS images is shown in Fig. 3.5.

Figure 3.5. Comparison of mis-classification of CIVE and ExG for both image types (NGV & NGVS) [67]

ExG demonstrated good segmentation results for NGVS images, whereas the CIVE exhibited poorer performance. In addition, ExG has shown lower mis-segmentation rate for NGV than the proposed MS-BPNN method.

Of further interest is the fact that although Zheng’s proposed method demonstrated better segmentation performance overall than both ExG and CIVE, it gave the longest average computing time for both types of tested images, which were: 91.9s per image and 10.8s per image. Comparing the mean running time for colour index-based methods, ExG was given slightly lesser than CIVE for both types of tested images: (3.8s per image and 0.5s per image), (3.9s per image and 0.6s per image) respectively.
3.3.3 Combination of machine learning approaches and colour indices

In [68] Zheng et al. introduced another method to improve the quality of crop image segmentation. The proposed method was based on the combination of two methods: one based on Mean Shift (MS) and another based on Fisher linear discriminant (FLD). The images that were used in the study were taken from different soybean fields, under actual field conditions, and at different times of day. As a benchmark, three colour index-based methods (NDI, ExGR, and CIVE) were compared with the MS-FLD method to evaluate its performance. The method of Otsu was used to determine a threshold for use with all colour indices images for binarisation. In addition, an averaging filter was used to remove noise. To evaluate the performance of the MS-FLD method and the three colour index-based methods, each of the test images was labelled manually with ‘1’ (white) for the green region, and with ‘0’ (black) for background region. The study has shown that MS-FLD obtained the highest average segmentation rate, at 97.98%. The performance of the colour index methods (in terms of average value of correct segmentation rate) are presented in Fig. 3.6.

It can be seen that all three colour indices performed well, and average performance is quite high at approximately 80%. However, according to Zheng, the colour index-based methods were not stable for all tested images; some resulting images showed that NDI and CIVE gave better segmentation than ExGR, whereas others showed that ExGR produced better segmentation than NDI and CIVE.

Again, it is of interest to compare the computation time of the MS-FLD method with the simpler colour-index methods. While MS-FLD demonstrated better segmentation performance than colour index-based methods, its average running time was higher than that obtained by vegetation index-based methods: 3.3906s per image and 0.0156s per image (averaged over the three colour indices), respectively.
3.3 Evaluation of plant extraction based on colour indices

3.3.4 Verification of combinations of colour indices

In [124] Guijarro et al. tested four green colour indices (ExG, CIVE, ExGR, and VEG) individually and simultaneously (using the COM1 combined index described above) to assess their performance for better automatic segmentation of plants. As noted earlier, the study by Guijarro found that when used individually, these indices may create either over-segmentation or under-segmentation results; when combined through COM1, these problems can be overcome. The study was conducted in barley and corn fields under various illumination conditions. Two scenes were taken into account: one scene contained plants and soil without sky and another one contained plants, soil, and sky. The combination method was proposed to increase the contrast between plant and soil, so the probability of distinguishing between plant and background image is increased. In order to accomplish this goal, the contrast was measured based on the gray level histogram (minimum uniformity). The uniformity was computed for each green image \( U_{GK} \), and the weight was obtained
for each one ($W_{GK}$), where $k = ExG, CIVE, ExGR, VEG$. Besides, the combined greenness ($G$) was computed and the mean threshold was chosen instead of the Otsu threshold to separate the plant region from the background. The average error in pixel classification for segmentation of green areas for each colour index based method is displayed in Fig. 3.7.

![Comparison of average error of greenness segmentation for COM1, CIVE, ExGR, ExG, and VEG](image)

Figure 3.7. Comparison of average error of greenness segmentation for COM1, CIVE, ExGR, ExG, and VEG [124]

The results of this study showed that the combination method (COM1) provided the lowest percentage of average error for greenness segmentation, whereas VEG showed the highest. Each of CIVE, ExGR, and ExG method gave similar values of average error.

Guijarro calculated the average weight of the four selected indexes over the set of the 240 images to find out their contributions to the average percentage error. The average weights were given: 0.12, 0.25, 0.30, and 0.33 for $W_{G-VEG}$, $W_{G-ExG}$, $W_{G-ExGR}$, and $W_{G-CIVE}$ respectively. He found that there was a reverse correlation between the obtained average weights and the percentage of error for greenness. For example, CIVE has given the highest average weight (0.33) and resulted in the lowest average percentage error over the others, whereas
VEG has given the lowest average weight (0.12) and caused the highest percentage error over the other methods.

In conclusion, if the colour indices are applied simultaneously, they produce better green-ness segmentation quality rather than when they are applied separately. CIVE’s contribution to the combined method is greater than any other method while VEG was the lowest. Both of ExG and ExGR have nearly the same contribution. There were one primary disadvantage associated with the combined method, which was increased computational time.

### 3.3.5 Colour spaces approaches compared to colour indices and machine learning

In [70] Yu *et al.* proposed a new method for crop segmentation based on colour segmentation called Affinity Propagation-Hue Intensity (AP-HI). Five other algorithms were compared with it to judge its performance. Among these, three colour index methods, namely, ExG, CIVE, ExGR were used with Otsu threshold and a fourth (VEG) was used with mean threshold method. The fifth method was a supervised learning algorithm called environmentally adaptive segmentation algorithm (EASA) [65]. Two experiments were carried out in two maize fields in China under different circumstances to identify the growth stages of maize. The image samples were acquired under various illumination conditions such as overcast, cloudy, and sunny days. Difficult backgrounds such as shadow, straws, pipes, and other equipment were included. The efficiency of each algorithm was evaluated through computing the mean and standard deviation (SD) of the quality factor defined in [125], based on mis-classification error.

The results of the study have shown that AP-HI gave the highest performance, at 96.68%. The performance of EASA was in second place; it outperformed the colour index-based algorithms with mean of plant extraction equal to 93.20%. The performance of colour index approaches can be seen in Fig. 3.8.
It can be seen that ExG demonstrated the highest mean of greenness segmentation over the remainder of selected colour indexes, whereas CIVE has shown the lowest at 68.9%. ExGR and VEG showed similar performance.

In conclusion, all colour indices demonstrated good adaptability in conditions of changing illumination (up to a certain degree of illumination) and complex environments, however CIVE did not perform as well as the other indices.

The AP-HI method was in dealing with various environment conditions and complex background up to certain degrees. However, this method has limitations especially during day light where some surfaces of the maize leaves acted like mirrors and reflected light.

In [69] Guo et al. introduced a new approach called Decision Tree based Segmentation Model (DTSM) for effective segmentation of vegetation from plant images. The study was conducted in wheat fields in Japan and the test images were taken under various light conditions. Three colour indices (ExG, MExG, and ExGR) were used as a benchmark to
3.3 Evaluation of plant extraction based on colour indices

evaluate the performance of the proposed method. The accuracy of the segmentation methods is assessed by the same method that used in [116]. Moreover, two tasks of segmentation quality were adopted in the study: one was based on the both plants and background regions include plant pixels or background pixels which was denoted as $Q_{seg}$ and another was based only on the plant region (including only plant pixels) and was denoted as $S_r$. The training process was carried out based on acquired images which were taken over a period of two years under different illumination conditions (sunny and non-sunny) in 2011 and in 2012 (henceforth referred to as Data-2011 and Data-2012). Otsu’s method was used with ExG and ExGR images for thresholding, while a zero threshold was applied with MExG.

DTSM outperformed the three colour indices on segmentation quality. The mean value of $Q_{seg}$ was 80.6% for the Data-2011 data set and 76.7% for the Data-2012 data set. In addition, DTSM gave the best mean of green segmentation quality ($S_r$) compared to the colour indices methods, with 83.3% for Data-2011 and 83.1% for Data-2012. The $Q_{seg}$ and $S_r$ for applied colour indices under non-sunny and sunny conditions are presented in Fig. 3.9 and Fig. 3.10 respectively.

According to the results presented in Fig. 3.9, the means of $Q_{seg}$ and $S_r$ for ExGR for the Data-2012 set are slightly higher than those of MExG, whereas the means of $Q_{seg}$ and $S_r$ of MExG were higher than those of ExGR for the Data-2011 data set. The ExG method produced the lowest segmentation quality ($Q_{seg}$ & $S_r$) compared to ExGR and MExG in both years. According to the results presented in Fig.3.10, the means of $Q_{seg}$ and $S_r$ for ExGR in both data sets under sunny condition were higher than the other colour indices.

In conclusion, the three colour indices considered have better segmentation qualities for $S_r$ than $Q_{seg}$ under both conditions. Comparing the results in Fig. 3.9 to those of in Fig. 3.10, the colour indices performed better quality segmentation under non-sunny conditions than under sunny conditions. This suggests that colour index methods may in general perform more poorly under sunny conditions. The advantage of the DTSM algorithm proposed in
Figure 3.9. Comparison of the segmentation quality ($Q_{seg} \& S_r$) of plant extraction for ExGR, MExG, and ExG for two different data sets under non-sunny conditions [69]

Figure 3.10. Comparison of the segmentation quality ($Q_{seg} \& S_r$) of plant extraction for ExGR, MExG, and ExG for two different data sets under sunny conditions [69]
3.3 Evaluation of plant extraction based on colour indices

[69] is that no threshold adjustments are required for plant segmentation, unlike colour index methods except those using Otsu’s method. However, a disadvantage of DTSM is that it relies on training data.

In [126] Bai et al. introduced a new method for crop segmentation based on the CIE Lab colour space, using morphological modelling. The study was conducted in a rice paddy field in China, and the images were taken under various conditions for different growth status of rice plant. To verify the robustness of crop segmentation using the method under complex illumination, it was compared with six plant segmentation methods that included: three colour index based- methods (ExG with Otsu threshold, ExGR, and CIVE); Environmentally Adaptive Segmentation Algorithm (EASA) [65]; colour image segmentation method using Genetic Algorithm with HSI colour space (GAHSI) [53]; and colour image segmentation method based on Affinity Propagation–Hue Intensity (AP-HI) [70]. Two well-known skin segmentation methods (segmenting colour pixels as either skin or non-skin classes) were also applied: Gaussian Mixture Modeling (GMM) [127, 128] and the Hue–Saturation–Intensity and B-Spline curve fitting method (HSI&B-Spline) [129]. Two approaches were used to measure the segmentation quality for the applied methods: one defined by the Automatic Target Recognition Working Group (ATRWG) [130] as noted earlier, and one described in [125].

The segmentation performance for the referred methods was evaluated in three ways. Firstly, the tested image were taken under different light conditions and used ATRWG metric to measure the performance. The study showed that Bai’s algorithm demonstrated the highest segmentation performance, with mean value of 87.2%, and standard deviation of 3.8%. The second highest performance of segmentation quality was given by GMM method with mean of 83.9%, and standard deviation of 7.2%. The performance of the three colour indices considered is shown in Fig. 3.11.
It can be seen that CIVE demonstrated the highest mean for segmentation quality whereas ExGR gave the lowest mean of segmentation quality. The method of ExG & Otsu also demonstrated good segmentation quality.

Secondly, the test images were sorted based on their imaging conditions; cloudy, overcast, and sunny. Each of set images was evaluated separately by using ATRWG metric. The experiment showed that Bai’s method also gave better segmentation quality under different sky conditions than the other methods with mean of segmentation quality of 85.7%, 86.0%, and 88.6% for cloudy, overcast, and sunny conditions, respectively. The segmentation quality performance for ExGR, ExG+Otsu, and CIVE are displayed in Fig. 3.12.

As can be seen from above figure, the best overall segmentation quality under the three conditions was obtained by CIVE, whereas the worst was obtained by ExGR. The ExG method demonstrated reasonably good segmentation quality under all three conditions.
3.3 Evaluation of plant extraction based on colour indices

Thirdly, the test images were taken under different lighting conditions and the method defined in [125] was used to measure performance. In this evaluation, the proposed method improved the mean of segmentation quality and reached up to 96.0% with standard deviation of only 1.5%.

The EASA, GAHSI, AP-HI, GMM, and HIS&B-spline gave mean of segmentation qualities as: 93.9%, 92.4%, 92.5%, 95.2%, and 87.7% respectively. The performance of the three colour indices considered is shown in Fig. 3.13.

It can be seen that CIVE demonstrated the highest mean for segmentation quality among applied colour indexes; CIVE also gave better performance that the other methods such as EASA, GAHSI, AP-HI, and HIS&B-spline. ExGR gave the lowest mean of segmentation quality. The method of ExG with Otsu threshold also demonstrated very good segmentation quality, at 91.80%.
In [71] Bai et al. proposed a new plant segmentation approach based on Lab colour space and a clustering method, namely, Particle Swarm Optimization (PSO) based k-mean. The images that were used in the study were captured under real conditions, in rice and cotton fields. Three segmentation approaches (ExG and Otsu, ExGR, and EASA) were used for benchmark purposes. Also, two methods that had been previously applied for segmenting human skin (GMM and ColourHist) were applied to assess the performance of the Bai’s method. The ATRWG method was applied to evaluate the quality of segmentation for each segmented image, and means and standard deviations of the segmentation accuracies were calculated. According to results of the study, Bai’s method obtained the highest performance of segmentation quality over the others, achieving 88.1% for the mean and 4.7% for standard deviation. The method of GMM demonstrated very good performance close to Bai’s method,
3.3 Evaluation of plant extraction based on colour indices

with mean of 86.9% and standard deviation of 6.9%. The method of ColourHist demonstrated good performance, 82.1% for the mean and 6.4% for standard deviation.

The method of EASA provided also good segmentation results, with mean of 80.2% and standard deviation of 7.8%. For the two colour index methods considered, ExGR and ExG with Otsu threshold, the means and standard deviations of are shown in Fig. 3.14.

![Figure 3.14. Comparison of mean and standard deviation of plant extraction for ExGR and ExG+Otsu. SD is indicted by error bar in the plot [71](image)](image)

As it can be seen from the Fig. 3.14, ExG with Otsu threshold demonstrated higher mean of segmentation quality than ExGR. The performance of colour index-based methods was poorer than the other algorithms in the study. Bai et al suggested this poor performance was because ExGR and ExG with Otsu threshold usually resulted in over-segmentation or under-segmentation. A disadvantage of Bai’s method is that it requires a number of processing steps, which may affect real time application.
3.3.6 Use of colour indices with images taken from the air

In [131] Torres-Sánchez et al. measured the accuracy of vegetation fraction (VF) mapping for wheat fields at different numbers of growing days after sowing from 35 to 75. The images of the fields were taken by a camera mounted on a UAV at different flight altitudes (30m, 60m). Six colour indices (ExG, ExGR, CIVE, Woebbecke Index as given in equation (3.4) [60], NGRDI, VEG, and two combined colour indices, COM1 [124] and COM2 [64], were applied to evaluate the VF mapping. The VF is the percentage of pixels classified as vegetation in a given area. The mean accuracy (A) and standard deviation (SD) were calculated for every index based on three factors: threshold, flight date, and altitude. In addition, the coefficient of variation was calculated to get the best average accuracies of every vegetation index along the six tested flight dates. According to the results of the study, the highest mean accuracy was obtained from the images that were captured at 30m flight altitude, so only results obtained at that altitude are considered here. The mean and standard deviation of all colour indices are shown in Fig. 3.15.

ExG gave the highest mean accuracy over the other Vegetation Index (VI) methods, at 90.20%, however, most of the other indices gave quite similar levels of performance. CIVE has the lowest mean accuracy over the other VI methods, at 77.16%.

3.3.7 Performance in the presence of strong illumination

In [132] Ye et al. introduced a novel method to improve the quality of crop image extraction under strong illumination conditions such as shadow and highlighted region due to sunshine. Ye suggested reasons for misclassification of crop extraction under a variety of illumination such as cloudy, sunny, and over-sunny weather. In cloudy weather, there are two factors that cause classifying of soil pixels as crop. One is the reduction in the red component in the image because of lack of illumination, the other one is that the colour of soil is close to dark green. In sunny weather, shadows generated depending on the relative position of
3.3 Evaluation of plant extraction based on colour indices

Figure 3.15. Comparison of the mean and standard deviation of plant extraction for ExG, VEG, COM1, COM2, ExGR, NGRD, WI, and CIVE of images were captured at 30m flight altitude [131]

the sun and the object cause classification of shadow pixels as plant pixels. In over-sunny weather, the dense sunshine produces specular reflection (white light spots) in the leaf or soil. This leads to mis-classifying of those pixels. Ye proposed a segmentation method based on Probabilistic Superpixel Markov Random Field (PFMRF). This was based on the assumption that colour gradually changes of hue intensity between highlighted areas of crops and neighbouring non-highlighted areas. The images that were used in the experiment were taken from two different crops (cotton and corn) at different stages of growth, under actual field conditions including on dark and bright days. To evaluate the performance of the PFMRF method, seven common algorithms were selected for comparison. Among them, four colour index-based methods (ExG, NDI, VEG, and CIVE) were applied. In addition, two learning-based segmentation methods (EASA and HI-AP) were applied. Hue Intensity
and Probabilistic Super-Pixel Markov Random Field (HI-MRF) proposed in [133] was also applied. A performance measure ($\lambda$) [134] based on the misclassification error was used.

The results of the study showed that the proposed PFMRF method gave the highest performance over all applied algorithms, with mean of 92.29%, and with the lowest SD, at 4.65%. The performance of HI-AP was in second place, at 88.52%, while the performance of EASA was almost the same, at 88.42%. The performance of HI-MRF was also high, at 87.74%. The performance of the four colour index-based methods can be seen in Fig. 3.16.

![Figure 3.16. Comparison of mean and standard deviation of crop extraction for ExG, NDI, VEG, and CIVE. SD is indicted by error bar in the plot [132]](image)

As it can be seen from the Fig. 3.16, CIVE demonstrated the best performance among colour index-based methods, at 86.8%, whereas ExG showed the lowest. Both NDI and VEG demonstrated good performance. All colour index-based methods demonstrated good adaptability in light changing, but failed when shadow and highlight conditions occur.

A summary of the established colour index-based segmentation methods, highlighting their primary advantages and disadvantages, is presented in Table 3.1.
### Table 3.1. Comparison of plant segmentation methods based colour indices

<table>
<thead>
<tr>
<th>Author</th>
<th>Method</th>
<th>Description</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
</table>
| [59]   | NDI    | Normalised Difference Index | 1) Easy to compute  
2) Somewhat robust to lighting, except for extreme values | 1) Does not perform well when the light is very high or very low  
2) Many false positives. |
| [59]   | ExG    | Excess Green Index | 1) Easy to compute  
2) Widely used  
3) Low sensitivity to background errors and lighting conditions  
4) Showed good adaptability in outdoor environment | 1) Does not perform well when the light is high or low. |
| [47]   | ExR    | Excess Red Index | 1) Easy to compute  
2) Although it relies only on red component, it still extracts green pixels  
3) Segment soil texture | |
| [45]   | CIVE   | Colour Index of Vegetation Extraction | 1) Low running time  
2) Showed good adaptability in outdoor environment | 1) Performs poorly when light is weak or strong  
2) Has poor adaptability with shadow. |
| [61]   | ExGR   | Excess Green Minus Excess Red Index | 1) Showed good adaptability in outdoor environments  
2) Can do two tasks: extracting green by ExG and eliminating background noise by ExR | 1) Does not perform well when the light is high or low  
2) Segments the pixel of shadow as plants (over-segmentation). |
| [121]  | NGRDI  | Normalised Green–Red Difference Index | 1) Reduces the differences in exposure settings selected by the digital camera  
2) Consists of two components (8): one is used to discriminate between green plants and soil, and other is used to normalise for variations in light intensity between different images | 1) Does not perform well when the light is high or low  
2) Limited use. |
| [62]   | VEG    | Vegetative Index | 1) Invariant to the colour temperature of a black body illuminant  
2) Insensitive to the amplitude of the illumination  
3) Requires a single threshold | 1) Does not perform well when the light is high or low  
2) Complex to implement. |
| [124]  | COM1   | Combined ExG, ExGR, and CIVE and VEG indexes | 1) Showed very good adaptability in outdoor environment  
3) Requires a single threshold | 1) Increase of computational time  
2) Does not perform well when the light is high or low  
3) Segments shadow as part of plant because of CIVE. |
| [63]   | MExG   | Modified Excess Green Index | 1) Showed very good adaptability in outdoor environment | 1) Does not perform well when the light is high or low  
2) Increased computational time, but less than COM1  
3) Segments shadow as part of plant because of CIVE. |
| [64]   | COM2   | Combined ExG, CIVE, and VEG indexes | 1) Showed very good adaptability in outdoor environment | 1) Does not perform well when the light is high or low  
2) Does not perform well when the light is high or low. |
3.4 Other Segmentation Approaches

The previous two sections have considered colour index-based segmentation methods in some detail, including their performance. This section briefly discusses some of the other segmentation approaches that have been recently proposed, in particular based on thresholding, and machine learning.

3.4.1 Threshold–Based Approaches

Threshold techniques that are applied in plant/weed detection based on image segmentation have generally assumed a two class problem, namely, plant vegetation class and soil background class. Thresholding is generally applied to a transformation of the original image in order to determine the class; for example, many of the colour-index based approaches considered earlier used either zero threshold or a threshold based on Otsu’s method. However, other more sophisticated approaches for threshold selection exist. Choosing the proper threshold plays an important role in segmentation. For example, if the threshold value is set too high, some important regions (plant pixels) may be merged with other regions (background pixels) which leads to under-segmentation, while a threshold that is set too low may lead to over-segmentation.

Thus, numerous researchers have applied different threshold technique to address these problems. These techniques are given as follows. Dynamic thresholding was applied in [135]. Hysteresis thresholding was applied in [136]. Fixed threshold is also a technique which was utilized in many studies such as [137, 138]. In [139] Tellaeche et al. applied entropy of a histogram to distinguish plant vegetation pixels from soil pixels. Otsu’s method is a threshold technique widely used in many applications of image processing based-segmentation. According to a survey carried out in [140] to compare the segmentation accuracy of nine threshold methods, Otsu’s method demonstrated the highest accuracy value over the others. This has inspired numerous researchers to utilize it particularly in plant
and weed segmentation. Otsu’s method was applied in [141] to segment tomato seedlings from background. It was also applied in [142] to separate the plant vegetation from the background; it was preferred to remove the noise pixels instead of using morphological dilation as it does not require as much computation as dilation operations. In [143, 144] Gebhardt et al. introduced an algorithm to segment weed leaves from grassland by converting RGB images into grayscale image intensity and then calculating local homogeneity images and obtaining a homogeneity threshold value to derive binary images. Finally, morphological opening was used to eliminate the remaining blades of grass in the binary images.

In [117] Kirk et al. introduced a new algorithm for pixel classification (plant or soil pixels) to work under a variety of illuminations. The algorithm is based on greenness and intensity pixels, which are derived from the combination of R and G pixel values, and an automatic threshold was applied based on the assumption of two Gaussian distribution functions of intensities; the Gaussian distribution with the lower mean represented the soil distribution and the one with the higher mean represented plant vegetation distribution. In [145] Jeon et al. applied another threshold technique to automatically segment plant pixels from soil pixels based on transformed RGB image. The output image from this transformation is an intensity image and this that can outline a plant region of interest, from which the segmentation could be performed with a suitable threshold.

In [116] Meyer and Camargo Neto have examined the segmentation quality for some colour indices with using automatic Otsu threshold and zero threshold methods; in particular, ExG and NDI were tested with an Otsu threshold, and ExGR was tested with zero threshold. The results showed that the fixed zero threshold was sufficient for binarisation of ExGR images, so the Otsu’s method was not required. Two different automatic threshold approaches were used and evaluated for vegetation segmentation in [124]: one was Otsu’s method and the other was based on mean intensity. The results showed that the Otsu threshold produced under-segmentation, i.e. some green pixels were not identified. Besides, it was slower than...
the mean intensity method. Therefore, the automatic threshold adjustment approach (the mean intensity value) was adopted in the study as it produced fast and robust segmentation. On the other hand, the mean intensity value was not found suitable in [63] to binarise a gray image which was generated by the combined colour indexes, because its value was less than the threshold value received with Otsu method. Therefore, the combination of Otsu and a morphological operation were used instead. The advantages and disadvantages of threshold based-approaches are summarised in Table 3.2.

3.4.2 Learning-Based Approaches

Although the colour-based approaches and threshold-based approaches have demonstrated promising segmentation results, there are a few cases where they do not perform well, particularly in sunny and overcast conditions. As a result, several studies have investigated more sophisticated approaches, including supervised and unsupervised machine learning approaches with simple transformation of colour features such as HIS, LUV, and L*a*b, or with colour index methods to extract the plant pixels from the background, in an attempt to improve the segmentation under variety of illumination conditions. For instance, in [120] Meyer et al. applied an unsupervised learning approach called fuzzy clustering to extract the area of interest from ExG and ExR images.

Several researchers have also proposed supervised learning approaches. In [65] Tian and Slaughter proposed the Environmentally Adaptive Segmentation Algorithm (EASA) and applied to normalized RGB images of outdoor fields to detect plants. Later, in [66] Ruiz-Ruiz et al. applied EASA with hue-saturation (HS) and only hue H, instead of the full RGB colour space information, to produce robust and fast plant image segmentation under complex field conditions. In [67] Zheng et al. proposed a supervised mean-shift algorithm with Back Propagation Neural Network to classify images into plant and non-plant regions. The features used in the algorithm were derived from RGB and HSI colour space. In [68]
### 3.4 Other Segmentation Approaches

#### Table 3.2. Comparison of threshold based segmentation methods

<table>
<thead>
<tr>
<th>Author</th>
<th>Method</th>
<th>Description</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>[135]</td>
<td>Dynamic threshold</td>
<td>Thresholds are dynamically set according to local rather than global characteristics. The approach is to partition the image into sub-images of size m x m pixels, and then choose a threshold for each sub-image.</td>
<td>1) Insensitive to shading or gradually changing illumination</td>
<td>1) Increase in computation time since it requires several steps.</td>
</tr>
<tr>
<td>[136]</td>
<td>Hysteresis threshold</td>
<td>Includes two thresholds, high and low. This leads to the creation of 3 classes: below low threshold (to be removed), above high threshold (to be retained), and between low and high thresholds (to be retained only if connected to a pixel above high threshold).</td>
<td>1) Effective in handling overlap between the modes in the histogram of an image</td>
<td>1) Various morphological operations were required to improve the segmentation, increasing computation</td>
</tr>
<tr>
<td>[137, 138]</td>
<td>Fixed threshold Entropy of a histogram</td>
<td>Empirical threshold selection Can be chosen through the peaks of gray-level histogram of an image</td>
<td>1) Simple</td>
<td>1) Various morphological operations were required to improve the segmentation, increasing computation</td>
</tr>
<tr>
<td>[139]</td>
<td>Otsu threshold</td>
<td>Based on finding the threshold that minimises the weighted within-class variance</td>
<td>1) Automatic method</td>
<td>1) Sensitive to light changes</td>
</tr>
<tr>
<td>[143, 144]</td>
<td>Homogeneity threshold</td>
<td>Local homogeneity is calculated for an image pixel and used to obtain a homogeneity threshold value to derive binary images.</td>
<td>1) Helpful in recognizing small objects</td>
<td>1) Increase in computation time since it requires several steps</td>
</tr>
<tr>
<td>[117]</td>
<td>Automatic threshold</td>
<td>The threshold value is selected based on Gaussian distribution functions of intensities; the Gaussian distribution with the lower mean represents the soil and the one with the higher mean represents plant vegetation.</td>
<td>1) Good in handling light changes</td>
<td>1) Increase in computation time since it requires several steps</td>
</tr>
<tr>
<td>[145]</td>
<td>Automatic threshold</td>
<td>The threshold value is determined by dividing the pixel distribution of the image into two groups by a pixel value ranging from 1 to 255. The pixel value that minimises the variance sum of two groups was used as the threshold value for each image</td>
<td>1) Provides adaptive segmentation 2) Automatic method</td>
<td>1) Requires threshold adjustment to update segmentation limit especially for high plant density</td>
</tr>
</tbody>
</table>

*Note: Numbers in square brackets refer to the corresponding references at the bottom of the page.*
Zheng et al. applied a supervised mean-shift algorithm, but with Fisher Linear Discriminant to separate green from non-green plant; the colour space used in the algorithm was LUV instead of RGB and HSI. Support vector machines (SVM) have been applied as the learning method to classify between masked and unmasked plant regions [64]. To address illumination problem such as shadow and specularly reflected regions, in [69] Guo et al. introduced a new method based on decision tree model to segment plant region form the background in RGB images.

The advantages and disadvantages of learning based-approaches are summarised in Table 3.3.

3.5 Discussion

According to some of the studies considered above, while colour index-based methods achieve good performance in many situations, and are generally computationally efficient compared to alternative approaches, they have some limitations: they may result in over-segmentation (excessive green) in one application and under-segmentation in another application, especially when a single index is applied by itself. This varies considerably with imaging conditions, and the fact that the same test data are not used in all studies makes direct comparison more difficult. Few comparative studies have been carried out using a common set of test data. One somewhat recent example was carried out in [116], to compare three green indices, namely, ExGR, ExG, and NDI.

The advantages and disadvantages of colour index-based methods can be summarised as follows:

**Advantages:**

- Simple methods that are easy to understand and implement.
### Table 3.3. Comparison of learning based segmentation methods

<table>
<thead>
<tr>
<th>Author</th>
<th>Method</th>
<th>Description</th>
<th>Colour model</th>
<th>Task</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>[65]</td>
<td>EASA</td>
<td>Environmentally Adaptive Segmentation Algorithm</td>
<td>RGB space</td>
<td>Detect plants</td>
<td>1) Adapts to most daytime conditions in outdoor fields</td>
<td>1) Only 45–66% of all the cotyledons were recognized under partially cloudy and overcast conditions 2) It requires sufficient training data to obtain good segmentation results.</td>
</tr>
<tr>
<td>[120]</td>
<td>FC</td>
<td>Fuzzy Clustering</td>
<td>RGB space</td>
<td>Extract the plant region of interest from ExG and ExR images</td>
<td>1) Identifying green plants from soil and residue</td>
<td>1) When plant pixel coverage is less than 10% in the image, there apparently is not enough colour information to cluster them.</td>
</tr>
<tr>
<td>[66]</td>
<td>EASA</td>
<td>Environmentally Adaptive Segmentation Algorithm</td>
<td>Hue-saturation (HS) and only hue H</td>
<td>Plant image segmentation under complex field conditions</td>
<td>1) Reduced the computation time 2) It is more robust to a variety of illumination than the [65]</td>
<td>1) It is not effective to segment plants at early growing stage where the cotyledons start to appear</td>
</tr>
<tr>
<td>[67]</td>
<td>MS-BPNN</td>
<td>Mean-shift algorithm with Back Propagation Neural Network</td>
<td>RGB and HSI colour space</td>
<td>Classify between plant and non-plant region</td>
<td>1) Demonstrate good segmentation performance under different illuminations</td>
<td>1) It suffers from long run time 2) Suffers from low segmentation rate on the green parts with shadows.</td>
</tr>
<tr>
<td>[68]</td>
<td>MS-FLD</td>
<td>Mean-shift algorithm with Fisher Linear Discriminant</td>
<td>LUV space</td>
<td>Separate green from non-green vegetation</td>
<td>1) No longer suffers from the low segmentation rate on the green parts with shadows</td>
<td>1) It suffers from long run time.</td>
</tr>
<tr>
<td>[64]</td>
<td>SVM</td>
<td>Support Vector Machines</td>
<td>RGB space</td>
<td>Classify between masked (soil and other materials) and un-masked (plants) plant regions</td>
<td>1) The method is able to identify plants (weeds and crops) when they have been contaminated with materials coming from the soil, due to artificial irrigation or natural rainfall</td>
<td>1) Relies on other steps (threshold).</td>
</tr>
<tr>
<td>[69]</td>
<td>DTSM</td>
<td>Decision Tree based Segmentation Model</td>
<td>RGB space</td>
<td>Segment the vegetation form the background</td>
<td>1) Addressing illumination problem such as shadow and specularly reflected regions 2) Not requiring a threshold adjustment for each image</td>
<td>1) It relies on training data.</td>
</tr>
<tr>
<td>[70]</td>
<td>AP-HI</td>
<td>Affinity Propagation-Hue Intensity</td>
<td>Hue-Intensity(HI)space</td>
<td>Separate the pixels of a crop and background under light conditions and complex environment</td>
<td>1) Robust and not sensitive to the challenging variation of outdoor luminosity and complex environmental elements</td>
<td>1) Misclassifying highly lighted region in leaves.</td>
</tr>
<tr>
<td>[126]</td>
<td>MM</td>
<td>Morphology Modeling</td>
<td>Lab colour space</td>
<td>Distinguishes the crop and background pixels under complex illumination conditions</td>
<td>1) Robust to the variation of illumination in the field</td>
<td>1) Despite utilizing different sizes of structure elements in the training phase, it did not give a significant improvement; the mean of segmentation qualities of MM was 87.2%. 1) It suffers from long run time as it depends on many processing steps.</td>
</tr>
<tr>
<td>[71]</td>
<td>PSO-MM</td>
<td>Particle Swarm Optimisation clustering and Morphology Modelling</td>
<td>Lab colour space</td>
<td>Distinguishes the crop and background pixels under complex illumination conditions</td>
<td>1) Robust to variation of illumination in the field</td>
<td>1) It suffers from long run time as it depends on many processing steps.</td>
</tr>
</tbody>
</table>
• Easy to modify their formulas to create a new colour index.

• Generally do not require training.

• Generally require low computation which makes them suitable for real time use.

• They are effective in normal condition where the light is neither very high nor very low.

• Some of the colour index-based methods have shown results that are comparable to other more sophisticated methods e.g. see study in [126].

**Disadvantages:**

• They require threshold optimisation to meet the particular target for final segmentation.

• They generally cannot perform good segmentation when the light is strong or poor.

• They are only suitable for segmentation where the dominant plant colour is green.

Threshold based-methods require several adjustments with different lighting conditions. Therefore, once change occurs, the segmentation error may increase. Moreover, some threshold techniques might be suitable for one case, but not for others.

The learning-based approaches demonstrated better performance over colour index-based methods under a variety of illumination conditions because they rely on a training phase, but this results increased computation time which is not preferable in real time applications. Moreover, in order to perform reliable segmentation results, substantial training samples are required.

While good segmentation performance has been achieved with the methods considered, several challenges remain:

• Lighting conditions: cloudy, overcast, and sunny conditions impact segmentation quality. For example, when the light is strong as on a sunny day, the surface of some
leaf types such as corn leaf, acts as a mirror (specular reflection); as a result, it may be segmented into the wrong category.

- Shadow, including shadow caused by a plant itself or by other objects (cast shadow), may be extracted as foreground (plant vegetation); as a result, the mis-segmentation rate is increased.

- Complex background (scene of the image), including straws, stones, soil colour, water pipes, and other residues, can affect the segmentation quality particularly if a background element has a green colour such as green pipe; as a result, it might be mis-segmented as plant.

These factors still remain as serious challenges for the available segmentation approaches. Therefore, further research is required to fully optimize the technology of computer vision for the complex conditions that may occur in commercial agriculture fields.

In addition to the development of specific algorithms for processing colour images, a number of studies have also considered other factors associated with acquiring images, and the issues that need to be considered in order to obtain good performance. In [146] Woebbecke et al. considered the detection of plants using a range of sensors (thermal and optical) and determined that the location and coverage of target plant components within the field of view of the sensor can significantly influence performance and must be taken into consideration. This work was extended in [147] who specifically considered the detection of bind weed and determined the maximum field of view for a given target size on bare soil. Both of these studies uses the Normalised Difference Vegetative Index (NDVI), which is the ratio of the difference between near infra-red reflectance and red components, and their sum. Criner et al. also emphasised the value of being able to configure the detection algorithm based on specific conditions, e.g. knowledge of field conditions or soil moisture can be used to adapt detection thresholds to maximise performance. Later studies reflected increasing use of digital visible spectrum cameras and relied on the indices based on R, G and B channels
and their derivatives discussed in this work. In [120] Meyer et al. described a system based on
the use of a number of clustering algorithms using colour indices. Images were acquired using
a camera that automatically set parameters such as focus, exposure time and white balance.
More recent studies have also examined the specifics of the imaging sensor. For example, in
[148] Dworak et al. used a low-cost single-chip camera again using NDVI, and compared it
to a much more expensive specialised imaging device. Good performance was achieved by
appropriately reconfiguring camera filters, coupled with algorithmic modifications. While
the topic of the optimal camera parameters (such as field of view) is beyond the scope of this
survey, previous studies suggest that the ability to adapt detection algorithm parameters such
as thresholds can provide an advantage in ensuring optimal performance.

By way of conclusion, Table 3.4 summarises the key conclusions from the review, and in
particular suggests specific algorithms that may perform well in particular conditions, based
on analysis of their performance based on studies from the literature.

Table 3.4. Suggested segmentation algorithms for use in different conditions.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Complexity</th>
<th>Performance</th>
<th>Accurate</th>
<th>Suitable application fields</th>
<th>Suggested algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour index-based</td>
<td>Simple</td>
<td>Effective</td>
<td>Low</td>
<td>Effective</td>
<td>For cloudy day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>accuracy</td>
<td>Poor segmentation result</td>
<td>CIVE, COM1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>if</td>
<td></td>
<td>For overcast day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the light is strong or poor</td>
<td></td>
<td>ExGR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For sunny day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>COM2 or ExG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>because both are</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>good for addressing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>shadow</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For cloudy day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Otsu</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For overcast and</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sunny days:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dynamic threshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>or Homogeneity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>threshold</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For cloudy day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EASA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For overcast day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AP-HI</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>For sunny day:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DTSM because it is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>good in addressing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>problems such as</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>shadow and specularly</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>reflected regions.</td>
</tr>
</tbody>
</table>

Threshold-based approach  | Fairly simple | Somewhat effective | Fairly good accuracy | Effective | Threshold adjustments are required | For cloudy day: Otsu |
|                          |              |                    |                     |           |                                   | For overcast and sunny days: Dynamic threshold or Homogeneity threshold |
|                          |              |                    |                     |           |                                   | For cloudy day: EASA |
| Learning-based approach   | Complex     | Expansive          | High accuracy        | Effective | Several training steps are required | For overcast day: AP-HI |
|                          |              |                     |                      |           |                                   | For sunny day: DTSM because it is good in addressing problems such as shadow and specularly reflected regions. |
3.6 Conclusions

Based on their prevalence in literature, this chapter presented a detailed survey of colour index based-methods for segmentation, along with a brief survey of other methods. While performing well in their own right, colour index-based methods are also widely used as a reference to evaluate the performance of other proposed methods. A detailed discussion of the performance of colour index based-methods, based on a number of recent studies, was presented. Threshold-based approaches were briefly discussed and their advantages and disadvantages were presented. In addition, the advantages and disadvantages of learning-based segmentation methods were briefly considered. The challenges and limitations that continue to hold for segmentation approaches were also highlighted. Finally, suggested segmentation algorithms for use in different conditions were give. The next chapter presents the development of a novel detection method for use in a variety of illumination conditions.
Chapter 4

Development of a Novel Crop Detection Algorithm for Varying Illumination Conditions

4.1 Introduction

Previous chapters have discussed the problem of automatic crop and weed detection in smart agriculture applications such as automatic weeding, and have highlighted some of the challenges that arise in this application, particularly when using such systems in field conditions. In this chapter, a novel algorithm based on colour features and morphological erosion and dilation is proposed and applied to the task of cauliflower detection. This algorithm segments cauliflower crop regions in the image from weeds and soil under conditions of natural illumination (cloudy, partially cloudy, and sunny). The algorithm uses the HSV colour space for discriminating crop, weeds and soil and defines a region of interest (ROI) by filtering each of the HSV channels between certain values (minimum and maximum threshold values). The region is then further refined by using a morphological erosion and dilation process. The method of moments is applied to determine the position and mass distribution of objects.

4.2 Background and Related Work

In the proposed algorithm, the HSV colour space is applied to discriminate cauliflower plants from among weeds, soil, and other residues under actual field conditions. The HSV channels are described as follows:

- **Hue**: Describes a pure colour (e.g. pure yellow, orange or red).
- **Saturation**: Describes how much a pure colour is diluted with white light.
- **Value**: Refers to the brightness of the colour.

This work was motivated by the fact that the HSV colour space is robust to illumination variation [149, 150]. In addition, the HSV space is more aligned with human colour perception [151, 152]. Because of these attributes, HSV has been widely used in several computer vision applications such as skin tone analysis [153], face detection [154], hand gesture detection [155], road signs detection [156], shadow removal [157], daisy detection in horticulture [158], and object tracking [159]. Furthermore, most available techniques for detecting plants and classifying them into crops or weeds have dealt with single images captured by a stationary camera, and rarely consider plant detection frame by frame based on video acquired in field conditions. In contrast, in this thesis a camera moving along rows of cauliflower seedlings was employed, and therefore a high degree of the natural instability that will normally exist in practice is captured, e.g. the natural vibration, speed variation etc. that occurs with a moving camera system. The accuracy and false positive detection of crops in frame sequences were considered in this context.
The HSV decision tree based method for greenness identification was proposed by Yang et al. [160]. The method was proposed to identify plants from maize seedling images acquired outdoors. The method was tested against different environmental conditions (sunny, clean soil; sunny, straw ash; sunny, wheat straw; cloudy, clean soil; cloudy, straw ash; cloudy, corn straw). The method involves several steps: firstly, the crop image is converted from the RGB colour space to the HSV colour space. Then, the background pixels were removed by comparing their hue values with the plant pixels (plant pixels can be obtained empirically). Next, the pixels of wheat straws whose hue values overlap with green leaves are eliminated based on their hues, saturations, and values. Thresholding is used for binarisation. Finally, with help of morphological operations, small objects with area less than 100 pixels are removed to isolate the green plants.

Yang’s method was tested and compared with common colour index-based methods such as excess green index (ExG) [60], the excess green minus excess red index (ExGR) [61], the vegetative index (VEG) [62], the colour index of vegetation extraction (CIVE) [45], a combined index (COM1) [161], and with Otsu’s Method [123]. The method demonstrated good results over other methods used for comparison. The method has also shown its ability to identify plants in different environmental conditions. However, the author concluded that the algorithm will give poor results if it is applied in poor or strong light conditions. The author also reported that the proposed algorithm was not tested for all the weather conditions. The work presented in this chapter attempts to develop a method for identification of green leaves based on the HSV colour space in three weather conditions (Sunny, Partially cloudy, and Cloudy), and with complex backgrounds (Stones, Shadow, Plastic film, etc.). The proposed method further distinguishes crops from weeds.
4.3 Proposed Detection Algorithm

This section outlines the proposed method in detail, including the different steps in the algorithm, and the rationale for choice of parameters. As mentioned in the introduction, the HSV colour space has performed optimally for several computer vision applications in varying image conditions and consequently has been applied here.

The CAMSHIFT algorithm or Continuously Adaptive Mean Shift algorithm was proposed in [162]. The algorithm is based on the Mean Shift algorithm [163], which operates on probability distributions. To track colored objects in video frame sequences, color image data are represented as a probability distribution (histograms). The color distributions derived from video may change over time, so the algorithm has to adapt dynamically to the distribution it is tracking. Thus, CAMSHIFT algorithm was proposed to overcome this issue.

The CAMSHIFT algorithm operates as follows:

1. Set the region of interest (ROI) of the probability distribution to the entire image.

2. Select an initial location for the Mean Shift search window. The selected location is the target distribution to be tracked.

3. Calculate a probability distribution of the region centred on the Mean Shift search window.

4. Iterate the Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and centroid location.

5. For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment.

6. Return to Step 3 and continue iterating until convergence is achieved, i.e. parameter updates fall below a specified threshold.
The proposed algorithm is similar to the well-known colour based segmentation CAMSHIFT algorithm in the following ways:

- Both algorithms use the HSV colour space.
- The ROI mainly contains one colour.
- The ROI does not change colour.
- There are no other objects in the scene with the same colour as the target object.
- The colour of the background differs from the ROI colour.
- Both algorithms use a statistical moment method to track the target object.
- Both algorithms adapt their search window size in the video sequence based on the zeroth order of the moment. However, the proposed algorithm differs from the CAMSHIFT algorithm as follows:

  - The proposed algorithm uses all three channels (hue, saturation, and value) while CAMSHIFT uses only hue. In the proposed algorithm, hue is used to identify the cauliflower region, while saturation and value are used to account for variations in illumination and for removing the background.

  - The proposed algorithm defines the target object based on filtering the colour ranges of all three HSV channels, while CAMSHIFT relies on a colour probability distribution image derived from colour histograms of the hue channel only.

  - Unlike the CAMSHIFT algorithm, the proposed algorithm does not require an initial search window location to be chosen. Thus, the proposed algorithm might be more effective for use in real conditions.

The proposed method is illustrated in the diagram in Fig. 4.1. The method consists of a filter stage, and a detection stage. The proposed algorithm was implemented in OpenCV.
Figure 4.1. General scheme for crop detection algorithm
4.4 Algorithm Steps

4.4.1 Algorithm Outline

This section describes the main steps in the detection algorithm. Figure 4.2 illustrates the output of different stages of pre-processing, where Figure 4.2(a) shows an example of an image frame from a test video. The main goal of segmentation is to segment plant pixels from background pixels. Firstly, the original frames of video are reduced to a quarter of their original size for computational efficiency (for the experiments described later, the original frame sequences were captured with resolution of 2704 × 1520 pixels). Secondly, a blur filter was applied to enhance and reduce the noise in the image. The Gaussian filter is efficient in noise removal [164] and this was applied with a 3x3 sliding window. The two dimensional Gaussian function was used and can be written as follows.

\[
G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.1}
\]

where \(\sigma\) is the standard deviation of Gaussian distribution (which controls the amount of smoothing or blurring).

The output of blur filter is shown in Figure 4.2(b). Thirdly, the blurred image (RGB image) was converted to the HSV colour space (Figure 4.2(c)). Then, the HSV image is filtered based on the ranges identified in parameter selection, and thresholding is carried out. The region of interest (ROI) is defined by first limiting each of the HSV channels to pixels that lie between values Cmax and Cmin for each channel (e.g. \(H_{\text{min}}\) and \(H_{\text{max}}\); \(S_{\text{min}}\) and \(S_{\text{max}}\); \(V_{\text{min}}\) and \(V_{\text{max}}\) for Hue, Saturation, and Value, respectively), pre-determined during a parameter selection phase described below. A threshold value is then chosen to create a binary image. The choice of threshold value is described below. The output image from this process is a thresholded black and white image (Figure 4.2(d)). Finally, after filtering erosion and dilation are applied to the output of filtered image. Morphological operations...
have played an important role in crop and weed recognition and detection, e.g. [157, 165]. Here, erosion with a 3x3 structure element was applied to remove the remaining isolated white pixels (corresponding to weeds) from the binary image. Dilation with a 7x7 structure element was employed to accentuate the shape of the remaining objects (crops). The output of morphological erosion and dilation can be seen in Figure 4.2(e).

After pre-processing, several steps are performed to detect and track the target object (cauliflower plant) in video sequences. Firstly, contours are extracted from the binary image. Here, a contour is a curve joining all the continuous pixels (e.g. along a boundary of an object) having the same colour or intensity. After filtering, the connected components (connected white pixels) are found in the binary image to denote shape features such as enclosed area and perimeter. Secondly, calculate the moments of the objects identified from contours. Moments are frequently used as features for image processing such as: remote sensing, shape recognition and classification, image coding and reconstruction [166]. In this work, moments are applied to characterise the white pixel collections representing the target object in the image. The zeroth moment is calculated as follows:

\[ M_{00} = \sum_x \sum_y I(x, y) \]  \hspace{1cm} (4.2)

where \( M_{00} \) is the zeroth moment and \( I(x, y) \) is the pixel value of each contour in the binary image (all of the white pixels in an object).

The first moments for \( x \) and \( y \) are given in the following formulas:

\[ M_{10} = \sum_x \sum_y xI(x, y) \]  \hspace{1cm} (4.3)

\[ M_{01} = \sum_x \sum_y yI(x, y) \]  \hspace{1cm} (4.4)
Figure 4.2. Pre-processing results. (a) Original image. (b) Blurred image. (c) HSV image. (d) Filtered image using HSV ranges. (e) Image after morphological erosion and dilation.
The centre of mass of each contour is then calculated as:

\[ xc = \frac{M_{10}}{M_{00}}, \quad yc = \frac{M_{01}}{M_{00}} \]  

(4.5)

where \( xc \) is the centre of mass of the object in the \( x \) direction and \( yc \) is the centre of mass of the object in the \( y \) direction.

The location of a cauliflower's centre of mass is an important feature because it represents the location with most leaves. Thus, for example, these coordinates can be used to guide target spraying. After filtering and object detection, false positives may still exist (such as a weed that might have the same hue values as cauliflower). Therefore, contour length along with contour area is used to refine the detected objects. The contour perimeter can be defined as set of points that make up the object boundary. Contour pixels are identified using the Canny edge detection algorithm [167]. Then a perimeter value is calculated as:

\[ P = \sum_x \sum_y I(x, y) \in \text{Contour} \]  

(4.6)

where \( P \) is the perimeter of the contour and \( I(x, y) \) is the pixel value of contour (the pixels that are located at the boundary of the object).

To obtain good segmentation results, a decision rule was established by setting minimum threshold values for the area and perimeter length of a detected object. Each detected contour was then compared to these area and length thresholds. The area and perimeter thresholds were derived from a range of images employed in the training phase. For confirmed detection,
the target object area must fulfil the following detection rule conditions:

\[
\text{Cauliflower Detected if} \quad ((\text{Area of the contour}) \geq (\text{The minimum object area})) \quad \text{AND} \quad ((\text{perimeter of the contour}) \geq (\text{the minimum object perimeter}))
\]  

(4.7)

Once the target object is located, a boundary box is drawn around it for identification purposes. As mentioned earlier, the proposed algorithm has similarities with the CAMSHIFT algorithm in adapting dynamically to changing pixel distributions as it relies on the zeroeth moment calculated for each image. This feature is useful for processing video sequences, since the plant size may change from frame to frame due to perspective changes, motion of leaves due to the wind, etc. Therefore, a dynamic window is useful to accurately detect crops of changing sizes.

Based on the zeroeth and first moments, the centre of mass calculated earlier, and second moment, the aspect ratio can be calculated as:

\[
M_{20} = \sum_x \sum_y x^2 I(x, y)
\]  

(4.8)

\[
M_{02} = \sum_x \sum_y y^2 I(x, y)
\]  

(4.9)

\[
\text{ratio} = \frac{M_{20}}{x_c} / \frac{M_{02}}{y_c}
\]  

(4.10)

Then, the search window dimensions can be obtained as:

\[
\text{width} = 2M_{00} \times \text{ratio}; \quad \text{height} = 2M_{00} / \text{ratio}
\]  

(4.11)
Before executing the proposed method with test data, several algorithm parameters must be tuned, based on the use of training images. Parameters are determined based on different video sets that were captured in different weather conditions (cloudy, partially cloudy, and sunny). The parameters selected with the training data are then used in the system when tested with a completely different set of video clips for the testing phase. The possible range for $H$ lies between $0^\theta$ and $180^\theta$ while the possible range for $S$ and $V$ lies between 0 and 255.

In order to determine suitable thresholds from the training images, histograms of the $H$, $S$ and $V$ channels were calculated to determine how distinguishable plants and weeds are. Figure 4.3 shows a sample image captured in cloudy conditions. The corresponding histograms for its $H$, $S$ and $V$ channels are shown in Figure 4.3; the histogram boundaries, which are indicated with red arrows, were determined empirically from the test images (other images captured in the same conditions exhibited similar characteristics).

From the histogram, it can be seen that the Hue channel gives greater contrast between cauliflower and weeds than the other channels (Saturation and Value). It can also distinguish the cauliflower region from the soil and other residues (e.g. small stones, pieces of straw etc.). The overlap region between cauliflower and weeds is not wide, which facilitates threshold selection for the algorithm. In this study, two threshold values were used. The minimum threshold separates cauliflower from weeds, and the maximum threshold separate the cauliflower from other background residues. From the $S$ channel histogram, soil and plant regions (cauliflower and weeds) can be readily distinguished; however, it is more difficult to separate the cauliflower from weeds on the basis of Saturation alone. The $V$ channel histogram shows that by itself it cannot be easily used to distinguish between the different categories, and it is mainly used here to impose a further constraint on the possible search space. The selected ranges of HSV cauliflower values under the corresponding weather conditions are presented in Table 4.1
4.4 Algorithm Steps

Figure 4.3. Example image (RGB) and its HSV channels; Hue image (H), Saturation image (S), and Value image (V)

Figure 4.4. The histogram of Hue, Saturation, and Luminosity Value of cropped image. The red arrows indicate the approximate boundaries of different types of material in the image, including: cauliflower (item of primary interest), weeds, soil, and other residues
Table 4.1. The ranges of HSV channels for cauliflower pixels under the corresponding weather conditions

<table>
<thead>
<tr>
<th>Conditions</th>
<th>HSV values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H-Min</td>
</tr>
<tr>
<td>Cloudy</td>
<td>80</td>
</tr>
<tr>
<td>Partially cloudy</td>
<td>70</td>
</tr>
<tr>
<td>Sunny</td>
<td>65</td>
</tr>
</tbody>
</table>

Moreover, to make our proposed algorithm robust to the three weather conditions considered (cloudy, partially cloudy, and sunny), a dynamic and automatic threshold calculation method to handle all these conditions is required. The S channel correlates with the light level, and therefore is a good choice to determine a light-dependent threshold automatically. With this assumption, the threshold value used to segment the cauliflower pixels from the background (weeds, soil, and other residues) is calculated as follows:

1. Determine the range of values occupied by each of the HSV channels for cauliflower-only pixels, for each condition as described above (performed empirically using the training data). A single range for S and V was found to be appropriate for all lighting conditions. In contrast, the appropriate range for H changes with lighting variations. The following table illustrates the HSV ranges of the cauliflower for all three conditions.

2. Determine the average S channel value for each frame, for each weather condition. This can be obtained by dividing the sum of the pixel values of the S channel for each frame, by the total number of pixels in the frame.

3. Select the minimum and maximum average value of S for each weather condition, over the entire video. These are denoted as $S_{\text{min-avg}}$ and $S_{\text{max-avg}}$.

Table indicates the selected average values of the S channel for the three conditions.
Table 4.2. The selected average range of S channel for each weather condition

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Range of average value of S channel for captured video under different conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_{\min-\text{avg}}$</td>
</tr>
<tr>
<td>Cloudy</td>
<td>12</td>
</tr>
<tr>
<td>Partially cloudy</td>
<td>16</td>
</tr>
<tr>
<td>Sunny</td>
<td>20</td>
</tr>
</tbody>
</table>

HSV threshold values can be selected based on the average value of $S$ for each frame. In particular, if the average value of $S$ falls within the pre-defined ranges for each weather condition as listed in Table 4.2, the corresponding HSV range is chosen from Table 4.2.

4.5 Testing framework and performance metrics

4.5.1 Test Data

To determine the accuracy and robustness of the proposed detection method, a suitable performance metric is required. A common approach in computer vision object detection applications is to measure the degree of intersection between detected objects and ground truth annotations. There are several evaluation methods used in this respect such as e.g. used in KITTI, proposed in [168] and the PASCAL Visual Object Classes (VOS), proposed in [169]. In this work, the PASCAL Visual Object Classes method was adopted to deal with the boundary box detections.

To test the proposed algorithm, a dataset was captured by a digital camera (GoPro Hero 4 Silver with maximum video resolution of 3840×2160 pixels, Effective Photo Resolution of 12.0 MP, and memory card max supported size of 64 GB) was used to acquire cauliflower images (video sequences) under a variety of illuminations: cloudy, partially cloudy, and sunny for different stages of growth (from June until the end of September, 2015). Additionally, various circumstances such as partial occlusion between crop and weeds, partial crop disappearance from the scene, leaves which were partially eaten by insects, light changes,
motion caused by the wind, different type of shadows, and various background (soil, nylon, stones, and other residues) were included. The video was captured in the west of Ireland. A top-view camera position was adopted to capture the image sequences with a frame resolution of 2704×1520 pixels. A standard desktop computer with an Intel i7-4790 CPU running 64-bit Windows 7 and 32 GB of RAM with OpenCV 2.4.10 installed was used for algorithm development and execution.

For algorithm evaluation, a dataset containing 11164 annotated cauliflower bounding boxes was manually constructed, which included the three weather conditions. A total of 1650 frames (550 frames for each condition) was used, with an input frame rate of 30fps. Several videos containing various weather conditions, cauliflower growth stages, and weed-free and weed-infested conditions were used. Separate data sets were employed for training and testing. For the training phase, the total frame numbers used was 22975 frames (6570 frames for sunny condition, 7763 frames for partially condition, and 8642 frames for cloudy condition) with an input frame rate of 30fps.

### 4.5.2 Performance evaluation

Accuracy is evaluated by calculating the overlap area $a_\theta$ between the detected bounding box $(B_{det})$ and the ground truth (annotated) bounding box $(B_{gt})$ for each test frame. Thus, the overlap ratio is defined as follows:

$$a_\theta = \frac{\text{area}(B_{det} \cap B_{gt})}{\text{area}(B_{det} \cup B_{gt})}$$

where $(B_{det} \cap B_{gt})$ is the intersection of the detected and ground truth bounding boxes and $(B_{det} \cup B_{gt})$ is their union. In [169] Everingham et al. proposed that for determining correct object detection, the area of overlap $a_\theta$ between the detected (predicted) bounding box $B_{det}$
and ground truth bounding box $B_{gt}$ must exceed 0.5 (50%). Thus, this threshold has been adopted to determine when correct detection of a crop takes place.

To quantify overall classification performance of the proposed algorithm, two common metrics are used in this study: Precision ($Pr$) and Recall ($Re$) [170]. Precision gives information about the validity of segmentation results and Recall gives information about the correctly identified cauliflower pixels in an image. Higher value of precision and recall indicate better performance by the segmentation method. Under segmentation is indicated by low recall while over segmentation is indicated by low precision. These metrics are defined as follows:

$$Pre = \frac{T_p}{T_p + F_p} \quad (4.13)$$

$$Rec = \frac{T_p}{T_p + F_n} \quad (4.14)$$

where $T_p$ (true positive) represents the number of correctly detected cauliflowers, $F_p$ (false positive) represents the number of false detections, and $F_n$ (false negative) represents the number of failed detections.

### 4.6 Results and discussion

#### 4.6.1 Performance Evaluation

The test-video was captured in a cauliflower field in the west of Ireland under various weather conditions; cloudy, partially cloudy, and sunny days. Different cauliflower growth stages were also accounted for. Additionally, different weed species such as Fat hen, Shepherd’s purse, and Duckweed are present at different growing stages, and weeds partially occlude the cauliflower leaves.

Optimum cauliflower detection performance was demonstrated with a minimum contour area threshold of 110 and a minimum perimeter of 30 for all imaging conditions. Typical
experimental results from the proposed algorithm are illustrated in Fig. 4.5 below. Fig. 4.5(a) illustrates detection results for overcast conditions (where the sun is totally obscured). This figure includes cauliflower crops, weeds, soil, and stones. The indicated red rectangles represent detections produced by the algorithm, and the green dots represent the centre of mass of each cauliflower. This figure also demonstrates how cauliflowers that are only partially visible in the frame (at the bottom) have been detected. Fig. 4.5(b) is an example result for partially cloudy conditions (where the sun is partially obscured). Fig. 4.5(c) is an example in sunny conditions (clear sky, no clouds). Along with cauliflower crops, weeds, soil and stones, this figure illustrates shadows (shadows cast by plants, and by other objects, in this case a person standing nearby).

The precision and recall are calculated for the three conditions. The results of the proposed algorithm are presented in Table 4.3.

Table 4.3 illustrates that the best Recall performance was demonstrated in partially cloudy conditions. The proposed method also demonstrated very good performance under cloudy condition (quite close to that in partially cloudy). The lowest performance was demonstrated under sunny condition. Most false detections occur in sunny conditions, where some weeds have similar colour, contour perimeter, and area to cauliflowers (as can be shown in Fig. 4.6). However, even in this case, Recall is still above 97.00%. Where false detections occur in cloudy and partially cloudy conditions, these are mainly due to weed leaves that have similar colour to cauliflowers, especially in the early stage of cauliflower growth. As a result, weeds

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Average metric</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Cloudy</td>
<td>99.69%</td>
<td>99.47%</td>
</tr>
<tr>
<td>Partially cloudy</td>
<td>99.40%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Sunny</td>
<td>98.04%</td>
<td>97.28%</td>
</tr>
</tbody>
</table>
Figure 4.5. An example of results from the proposed method for a set of images were captured in various light conditions. (a) Cloudy/overcast day; (b) Partially cloudy day; (c) Sunny day. Cauliflower plant regions are highlighted with red rectangles. Green dots indicate the location of the centre of mass of each crop.
Development of a Novel Crop Detection Algorithm for Varying Illumination Conditions

may appear as having the same colour, contour perimeter, or area as cauliflowers, resulting in mis-classification.

Figure 4.6. An example of false detection results from the detection algorithm for a test image, frame numbers 226 captured in sunny condition. A single weed was wrongly detected and highlighted with a red rectangle as a cauliflower. Each cauliflower plant in the row is independently detected by the algorithm and highlighted with red rectangle; ground truth is indicated by blue rectangles; yellow rectangles indicate the overlap between detections and annotations. Green dots indicate the location of the centroid of each plant

Despite dealing with a variety of field conditions, the performance of the proposed algorithm over all conditions (given in Table 4.4) is very good with an average precision of 99.04% and recall of 98.91%. Such results will greatly increase accuracy of herbicide and pesticide administration leading to a reduced environmental impact, lower consumable costs and ultimately a higher crop-quality.
Table 4.4. Overall performance of proposed algorithm.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed algorithm</td>
<td>99.04%</td>
<td>98.91</td>
</tr>
</tbody>
</table>

4.6.2 Comparison with HSV decision tree method

To further illustrate the performance of the proposed method, its segmentation performance was compared with the method for greenness identification based on HSV decision tree which proposed by Yang and described in [160]. Several scenarios were used for the testing and evaluation, including some particularly challenging ones: cloudy, partially cloudy, sunny, heavily infested weeds, shadow, and plastic film.

Figure 4.7 compares Yang’s method and the method proposed here through sample images of different complexity, along with segmentation output and detection output. Comparing the second and third columns in Figure 4.7 (segmentation results), in general, Yang’s method showed good green plant segmentation results in most test conditions. However, it failed to remove the all plastic film pixels as shown in Fig. 4.7: (b), (f), (j), and (n) while the proposed method has removed all the plastic film pixels as shown in Fig. 4.7: (c), (g), (k), and (o). The effect of the shadow was not fully addressed by Yang’s method as can be seen in Fig.4.7(f) while the proposed method has removed the all the shadow pixels as can be seen in Fig. 4.7(g).

4.7 Conclusions and future work

In this chapter, an algorithm for detection of cauliflowers from video streams has been proposed. Despite several challenges, such as the similarities in colour reflectance between cauliflower plant and some weed species, and variations in lighting conditions, the proposed algorithm demonstrated very good classification performance. Performance was assessed by comparing the obtained results with those of ground truth methods (manual annotation). A
Figure 4.7. Comparison between proposed method and [160]. First column shows test images. The second column includes the segmentation results from Yang’s method. The third column includes the segmentation results of the proposed method, and the fourth column includes the detection results from the proposed methods (Cauliflower plant regions are highlighted with red rectangles).
sensitivity of 98.91% and precision of 99.04% was achieved. The moment method allows the proposed algorithm to detect crop position, which can be useful for crop tracking in a video sequence and also for herbicide application. Additionally, the proposed algorithm can adapt its parameters to different sized crops, thereby enabling detection at different crop growth stages without explicit parameter reconfiguration. Comparison of the segmentation performance of the proposed method (classification between plant pixels and the background pixels) with greenness identification based on HSV decision tree method (Yang et al., 2015) was also performed, with the proposed method demonstrating better segmentation performance. One possible shortcoming associated with the proposed algorithm is its reliance on colour. If the cauliflower leaf colour changes because of disease, or because of very sunny conditions, the misclassification rate may increase. Besides identification of crop positions to be protected from weeding, the proposed method is also useful for other purposes that require plant location, and can be applied in several applications such as crop fertilization, plant species recognition, and growing phase determination. These applications are growing in importance with increasing interest in smart agriculture. Although the experimental results indicated that the proposed algorithm demonstrates a high degree of robustness against various conditions, there is still an opportunity for improvement especially reducing the false detection rate for sunny conditions. This is addressed in the next chapter through the addition of tracking to the detection algorithm.
Chapter 5

Enhanced Crop Detection System using Kalman Filter-based Tracking

5.1 Introduction

The previous chapter presented a new detection algorithm for cases of varying illumination. Despite the good performance of the algorithm, there are still situations where the algorithm fails to detect, especially in sunny conditions and also due to image capture conditions (vibration, variations in speed etc.). Hence, there is scope for further improvement.

In this study, Kalman filtering and the Hungarian algorithm are used to track multiple crop plants in video sequences. The algorithm consists of two steps. Firstly, Kalman filtering is used to predict the new position of an object (a cauliflower plant in this case) in video sequences. Secondly, a data association algorithm (the Hungarian algorithm) is used to assign each detected crop that appears in each image to the correct crop trajectory. To achieve this, the centroids of detected objects are determined and fed to the proposed tracking algorithm. The proposed method is described in detail in the following section, while Section 5.3 discusses the method used for performance evaluation. Discussion of results is presented in Section 5.4. The work presented in this chapter has been published in

### 5.2 Proposed method

While detection allows us to locate object positions in each frame, this in itself does not provide information about the movement of individual objects between frames so objects cannot be tracked over time. This can present problems if detection in each frame is challenging, and object tracking can provide additional robustness. Explicit tracking is necessary to follow object instances as they move through the field of view, according to an assumed movement model. The use of a tracking algorithm such as Kalman filtering allows refinement of the detection co-ordinates to produce a smoother track across multiple frames.

Fig. 5.1 shows a block diagram of the tracking algorithm proposed in this work. This includes: (i) detection stage (previously described in Chapter 4); (ii) extraction of feature centroid (the x and y coordinates of the centre of each detected object); these coordinates will be used to predict the current location of the track; (iii) tracking stage, which tracks detected objects and includes association of detected objects with plant trajectories.

The proposed tracking algorithm relies on an initial detection algorithm which has been described in Chapter 4. The method was tested against different environmental conditions (cloudy, partially cloudy, and sunny) for different stages of growth. Various circumstances, such as partial occlusion between crop and weeds, partial crop disappearance from the scene, leaves which were partially eaten by insects, light changes, motion caused by the wind, different type of shadows, and various backgrounds (soil, nylon, stones, and other residues) were also included.

As described in Chapter 4, cauliflower plants against weeds were detected with a sensitivity (Recall) of 98.91%. However, although the detection performance is high, errors
5.2 Proposed method

Figure 5.1. General scheme for the tracking algorithm

occurred more frequently in sunny conditions (with an error rate that was 2.72% higher than non-sunny conditions), hence the motivation to address this issue in this chapter.

5.2.1 Kalman filter

The Kalman filter has been widely used and successfully implemented in many different object tracking applications. The main advantage of the Kalman filter is that it can provide a reasonably accurate guess about the future position of any given object in a dynamic environment. To understand the Kalman filter process, assume that there is a state vector of data \( x_k \) of an unknown system, we wish to predict its behaviour (e.g., position, velocity, etc.) at discrete times \( k \), based on its previous behaviour which stored in the vector \( x_k \). Here, the description follows the granular approach taken in [84] to describe the algorithm, wherein the algorithm comprises four stages.

1- Process equation

The state of the system is determined from the following equation:

\[
x_k = A x_{k-1} + w_{k-1}
\]  

(5.1)
where $A$ represents the transition matrix, $x_k$ denotes as the state vector at time $k$, and $x_{k-1}$ denotes as the state vector at time $k - 1$. $w_{k-1}$ is the Gaussian process noise with zero mean and covariance $Q$.

2- Measurement equation

Outputs of the system can be calculated as follows:

$$z_k = Hx_k + v_k$$  \hspace{1cm} (5.2)

where $H$ is the measurement matrix and $z_k$ is the measurement observed at time $k$. $v_k$ is the Gaussian measurement noise with zero mean and covariance $R$. The state vector ($x$) which representing the positions and the velocity can be expressed by:

$$x = \begin{bmatrix} p_x & p_y & v_x & v_y \end{bmatrix}^T$$  \hspace{1cm} (5.3)

where $p_x$, $p_y$ are the center positions along the x-axis and y-axis, and $v_x$, $v_y$ are the velocities in the directions of the x-axis and y-axis.

The transition matrix is given as:

$$A = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$  \hspace{1cm} (5.4)

where $\Delta T$ is time increment, which is the frame interval in a video sequence. The measurement matrix is given as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$  \hspace{1cm} (5.5)
Since Kalman filtering relies on an estimation process, errors or noise can interfere with the tracking ability. This noise can be generated from several sources such as e.g. plants moving in the wind. The noise in the states can be expressed as a covariance matrix \((Q)\). In addition, whenever taking a measurement of the tracking object position, errors may occur, which will cause some variation of the true value of actual position of tracking object. The error in measurement can be expressed as the covariance matrix of the measurement noise\((R)\) and should be taken into account in order to get high tracking performance.

Choosing correct \(R\) and \(Q\) matrices is an important design factor for better performance of the Kalman filter. The optimal value of the \(Q\) matrix was determined empirically as part of system performance evaluation to be:

\[
Q = \begin{bmatrix}
\Delta T^4/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^4/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2 \\
\end{bmatrix}
\] (5.6)

The \(R\) matrix was obtained empirically as follows:

\[
R = \begin{bmatrix}
0.01 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0 \\
\end{bmatrix}
\] (5.7)

The method used to determine these optimal \(Q\) and \(R\) values is described in Section 5.2.1.2.

3- Update equations

To predict the state of \(x_k\), the information provided by \(z_k\) is used. Thus, a priori estimate of state \(\hat{x}_k^-\) and covariance error \(P_k^-\) can be computed for the next time step \(k\) as follows:

\[
\hat{x}_k^- = A\hat{x}_{k-1} + W_k
\] (5.8)
\[ P_k^- = A P_{k-1} A^T + Q \]  

where \( Q \) is the process noise covariance.

4- Measurement updates equations

This is also called a correction process. It involves three processes:

• Calculation of Kalman Gain

\[ K = P_k^- H^T (H P_k^- H^T + R)^{-1} \]  

Kalman gain depends on the accuracy of a measurement. If an accuracy of the measurement is high, the Kalman gain has high value. Otherwise, the Kalman gain has relatively low value.

• Update estimate with measurements \( z_k \)

\[ \hat{x}_k = \hat{x}_k + K (z_k - H \hat{x}_k^-) \]  

• Update covariance error

\[ P_k = (1 - KH) P_k^- \]  

where \( P \) is the prediction error covariance, \( R \) is the measurement error covariance and \( K \) is the Kalman gain.

5.2.1.1 Kalman filter for multi-object tracking

In this study, the detected cauliflower’s centroid was used to track it over successive frames. To track each object, a separate Kalman filter tracker is required for every cauliflower. Thus, the proposed system consists of a multi-object tracker to track every existing moving object (cauliflower) in the video sequence. As the number of objects increases, the estimation
process gets more complex. Therefore, an association algorithm is required in order to correctly associate each detected object with an object track. The association algorithm will be described in detail in subsection 5.2.2.

5.2.1.2 Optimisation of Kalman Filter

The choice of the values of $Q$ and $R$ matrices in the Kalman filter impacts the system performance. A series of experiments was conducted wherein $Q$ and $R$ were varied; in each experiment, the value of $Q$ was held constant and $R$ varied such that the diagonal elements took values of 1.0, 0.1 or 0.01. The starting point for evaluation was the value for $Q$ used in [171]. In total, 12 different combinations of $Q$ and $R$ were investigated and the recall was estimated for each one. Generally-speaking, smaller values of the diagonal elements of $R$, and smaller values for the elements of $Q$, results in better performance, however, performance eventually saturated.

The full set of $Q/R$ combinations and results is given in following experiments (details of the performance evaluation method are given in Section 5.3):

a) Experiment 1

$$Q = \begin{bmatrix} 0 & \Delta T^2/2 & 0 \\ \Delta T^2/3 & 0 & \Delta T^2/2 \\ 0 & \Delta T & 0 \end{bmatrix} \times 0.1, \quad R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \text{Recall} = 98.3144\%$$

$$Q = \begin{bmatrix} 0 & \Delta T^2/2 & 0 \\ \Delta T^3/3 & 0 & \Delta T^2/2 \\ 0 & \Delta T & 0 \end{bmatrix} \times 0.1, \quad R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \end{bmatrix}, \quad \text{Recall} = 99.0106\%$$
92

Enhanced Crop Detection System using Kalman Filter-based Tracking

\[
Q = \begin{bmatrix}
\Delta T^3 / 3 & 0 & \Delta T^2 / 2 & 0 \\
0 & \Delta T^3 / 3 & 0 & \Delta T^2 / 2 \\
\Delta T^2 / 2 & 0 & \Delta T & 0 \\
0 & \Delta T^2 / 2 & 0 & \Delta T
\end{bmatrix},
R = \begin{bmatrix}
0.01 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0
\end{bmatrix}, \quad Recall = 99.3038\%
\]

(5.15)

b) Experiment 2

\[
Q = \begin{bmatrix}
\Delta T / 4 & \Delta T / 2 & 0 \\
0 & \Delta T / 4 & \Delta T / 2 \\
\Delta T / 2 & 0 & \Delta T \\
0 & \Delta T / 2 & 0
\end{bmatrix},
R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}, \quad Recall = 99.0473\% \quad (5.16)
\]

\[
Q = \begin{bmatrix}
\Delta T / 4 & \Delta T / 2 & 0 \\
0 & \Delta T / 4 & \Delta T / 2 \\
\Delta T / 2 & 0 & \Delta T \\
0 & \Delta T / 2 & 0
\end{bmatrix},
R = \begin{bmatrix}
0.1 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0
\end{bmatrix}, \quad Recall = 99.3038\%
\]

(5.17)

\[
Q = \begin{bmatrix}
\Delta T / 4 & \Delta T / 2 & 0 \\
0 & \Delta T / 4 & \Delta T / 2 \\
\Delta T / 2 & 0 & \Delta T \\
0 & \Delta T / 2 & 0
\end{bmatrix},
R = \begin{bmatrix}
0.01 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0
\end{bmatrix}, \quad Recall = 99.3404\%
\]

(5.18)
5.2 Proposed method

c) Experiment 3

\[
Q = \begin{bmatrix}
\Delta T^4/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^4/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2
\end{bmatrix}, \quad R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}, \quad \text{Recall} = 98.8641\%
\]

(5.19)

\[
Q = \begin{bmatrix}
\Delta T^4/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^4/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2
\end{bmatrix}, \quad R = \begin{bmatrix}
0.1 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0
\end{bmatrix}, \quad \text{Recall} = 99.3038\%
\]

(5.20)

\[
Q = \begin{bmatrix}
\Delta T^4/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^4/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2
\end{bmatrix}, \quad R = \begin{bmatrix}
0.01 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0
\end{bmatrix}, \quad \text{Recall} = 99.3404\%
\]

(5.21)

d) Experiment 4

\[
Q = \begin{bmatrix}
\Delta T^2/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^2/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2
\end{bmatrix}, \quad R = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}, \quad \text{Recall} = 98.8641\%
\]

(5.22)

\[
Q = \begin{bmatrix}
\Delta T^2/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^2/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2
\end{bmatrix}, \quad R = \begin{bmatrix}
0.1 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0
\end{bmatrix}, \quad \text{Recall} = 99.3038\%
\]

(5.23)
\[ Q = \begin{bmatrix} 
\Delta T^2/4 & 0 & \Delta T^3/2 & 0 \\
0 & \Delta T^2/4 & 0 & \Delta T^3/2 \\
\Delta T^3/2 & 0 & \Delta T^2 & 0 \\
0 & \Delta T^3/2 & 0 & \Delta T^2 
\end{bmatrix}, \quad R = \begin{bmatrix} 
0.01 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0 
\end{bmatrix}, \quad \text{Recall} = 99.3038\% \]

Based on these experiments, the combination of \( Q \) and \( R \) that gave the highest Recall were chosen. Since the lowest values of \( \Delta T \) and \( R \) from the experiments above gave the best performance, they were chosen for the system as given by Equations (5.6) and (5.7).

### 5.2.2 Hungarian data association algorithm

In this study, the Hungarian algorithm [95] is used to assign (match) the detections in each frame to tracked objects (the predicted positions from the Kalman filters), and to determine which identified objects have gone missing and which ones should be assigned to a new track. The algorithm is based on a distance (cost) matrix that contains the Euclidean distances between each combination of tracks (predictions) in the rows of the matrix, and detections in the columns; distances are calculated based on the centroids of predicted and detected plants. A smaller distance implies a higher probability of correct associated of detections to predictions. The size of the distance matrix can vary over time as crop density varies and plants disappear and appear in the field of view. The algorithm consists of a number of steps, as described in Table 5.1.

### 5.2.3 Additional post-processing

To increase tracking and detection reliability, the following heuristics have also been added:

1. Remove pairs that are matched over a large distance.

2. If a track has no detection assigned to it, then increment the skipped frame counter.
### Table 5.1. Hungarian algorithm steps

<table>
<thead>
<tr>
<th>Steps</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Subtract the minimum number in each row from the entries of that row. This is called row reduction. Create a new matrix to contain the results.</td>
</tr>
<tr>
<td>2.</td>
<td>Subtract the minimum number in each column of the new matrix from every number in the column. This is called column reduction. Create another matrix to contain the results.</td>
</tr>
<tr>
<td>3.</td>
<td>Check for the possibility of an optimum assignment. This can be done by determining the minimum number of “covering lines” (rows or columns) needed to cover all zeros in the matrix. If the number of lines equals the number of rows, an optimum assignment is possible. In that case, go to Step 6. Otherwise, go to Step 4.</td>
</tr>
<tr>
<td>4.</td>
<td>If the number of lines is less than the number of rows, execute the following steps:</td>
</tr>
<tr>
<td>4.1</td>
<td>Of the matrix entries (“cells”) that are not covered by the lines identified in Step 3, subtract the minimum uncovered cell value from all the uncovered cell values in the matrix.</td>
</tr>
<tr>
<td>4.2</td>
<td>Add the minimum uncovered cell value to the all numbers at intersection points of the lines identified in Step 3.</td>
</tr>
<tr>
<td>4.3</td>
<td>All other entries remain the same.</td>
</tr>
<tr>
<td>5.</td>
<td>Repeat Steps 3 and 4 until an optimal matrix is obtained.</td>
</tr>
<tr>
<td>6.</td>
<td>Make the assignments. Begin with rows or columns with only one zero. Match items that have zeros, using only one match for each row and each column. Delete the rest of the cells.</td>
</tr>
</tbody>
</table>
3. If a track’s skipped frame counter exceeds a threshold of 10 (the threshold of 10 was obtained empirically), remove the track.

4. If the detection has been assigned, then update the track using the detection coordinates.

5. If not, continue using Kalman Filter predictions.

5.3 Testing framework and performance metrics

5.3.1 Performance evaluation

To illustrate the efficiency and robustness of the proposed tracking method, an appropriate metric is required. There are several common approaches that are used in computer vision for object detection and tracking applications to measure the degree of alignment between detected/tracked objects and ground truth annotations. For example, the PASCAL Visual Object Classes (VOS) benchmark [169] and the KITTI database and benchmark [168] are two commonly used approaches. In this work, the same approach that was used in Chapter 4 is applied, i.e. the PASCAL Visual Object Classes method. Separate data sets were employed for training and testing for sunny condition. For the training phase, 6570 frames of video taken in sunny condition were used, with an input frame rate of 30 fps. For testing phase, a dataset consisting of 550 frames, including 2734 manually annotated cauliflower bounding boxes, was used. This is the same data set for sunny conditions that was used in Chapter 4, thus allowing a direct comparison of performance.

5.3.2 Bounding box and Tracking evaluation

Accuracy is evaluated by calculating the overlap area between the tracked bounding box ($B_{trk}$) and the ground truth (annotated) bounding box for each test frame. In this work, the same approach that was used in Chapter 4 is applied. In this work, as minimizing the
number of failed detections of cauliflowers across multiple frames is the goal, the tracking
and detection algorithm performance was reported by using the Recall metric [172]. Recall
gives information about the correctly identified cauliflowers in an image. The recall metric
(Rec) is defined as follows:
\[
Rec = \frac{T_p}{T_p + F_n}
\]  
where \( T_p \) (true positive) represents the number of correctly detected cauliflowers, and \( F_n \)
(false negative) represents the number of failed detections.

5.4 Results and discussion

5.4.1 Performance of tracking algorithm

Figure 5.2 shows an example of the experimental results from the detection algorithm. This
includes two frames from the same test video: frame number 508 as shown in Fig.5.2 (a),
consists of a row with six cauliflower plants identified by numbers from 16 to 21; frame
number 457 as shown in Fig.5.2 (b) contains a row with five cauliflower plants identified
by numbers from 14 to 18. The red rectangles in the figure represent detection using the
algorithm of Chapter 4, the blue rectangles indicate ground truth (annotations); the yellow
rectangles represent the overlap area between the detection and annotation bounding boxes;
this must exceed 0.5 (50%) for detection to be declared. Figure 5.2(c) and (d) illustrates the
results from processing the same frames using the proposed algorithm.

In Figure 5.2(a), (b), it can be seen that there are two types of errors (mis-detections): the
first type of error is where plants are not detected at all by the detection algorithm. Frame
number 508 in Fig. 5.2(a) has two undetected cauliflowers (number 16 and 21). The second
type of error is where a plant is detected, however, the overlap between the detection and
annotation bounding boxes does not exceed 50%, and therefore a detection is not declared.
As shown in Fig 5.2(b), frame number 457 has one undetected cauliflower (number 16 which
Figure 5.2. An example of results from the detection algorithm Chapter 4 for a pair of test images, frame numbers 508 (a) and 457 (b) captured in sunny condition. Each cauliflower plant in the row is independently detected by the algorithm and highlighted with a red rectangle; ground truth is indicated by blue rectangles; yellow rectangles indicate the overlap between detections and annotations. Red dots indicate the location of the centroid of each plant. The results from the proposed algorithm of test images (frame number 508 (c) and 457 (d)). Each cauliflower plant in the row is independently tracked by the proposed algorithm and highlighted with red rectangle; ground truth is indicated by blue rectangles; yellow rectangles indicate the overlap between detections and annotations. Red dots indicate the location of the centroid of each crop.
is located in the middle of the image) where the overlap criteria was not satisfied, because the detection bounding box is much smaller than the plant itself. The remaining cauliflowers in both frames were detected correctly.

However, it can be seen from Fig 5.5.2(c) and (d), in both frame numbers (508 and 457), applying the proposed tracking algorithm results in correct detection of all cauliflowers.

The proposed algorithm was evaluated over the whole sunny condition video sequences used in Chapter 4, consisting of 550 frames of video with 2734 annotated cauliflowers, and the results are summarized in Figure 5.3. Here, a graph was plotted to show the percentage of cauliflowers that has been detected per frame, as a function of frame number, in both algorithms (detection only, and detection plus tracking).

Examining Fig. 5.3 (a), from the results using only the detection algorithm, it can be seen that there are missed cauliflowers in a number of frames. For example, about 20% of cauliflowers were missed among frame numbers 248-271, during the interval from frames 450-460, and frame number 503. In addition, there are about 66.6% missed in frame number 508, 60% were missed in frame number 506, and about half of cauliflowers were missed in frame numbers 507 and 509. The primary factors influencing missed detection were the sunshine, which affects the filtering of the pixels in HSV space, and the perspective variations as plants move through the field of view. After applying the proposed method (tracking method), most of these missed detections were eliminated. Examining the results using tracking in Fig. 5.3(b), it can be seen that the missed detections have been substantially corrected in frame numbers 248-271, 453-460, 503, 508, and 509. For frame number 507 and 509, the misdetection rate is reduced by 50%. However, a few frames still have misdetections, though the rate is less than 20% for most frames. The performance of both algorithms is summarised in terms of average detections across the entire test set. With detection only (no tracking), the Recall value is 97.28% while addition of tracking results in an increase to 99.3404%, as shown in Table. 5.2.
Figure 5.3. The plotting results of detection and tracking algorithms.
Table 5.2. Comparison of performance for video recorded in sunny conditions

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection Only</td>
<td>97.28.04%</td>
</tr>
<tr>
<td>Tracking (the proposed algorithm)</td>
<td>99.3404%</td>
</tr>
</tbody>
</table>

By applying tracking, there is an improvement in the detection performance of 2.0604

5.4.2 Comparison with Optical Flow based on the Lucas-Kanade algorithm (OF-LK)

To further demonstrate the performance of the proposed method, the proposed algorithm was compared with another tracking algorithm that uses Optical flow based on the Lucas-Kanade (LK) method. While several approaches to optical flow exist, the Lucas-Kanade method [173] is one of the most effective ones. The tracking system includes three main stages: (i) Object detection; (ii) Feature extraction; (iii) tracking features (objects). The detection stage was done by the method presented in Chapter 4. The feature extraction was done based on Harris corners [174]. Since the object of interest here (cauliflower plant) has a somewhat rigid structure, it was found that it is possible to apply Harris corners to extract reasonably good corner features to be tracked over the video sequences. The final stage is to track those extracted corner points using optical flow, whereby detected corners in a given frame will be passed to the OF algorithm as previous points, which along with the next frame will produce new points as output, which then become input points in the next iteration. Fig. 5.4 shows a sample image captured in sunny conditions and the results of each algorithm stage (object detection, feature extraction and tracking).

The recall with the OF-based approach, when used on the same data set as the proposed algorithm, was found to be 80.01%. The following figure shows the results of the OF-LK algorithm as a function of frame number; this can be compared with Figure 5.3 Overall comparison of the OF-LK algorithm with the proposed algorithm is given in Table 5.3.
Figure 5.4. Example image and output of stages of the OF-LK algorithm. (a) Original image with blue annotated rectangle; (b) Detection results; (c) Harris corners results with corner points indicated by red stars; (d) Tracking results of the OF-LK algorithm (red box), and the overlap between tracked plant and ground truth (yellow box).

Figure 5.5. The plotting results of tracking algorithm using Optical Flow based on the Lucas-Kanade algorithm.
5.5 Conclusion

Table 5.3. Comparison of performance for video recorded in sunny conditions

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracking (Optical flow + Lucas-Kanade algorithm)</td>
<td>80.01%</td>
</tr>
<tr>
<td>Tracking (the proposed algorithm)</td>
<td>99.3404%</td>
</tr>
</tbody>
</table>

As can be seen from Fig 5.5, there are many more false detection errors with OF-LK compared to the proposed algorithm (see Fig. 5.3(b)); indeed, the detection algorithm alone exhibits better performance than the OF-LK algorithm (80.01% compared to 97.28%).

The poor performance of the OF-LK algorithm may be due to several reasons: (i) The motion of the camera will result in spurious and noisy features, which will affect the performance of the system; (ii) Feature extraction from the detected plants (corners) will also be affected by lighting variations, plant motion due to wind etc.

5.5 Conclusion

In this chapter, a multi-target tracking algorithm based on Kalman filtering and the Hungarian algorithm was applied to improve performance of the detection algorithm proposed in Chapter 4. The motivation for this was to improve performance particularly in sunny conditions. The Kalman filter was used to predict the next positions of the detected crops in video sequences, while the Hungarian algorithm was used to assign predictions to the tracks.

By adding tracking to the detection algorithm, overall performance was improved from 97.28% to 99.3404%. The proposed tracking algorithm was compared to the Optical Flow with the Lucas-Kanade algorithm, and was shown to display superior performance. The proposed tracking algorithm is also able to address some issues that arose in recent work by Stein et al. [175]. Stein’s method considered any fruit that was occluded by a tree trunk as a new piece of fruit and was therefore counted twice. Because of the prediction property of
Kalman Filter, the proposed method was able to track the cauliflowers even if they partially disappeared from the scene or were mis-detected because of sunshine.

The next chapter addresses a further problem associated with detecting plants using image processing, namely, dealing with plants in conditions of high occlusion.
Chapter 6

Detection of plants in conditions of high overlapping

6.1 Introduction

In smart agriculture applications utilising machine vision to distinguish between plants and weeds, overlapping between plants is one of the most challenging problems, especially for image segmentation and classification. In order to solve this issue, a novel approach based on feature detection from the main stem (main veins of crop leaves) is proposed. The proposed algorithm is divided into three main stages: (i) Crop and edge detection; (ii) Main stem detection and calculation of intersection points; (iii) Establishing cluster centres corresponding to individual plants. The proposed algorithm is used for detection of overlapping plants in two crops of interest: cauliflower and cabbages. The algorithm is described in detail in the following section, followed by description of a method that is used for performance evaluation. Discussion of results is presented in Section 5.3. The work described in this chapter is currently under review for publication in *Image and Vision Computing* journal.
6.2 Proposed Algorithm

The proposed algorithm is based on the assumption that each leaf of the cauliflower or cabbage plant has a map of veins which distribute water and nutrients throughout the leaf. In general, there are two types of veins to consider: one is called the main vein (the thicker vein that divides the leaf into two parts), and the other is called a branched vein (the smaller veins that are branches off the main veins). All the main veins of each plant leaf meet (intersect) at the center of the plant. The goal is to detect the main veins in the leaves and use their intersection to detect the center of the plant. The flow chart of the proposed algorithm is shown in Fig.6.1, while Figure 6.2 gives an example of the output of each step in the algorithm. Each step in the method will now be described in more detail.

6.2.1 Algorithm steps

6.2.1.1 Crop and edge detection

Firstly, it is necessary to detect the cauliflower and cabbage crop itself. Thus, crop and edge detection processes are carried out in a parallel way. For crop detection, in this work, the automatic crop detection algorithm already described in Chapter 4 is applied. The result of the crop detection algorithm for the original image in Figure 6.2(a) is given in Fig.6.2 (b). For edge detection, firstly, the original image is resized to quarter-size in order to reduce the computational requirements for subsequent steps. Secondly, a Gaussian Blur filter with a 3x3 kernel is applied in order remove the tiny veins of cauliflower leaves. The two dimensional Gaussian function can be described as follows:

\[
G(x,y) = \frac{1}{2\pi \sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}
\]

(6.1)

where \(\sigma\) is the standard deviation of Gaussian distribution (which controls the amount of smoothing or blurring). The output image from the blurring process is shown in Fig.6.2(c).
6.2 Proposed Algorithm

Figure 6.1. The flow chart of the proposed algorithm
Thirdly, the blurred image is converted into a grayscale image in order to highlight the most visible features of the target object. The output image of this transformation is shown in Fig. 6.2(d). Fourthly, the Sobel edge detector is applied in order to emphasize the structure of cauliflower leaves, particularly their main veins, and the resulting output is then converted to a binary image. The gradient magnitude of the Sobel operation is calculated for a given image based on $x$ and $y$ gradient components using the following 3x3 kernels:

$$G_x = \begin{vmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{vmatrix} * I(x,y)$$ (6.2)

$$G_y = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{vmatrix} * I(x,y)$$ (6.3)

$$|G| = \sqrt{G_x^2 + G_y^2}$$ (6.4)

The output of the edge detection process is shown in Fig. 6.2(e). Finally, a subtraction operation between the complement of the crop detection image (Fig. 6.2(f)) and the output image from edge detection (Fig. 6.2(e)) is applied in order to remove weeds and other residual edges. The output of the subtraction operation is given in Fig. 6.2(g).

### 6.2.1.2 Main vein detection and determination of intersection point

While the result of the previous step produces edges in the image (e.g. Figure 6.2(f)), it is the main veins that are of primary interest, and these appear as strong lines in the edge detected image. To isolate these, the Hough transform is applied to the image. Lines can have horizontal, vertical or sloped orientation.
The Hough transform is a feature extraction technique widely used in computer vision and related areas, to isolate features of particular shapes. This is carried out in a parameter space from which object candidates are obtained as local maxima. The original Hough transform was used for identification of lines, but the method was extended to identify positions of circles, ellipses, and other arbitrary shapes. The Hough transform as it is universally used today was invented by Duda and Hart [176], who called it a "generalized Hough transform". Duda and Hart used angle-radius parameterization as a general representation for lines of all orientations, whereby all lines can be presented in the polar system as in follows:

\[ r = x\cos \theta + y\sin \theta \quad (6.5) \]

where \( x \) and \( y \) are the coordinates of the pixel in the edge image, \( r \) is the distance or radius from the origin of the image’s coordinate system to the line, and \( \theta \) is the angle between the image’s x-axis and the line. By using this equation, all lines can be represented by \( \theta \in [0, 180] \) and \( r \in \mathbb{R} \) (or \( \theta \in [0, 360] \) and \( r \geq 0 \)), where \( \mathbb{R} \) is real number. In this work, \( \theta \in [0, 180] \) is used.

To reduce the number of lines that need to be considered, since not all straight lines represent main veins, a minimum length of the target line (main veins) and a minimum gap between lines have defined. The minimum length of the line is set to 30 while the minimum gap between segmented lines is set to 7. These values were selected empirically based on the training images (800 images contain pair of cauliflowers and two cabbage spices (one with flat leaves and other with frizzy ones), which are separated from the testing images). These values depend on several factors such as the position of the camera and growth stage of plants; this is discussed further later in the chapter. Fig. 6.2(h) shows the selected lines following this process, representing the main veins of the crop leaves.
Once the target lines are detected, the intersection points for each possible combination of pairs of lines can be calculated. The intersection points that are obtained from the detected target lines are shown in Fig. 6.2(i).
6.2.1.3 Finding the center of the first and next cluster

The intersection points of clusters of leaf veins are used as the starting point for finding the centers of the individual plants. Assuming two plants are overlapping, to find the center of the first and second plants, it is assumed that the overlap region follows the form of two overlapping circles as shown in Fig. 6.3. The green circle represents plant A and the orange one represents plant B, and inside each of them is several random small points (shown in red), which represent the intersection points (see Fig. 6.2(i)). The following steps are used to calculate the centers of two overlapping crops (cauliflowers or cabbages).

1. Calculate the distances between all intersection points as follows:

\[ D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \] (6.6)

Figure 6.3. An example of two overlapping objects (A and B)
where $D$ is a matrix of all of the distances between points, and the $ij^{th}$ element $D_{ij}$ is the distance between points $i$ and $j$. The values $x_i$ and $y_i$ are the coordinates of the first intersection point while $x_j$ and $y_j$ are the coordinates of the second intersection point.

2. Search for intersection points that have a distance of 6 pixels or less from each other (the threshold of 6 was obtained empirically). Count how many intersection points are close to each of the original points and save them in a new vector called $CountM$. Order the elements of $CountM$ from highest to lowest.

3. Find the coordinates of the three intersection points that have the largest number of neighbours within the threshold distance; call these $Max1(x,y),Max2(x,y)$ and $Max3(x,y)$

4. Calculate the average of these three co-ordinates, which is taken to represent the centre of the first cluster, and is treated as the centre of a plant.

$$Avg(x,y) = \frac{Max1(x,y) + Max2(x,y) + Max3(x,y)}{3} \quad (6.7)$$

5. To find the next cluster, set all points within 150 pixels (obtained empirically) of the first plant centre (calculated according to equation (6.7)) to zero, and repeat steps 3 and 4.

### 6.2.2 Performance evaluation

Evaluation of the segmentation performance for overlapping objects is more difficult to perform compared with the segmentation of a single object because of the occlusion of the boundaries of overlapping objects. Since the centroid points are used to identify two overlapping objects, the evaluation is conducted based on calculation of the Euclidean distance in pixels between the actual centroid of each plant (as the gold standard, determined
manually from the test data through visual inspection) and the automatically detected centroid of the same plant (obtained by the proposed method).

The distance error can be computed as follows:

\[
\text{Distance error} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
\]

(6.8)

here \(x_1, y_1\) are the actual coordinates of centroid of the crop determined manually from the test data, and \(x_2, y_2\) are the detected coordinates of centroid of the crop.

6.3 Results and Discussion

The test images were captured in cauliflower and cabbage fields in the same conditions as the data sets used in earlier chapters, and as described in section 4.5.1. This experiment includes one cauliflower species and two varieties of cabbage: one with flat leaves and another with textured leaves. The crops (cauliflowers and cabbages) used in this experiment are in late stages of growth where crops are overlapped with each other. The partial occlusion of crop leaves caused by weeds is also present in the data set, thus adding to the challenge. Examples of the output of the proposed algorithm for typical test images (including heavy weed coverage) are shown in Fig.6.4.

From Fig.6.4 (a) it can be seen that the two overlapping cauliflowers were detected individually based on the detection of their centroids. As can be seen from Fig.6.4 (b), the cauliflowers are presented in heavy weeds, however, the centroids can still be detected even when partially occluded by weeds (the cauliflower on the right-hand side). From Fig.6.4 (c), which represents an example of cloudy conditions, and (d) in sunny conditions, it can be seen that the two overlapping cabbages with flat leaves were detected individually based on the detection of their centroids. From Fig.6.4 (e) and (f) it can be seen that the two overlapping cabbages plants with textured leaves were detected individually under sunny conditions.
Figure 6.4. An example of results from the proposed algorithm. (a) Two overlapping cauliflowers in normal conditions (low weed coverage). (b) Cauliflower occluded by heavy weeds. (c) Two overlapping cabbages with leaves of flat surfaces in normal conditions. (d) Two overlapping cabbages with leaves of flat surfaces in sunny conditions. (e) Two overlapping cabbages with leaves with textured surfaces in sunny conditions. (f) Two overlapping cabbages with leaves of textured surfaces in sunny conditions. The blue rectangle represents the first plant, and the green one represents the second plant. These rectangles are automatically drawn by the proposed algorithm. Red dots indicate the location of the centre of each crop automatically determined by the algorithm.
6.3 Results and Discussion

The proposed algorithm is effective even when the crop is partially occluded by weeds (as can be seen in Fig. 6.4(b)). Thus, the advantage of using this algorithm is that as long as at least two main veins are detected this can produce one intersection point that can be used as the predicted centroid of the plant.

To assess the performance of the proposed algorithm quantitatively, test images with crops (cauliflowers and cabbages) overlapping with each other in groups of two were used. Two hundred individual plants exist in the test set (100 images, each containing one pair of plants). The distance error of those samples was calculated. Fig. 6.5 illustrates the distance error for each tested sample.

![Distance error graph](image)

Figure 6.5. The distance error of the proposed method for each test image.

As can be seen from Fig 6.5, the peak error does not exceed 1.5 centimetres. The average distance error over the data set is equal to 0.375 centimetres (indicated by the straight line in
Fig 6.5. To further analyse the cumulative error performance of the proposed algorithm is plotted as a function of distance error in Fig. 6.6.

![Overall performance](image)

Figure 6.6. The overall performance of the proposed algorithm

As can be seen from the Fig 6.6, the locations of 185 crops (92.5% of test samples) were correctly detected with distance error of less than one centimetre. Only 7.5% of test cases have a distance error greater than 1 centimetre. Analysis of these higher error cases reveals the likely cause of failure to be bending of the plant leaves, which affects the intersection point and leads to the deviation of the estimated intersection point from the true value.

The algorithm includes some empirically chosen thresholds, which will depend particularly on the position of the camera (the distance between camera and the plant) and the growing stage of the plant. Several algorithm parameters must be tuned, based on the use of training images. In this study, the distance between the camera and the plant is fixed (160 cm), and the algorithm parameters were chosen based on that (e.g. the minimum length
of segmented line of the main vein is 30 pixels, approximately 3.16 cm in actual size, or 0.79 cm in the resized image). However, to apply the proposed algorithm in real time, this parameter will change according to the actual camera height, which may vary in the case of an uneven surface in the field. To address this issue, it is assumed that, based on a known set of training data (as for example used in this study), the optimum minimum length ($L_1$) of the main vein at a known camera height ($H_1$) can be estimated. In practice, the actual height ($H_2$) may vary, however, it is assumed that $H_2$ is known from GPS data or any other distance measuring device. This information can be used to determine the minimum length of the main vein, denoted by $L_2$, as illustrated in Figure 6.7. In this figure, $H_1$ represents the height of the first triangle and $L_1$ the length of the main vein (derived empirically from a set of training data). Similarly, $H_2$ represents the height of the second triangle and $L_2$ the length of the vein.

From triangle similarity it can be stated as:

$$\frac{H_1}{H_2} = \frac{L_1}{L_2}$$

(6.9)

Then, the new minimum length of the main vein can be obtained as follows:

$$L_2 = \frac{L_1 \times H_2}{H_1}$$

(6.10)

This analysis assumes that $H_2$ is known accurately. However, while technologies such as Differential GPS (DGPS) are quite accurate, there will still be some errors caused by inherent error in the DGPS signal, or due to uneven ground as the vehicle on which the camera is mounted vibrates. Specifically, $H_2$ might be slightly more or less than the actual camera height. As result, this will cause an error in $L_2$. If the camera is too high and $H_2$ is underestimated, some main veins will be missed, while if the camera is too low, some branched veins may be erroneously selected as main veins. However, a strength of the
Figure 6.7. An example of different camera attitudes and their corresponded appearance of the main veins of the same cauliflower. The green lines represent the main vein.
algorithm is that, as long as there is sufficient density of overlap points to determine centroids, not all main veins need to be detected.

Parameters are determined based on different images that consists of two adjacent crops with different degree of occlusions (partially to highly overlapped). The parameters selected with the training data are then used in the system for testing with completely different images.

6.4 Conclusions

In this chapter, a novel algorithm for automatic detection of the location of plants, where overlapping occurs between adjacent plants, was presented. The proposed algorithm demonstrated high performance in detecting the centroids of overlapping plants (cauliflower and cabbage) under various weather conditions such as cloudy and sunny conditions, in particular, 92.5% of test images had distance error less than 1 centimetres when compared to manually annotated plant locations. The primary advantage of the proposed method is that the center of cauliflower or cabbage can be defined even in cases where there is high occlusion caused by other plants, as long as there is an intersection point detected between two main veins. In addition, the proposed algorithm has demonstrated good performance over a database containing more than one crop type, and can be readily applied to other crops that have similar vein structures. A shortcoming associated with the proposed algorithm is that where bending of leaves occurs, this will cause a deviation of detected centroid away from the actual centroid. As a result, the distance error will increase.

The next chapter in this thesis revisits the topic of detection, using alternative techniques to those already discussed, in particular machine learning techniques such as deep neural networks and Support Vector Machines (SVM).
Chapter 7

Plant Classification using Machine Learning

7.1 Introduction

Although the proposed plant detection algorithm that was introduced in this thesis has shown very good performance against various weather conditions, and the tracking algorithm has played an important role in addressing the mis-detections that occurred, particularly in sunny conditions, the information about another class (weeds) still needs to be obtained. In this chapter, the use of deep learning approaches for plant classification (cauliflower and weeds) in smart agriculture applications was investigated. To perform this, five approaches were considered, two based on well-known deep learning architectures (AlexNet and GoogleNet), and three based on Support Vector Machine (SVM) classifiers with different feature sets (Bag of Words in L*a*b colour space feature, Bag of Words in HSV colour space, Bag of Words of Speeded-up Robust Features (SURF)). Two types of datasets were used in this study: one without Data Augmentation and the second one with Data Augmentation. Each algorithm’s performance was tested with one data set similar to the training data, and a second data set acquired under more challenging conditions such as various weather conditions, heavy
incidence of weeds in cauliflower images, and several weed species that have similarity of colour and shape to the crops. The work described in this thesis is in preparation for submission to the relevant conference.

## 7.2 Machine Learning Algorithms and Features

### 7.2.1 SVM model

Support Vector Machines (SVMs) [177, 178] are supervised learning models with associated learning algorithms that analyze data and recognize patterns. They are used for classification and regression analysis. SVM can perform both linear classification and non-linear classification. SVMs are used with features extracted from the input data; in this work, the Bag of Words (BoWs) approach is used with different base features.

### 7.2.2 Bag of Words (BoWs)

The BoWs [179] features of images can be obtained by using the K-Means clustering algorithm on features extracted from those images. The features may be shape or structure features, or colour features. The algorithm iteratively groups the descriptors into k mutually exclusive clusters. The resulting clusters are compact and separated by similar characteristics. Each cluster center represents a feature, or visual word. Three separate features are considered here to create individual BoWs representations: one based on feature descriptors (SURF [180]) and two colour features (L*a*b and HSV colour spaces).

To create Bag of Words features from the colour information, the following steps are used.

1. Convert RGB images to the L*a*b* or HSV colour space.
2. Compute the "average" L*a*b* or HSV colour within 16-by-16-pixel blocks. The average value is used as the colour portion of the image feature.

3. Reshape L*a*b* or HSV average values (features) into a Kx3 matrix, where K is the number of features.

4. Normalize each channel to the root mean squared value of the channel.

5. Augment the colour feature by appending the [x y] location within the image from which the colour feature was extracted. This technique is known as spatial augmentation. Spatial augmentation incorporates the spatial layout of the features within an image as part of the extracted feature vectors. Therefore, for two images to have similar colour features, both the colour and the spatial distribution must be similar.

6. Normalize pixel coordinates to handle different image sizes.

7. Concatenate the spatial locations and colour features.

8. Add the variance of each channel as an additional feature.

### 7.2.3 Deep learning approaches

In this work, two common deep learning approaches were used for plant classification: AlexNet [113], GoogleNet [114]. These were applied individually on raw images (RGB colour images), and the ones that had preprocessing.

#### 7.2.3.1 AlexNet

AlexNet [113] is a neural network model with 60 million parameters in 25 layers and is available pre-trained on the ImageNet database which contains more than a million images in 1000 categories. Specifically, it consists of five convolutional layers followed by three fully connected layers as illustrated in Fig. 7.1. The main purpose of the convolutional layer is to
extract features from the input images while the fully connected layers are used to classify the extracted features to the desired class.

![Architecture of AlexNet](image)

**Figure 7.1.** An illustration of the architecture of AlexNet [113]

### 7.2.3.2 GoogleNet

GoogleNet [114] has 21 layers, with fewer parameters than AlexNet (about 7 million) [181]. GoogleNet has a different architecture to AlexNet, and uses combinations of “inception” modules, each including some pooling, convolutions at different scales and concatenation operations. It also uses 1x1 feature convolutions that work like feature selectors. The advantage of using 1x1 convolutions is to reduce the number of parameters. These components are shown in Fig. 7.2.

### 7.3 Testing framework and performance metrics

In this project, there are two types of training data: One is original dataset (raw images) and second data is Data augmentation (preprocessing images).
7.3 Testing framework and performance metrics

7.3.1 Training Set

The datasets used in this work were derived from the same data base used in previous chapters, and consist of images converted to size 227x227 for training and testing purpose; the size conversion is necessary to match the expected input size of the neural networks employed for classification. A set of 800 cauliflowers images and a set of 1000 weed images was used. 80% of the dataset (randomly chosen) was used for training [108] (referred to as the training set), and the remaining (20%) for testing (referred to as the test set). The characteristics of the training and testing sets were broadly similar. Fig. 7.3 shows examples of the training set.

7.3.2 Data Augmentation (DA)

Since there is relativity small original dataset and to perform better classification accuracy, this dataset needs to increase. That can be achieved by augment the original images via several random transformations (pre-processing) to generate new training samples (more
Figure 7.3. Sample images from training set. Figure (a) represent the positive set (cauliflowers), while figure (b) represent the negative set (weeds)

data) without changing the classes labels. This process called Data augmentation. In this project, seven image transformations is done on the original dataset. These operations are:

1. **Rotation-range** is a value in degrees (0-180), a range within which to randomly rotate pictures. The used value is 40.

2. **Width-shift and height-shift** are ranges (as a fraction of total width or height) within which to randomly translate pictures vertically or horizontally. The value of 0.2 is used for both shifts.

3. **Rescale** is a value by which is used to multiply the data before any other processing. The used value is 1/255.

4. **Shear-range** is for randomly applying shearing transformations. The used value is 0.2

5. **Zoom-range** is for randomly zooming inside pictures. The used value is 0.2.

6. **Horizontal-flip** is for randomly flipping half of the images horizontally.

7. **Fill-mode** is the strategy used for filling in newly created pixels, which can appear after a rotation or a width/height shift.
Thus, the total generated data is 12600 images (5600 images for cauliflower and 7000 for weeds). 80% of the DA (randomly chosen) was used for training, and the remaining (20%) for testing.

### 7.3.3 Challenge set

In addition to the test set described above, a further 200 images (100 images for cauliflowers and another 100 images for weeds) were selected for evaluating the algorithms. These images were selected carefully with more challenging conditions than used in the original training and test sets, and include cauliflowers surrounded by heavy weeds, leaves of cauliflowers turned over or otherwise distorted, shadows, sunshine, weeds with similar colour and shape to cauliflowers, and blurred images etc. This data set is referred to as the challenge set. Fig 7.4 shows examples from this set.

![Sample images of Evaluation set. Figure (a) represent the positive set (cauliflowers), while figure (b) represent the negative set (weeds)](image_url)
7.3.4 Performance Evaluation

To evaluate crop vs. weed accuracy of the given algorithms, the primary metric used is the classification accuracy, defined as follows:

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{TrueNegative} + \text{FalsePositive} + \text{FalseNegative}}$$

(7.1)

For precision agriculture, it is important to keep the number of False Negatives small, which means keeping the number of crops classified as weeds (and thereby removed) as small as possible.

7.4 Results and Discussion

7.4.1 Parameter Selection

For GoogleNet and AlexNet, pre-trained networks were used, with transfer learning using the training set to retrain the output layers. To retrain GoogleNet to classify new images, the last three layers of the original network were retrained to produce three new layers (a fully connected layer, a softmax layer, and a classification output layer). A similar approach was taken to apply transfer learning with AlexNet.

The AlexNet training parameters were as follows: Number of iterations: 1400, train batch size: 10, base learning: 0.00001, Epoch: 10.

The GoogleNet training parameters were as follows: Number of iterations: 890, train batch size: 10, base learning: 0.00001, Epoch: 10. In adaptation, a new layer was added to the layer graph in GoogleNet: a dropout layer with a probability of 60% dropout. The advantage of this layer is that it can prevent Neural Networks from overfitting and improve the quality of features [19, 20]. For all of the SVM+BoWs feature combinations, the parameters were set to the same values. The features from the training images were processed by K-Means
Table 7.1. Average test accuracy on Test Set

<table>
<thead>
<tr>
<th>Model</th>
<th>Without Data Augmentation</th>
<th>With Data Augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cauliflower</td>
<td>Weed</td>
</tr>
<tr>
<td>AlexNet</td>
<td>100.00%</td>
<td>99.44%</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>97.62%</td>
<td>100.00%</td>
</tr>
<tr>
<td>SVM+BoWs(HSV)</td>
<td>96.00%</td>
<td>99.00%</td>
</tr>
<tr>
<td>SVM+BoWs (L<em>a</em>b)</td>
<td>97.00%</td>
<td>99.00%</td>
</tr>
<tr>
<td>SVM+BoWs (SURF)</td>
<td>97.00%</td>
<td>86.00%</td>
</tr>
</tbody>
</table>

clustering to create a 500-word visual vocabulary. Once the features are extracted, the SVM classifier is applied. SVM parameters were empirically optimized to obtain the best performance, with the following parameters used: the SVM kernel was chosen as ‘rbf’, the cost parameter (c) is set to 1.8, and Gamma (the kernel width) is set to 0.09.

### 7.4.2 Results from test Set

Table 7.1 illustrates the accuracy of each algorithm for the test set on the test dataset. Two versions of the test set were used for evaluation: the original test set, and a version of the test set to which data augmentation techniques were applied. Data augmentation [182–184] is commonly used in deep learning applications and involves increasing the size and variability of a dataset through transformations of the original data. These transformations typically include e.g. blurring, rotation and translation operations. These transformations generated 12600 images (5600 images for cauliflower and 7000 for weeds). The columns marked "Cauliflower" and "Weed" give detection rate on those subsets of the test set, while "Overall" accuracy is the overall performance on the database.

From Table 7.1, it can be seen that the deep learning approaches gave the highest average test accuracy for dataset without Data Augmentation (99.72% and 98.81% for AlexNet and GoogleNet, respectively) and for dataset with Data Augmentation (99.79% and 99.83% for AlexNet and GoogleNet, respectively), whereas the SVM+BoWs of SURF features has the lowest performance at 91.50% and 93.00% for dataset without Data Augmentation and
Table 7.2. Overall results on the challenge Set

<table>
<thead>
<tr>
<th>Model</th>
<th>Without Data Augmentation</th>
<th>With Data Augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cauliflower</td>
<td>Weed</td>
</tr>
<tr>
<td>AlexNet</td>
<td>95.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>96.00%</td>
<td>99.00%</td>
</tr>
<tr>
<td>SVM+BoWs(HSV)</td>
<td>94.00%</td>
<td>93.00%</td>
</tr>
<tr>
<td>SVM+BoWs (L<em>a</em>b)</td>
<td>89.00%</td>
<td>98.00%</td>
</tr>
<tr>
<td>SVM+BoWs (SURF)</td>
<td>82.00%</td>
<td>95.00%</td>
</tr>
</tbody>
</table>

dataset with Data Augmentation, respectively. The SVM+BoWs of HSV and L*a*b colour features show good performance (97.5 and 98.0%, respectively) for dataset without Data Augmentation, and (99 and 99.5%, respectively) for dataset without Data Augmentation.

In total, all models achieved higher classification accuracy when they applied on the test set of the dataset with Data Augmentation than the one without Data Augmentation. The next sub-section discusses performance on the more difficult challenge set.

From Table 7.1, it can be seen that the deep learning approaches gave the highest average test accuracy (99.72% and 98.81% for AlexNet and GoogleNet, respectively), whereas the SVM+BoWs of SURF features has the lowest performance at 91.50%. The SVM+BoWs of HSV and L*a*b colour features show good performance (98.0%). In total, all models achieved high classification accuracy on the test set. The next sub-section discusses performance on the more difficult challenge set.

### 7.4.3 Results from challenge Set

Table 7.2 summarises the performance of each algorithm and the overall test accuracy on the Challenge Set by using the output of training phase for both variants of the challenge dataset (without and with Data Augmentation) for all models tested.

According to the results in Table 7.2 all of the models without Data Augmentation have higher accuracy on the weed images than cauliflower images, except for the SVM+BoWs based on the HSV colour space. For images correctly classified as cauliflowers, GoogleNet
demonstrated the highest classification accuracy without Data Augmentation and GoogleNet and AlexNet demonstrated the highest classification accuracy with Data Augmentation, whereas the SVM+BoWs of SURF gave the lowest classification accuracy in both types of dataset. AlexNet demonstrated the second highest classification accuracy for the dataset without Data Augmentation. The SVM+BoWs of HSV colour space achieved the third highest classification accuracy for the dataset without Data Augmentation, whereas the SVM+BoWs of L*a*b colour achieved the third highest classification accuracy when the dataset with Data Augmentation was used. The SVM+BoWs of L*a*b and HSV colour gave similar classification accuracy when they used the dataset without Data Augmentation and with Data Augmentation.

In terms of overall test accuracy (the last column in Table 7.2), AlexNet demonstrated the highest overall accuracy results (99.40%) and GoogleNet demonstrated the second highest overall accuracy results (98.81%) with Data Augmentation, whereas AlexNet and GoogleNet demonstrated the highest overall accuracy results (97.5%) without Data Augmentation. The SVM+BoWs of SURF features gave the lowest performance at 88.5% and 91.50% for the dataset without and with Data Augmentation, respectively. The SVM+BoWs of HSV and L*a*b colour features achieved performance of 93.5% on the dataset without Data Augmentation, whereas when the dataset with Data Augmentation was used, the SVM+BoWs of L*a*b demonstrated higher classification accuracy (96.5%) than SVM+BoWs of HSV (95.5%).

In conclusion, deep learning approaches outperformed the other systems in terms of accuracy on the two types of dataset. Although both used deep learning approaches with different network architectures (AlexNet has 25 network layers and a greater number of parameters, while GoogleNet has fewer network layers and fewer parameters), they demonstrated similar classification accuracy on the dataset without Data Augmentation. However, when the dataset increased, AlexNet demonstrated higher classification accuracy than GoogleNet.
The SVM classifier has shown results that are competitive with the more sophisticated methods like AlexNet, especially with BoW of L*a*b and HSV colour spaces for the two types of dataset. Moreover, SVM+BoWs of HSV colour space gave the highest correct classification of cauliflower (as can be seen in Table 7.2) for the two types of dataset. This may be because HSV is an “intuitive” colour space that is aligned with human colour perception [149] and is somewhat robust to illumination variation [185]. The SVM+ BoWs of SURF features exhibited lower accuracy than other features with SVM for the two types of dataset. One reason for this may be that SURF is not robust to illumination variation [186].

In total, all models achieved higher classification accuracy when they were applied on the challenge set of the dataset, especially when trained on the dataset with Data Augmentation.

7.5 Conclusions

In this chapter, a number of approaches were applied to the task of plant classification to distinguish between cauliflowers and weeds. Two types of dataset were used in this project, one without Data Augmentation and other with Data Augmentation. In addition, the approaches were tested on two different test data sets, one with similar characteristics to the training data, and one with more challenging characteristics (the Challenge set). The results show that most of the systems tested are capable of achieving good performance on the Challenge set. In addition, with help of Data Augmentation, all systems increased their classification accuracy compared to the ones without Data Augmentation. AlexNet, based on deep learning, demonstrated the highest plant segmentation accuracy (97.50% and 99.40%) on dataset without Data Augmentation and with Data Augmentation, respectively. The SVM+BoWs of L*a*b and HSV colour space also demonstrated good performance, comparable to the more sophisticated CNN-based AlexNet and GoogleNet.

Overall, this study demonstrated the utility of a range of approaches for plant and weed classification. Future work will include further testing with a larger data base with more
7.5 Conclusions

plant species, in similarly challenging conditions, and comparison with other deep learning architectures.
Chapter 8

Conclusions and Future Work

8.1 Project Summary and Conclusions

This thesis has considered some of the issues that face computer vision technologies in agricultural applications. Some of those issues related to the environment itself such as various weather conditions, shadow, and damaged plants. Others are caused by image capture conditions such as camera vibration and variations in speed.

A comprehensive and critical survey on image-based plant segmentation techniques has been conducted as given in Chapter 3. This survey reviews the three primary segmentation approaches, namely, (i) colour index-based segmentation, (ii) threshold-based segmentation, (iii) learning-based segmentation. The study focused intensively on colour index-based segmentation as it has been used most widely in the literature. The study also highlighted the advantages and the limitations for each algorithm. Although several plant extraction based approaches have achieved good segmentation performance, several challenges remain such as effect of lighting conditions, shadow, and complex backgrounds. Finally, recommendations for segmentation algorithms for use in different conditions were given.

A novel algorithm for automatic crop detection in video sequences under various weather conditions was proposed in Chapter 4. Since the HSV colour space is aligned with human
colour perception, it was applied for discriminating crop, weeds and soil. In addition, shape analysis was used for determining the crop region. The performance of the algorithm was assessed by comparing its performance with ground truth methods (manual annotation). A sensitivity of 98.91% and precision of 99.04% was achieved. A shortcoming associated with the proposed algorithm is its reliance on colour. If the cauliflower leaf colour changes because of disease, or because of very sunny conditions, the misclassification rate may increase.

In order to increase the robustness in the crop detection algorithm (decreasing the misclassification rate), particularly in sunny conditions, a multi-target tracking algorithm based on Kalman filtering and the Hungarian algorithm was proposed in Chapter 5. The recall matrix was used to evaluate the detection and tracking performance. With the help of tracking, detection failures were reduced, especially in sunny conditions, such that overall detection performance was raised from 97.28 to 99.34%.

Occlusion is one of the most difficult problems in machine vision applied in outdoor scenes since occluded objects are difficult to identify. In order to address this issue, a novel algorithm for detection of plant locations in conditions of high overlapping, based on feature detection from the main stem (main veins of cauliflower leaves), was proposed in Chapter 6. Results of the proposed algorithm show that the locations of 92.5% of cauliflowers were correctly detected with distance error of less than one centimetre, in conditions of high occlusion.

The topic of actual classification of plants and weeds was considered in Chapter 7. Plant classification is important in smart agricultural applications because it can give clear information about crops or weeds, so the right treatment system can be applied, e.g. use of fertiliser or herbicide. Two well know deep neural network architectures (AlexNet and GoogleNet) were used. In addition, SVM classifier with Bag of Words feature sets was applied, based on different feature extraction approaches: L*a*b colour space, HSV color space, and Speeded-up Robust Features (SURF). Results demonstrated that the best overall
performance was achieved by AlexNet, while the SVM-based approaches achieved good performance, at a substantially lower computational cost. Data augmentation was also found to be helpful in obtaining better performance.

### 8.1.1 Primary Contributions

The primary contributions of this thesis are as follows:

- Comprehensive review of plant segmentation approaches and analysis of their strengths and limitations conducted. Segmentation algorithms for use in different conditions based on analysis of their performance based on studies from the literature were suggested.

- A new algorithm for crop detection under various weather conditions based on colour space and shape analysis was proposed, and shown to demonstrate high performance in different conditions.

- A more robust algorithm using multi-object tracking was proposed to deal with particularly difficult sunny conditions.

- A new algorithm for separating partially and highly occluded plants was proposed.

- Two well-known deep learning architectures (AlexNet and GoogleNet), and three based on Support Vector Machine (SVM) with different feature sets (Bag of Words in L*a*b colour space feature, Bag of Words in HSV colour space, Bag of words of Speeded-up Robust Features (SURF)) were applied to classify crops and weeds.

### 8.2 Suggestions for Future Work

For better recognition of cauliflower at different growing stages, machine vision systems need to be able to identify features in a similar way to humans. A person can readily
distinguish cauliflower plants from weeds with little training, and regardless of cauliflower being occluded, curled, lying down or eaten by insects. A possible way to increase the recognition performance of a machine vision system is to incorporate a side view (the profile of the crop) with the top view. This option could provide more information about plant characteristics and thus would lead to more correct recognition of cauliflower plants and weeds.

In addition, for the plant detection algorithm proposed in Chapter 4, the appropriate color indices were used based on their prevalence in the literature against various weather conditions (cloudy, overcast, and sunny), as outlined in Chapter 3. As can be seen from Table 3.4, for cloudy day, CIVE and COM1 are suggested, while for overcast conditions, CIVE and ExGR are good candidates, and for sunny conditions, COM2 or ExG are appropriate because both are good for addressing shadow. Then, to make the plant detection algorithm more automated, the weather conditions could be identified, and this information can be used to automatically select the most appropriate index for the local conditions.

According to the survey conducted by Mohandes et al. [187] the combination of several individual classifiers can enhance the overall performance of image classification tasks in several applications. For plant classification algorithm, the plant classification accuracy can be increased by combing different classifiers (e.g. DNNs and SVMs) and make a final decision on evidence collected from those classifiers, e.g. through majority voting or other decision fusion means.

While this thesis focused on the problems of detection and classification, the overall goal is to develop real-time systems for automated crop control and weeding, so the algorithms presented here should be evaluated in the context of a weeding system to determine functional performance. In fact, processing time is a major concern in real-time machine vision applications. Since the ultimate goal of this research is to develop a real-time robotic control system, computationally intensive steps should be avoided, or implemented using
platforms that permit real-time operation. Although the algorithms presented in this thesis have demonstrated very good performance in crop detection and plant classification, most of these algorithms involve several image processing steps, particularly in crop detection and occlusion handling algorithms. In general, object detection is much more computationally-intensive, and energy-consuming compared to the image classification task, since numerous possible object proposals need to be evaluated. As result, it is difficult for object detection algorithms to be integrated into embedded systems with limited computing resources and energy supply [188].

In the crop detection algorithm, morphological operations such as erosion and dilation were applied. Erosion with a 3x3 structuring element was applied to remove the remaining isolated white pixels (corresponding to weeds) from the binary image. Dilation with a 7x7 structure element was employed to accentuate the shape of the remaining objects (crops). These two operations contribute to an increase in the computation time of the algorithm, particularly when they use different size of structure elements [189]. To reduce the computation time of the algorithm and hence support real-time operations, it can be implemented on a GPU rather than a CPU as it will speed up the processing time substantially compared to using CPU [189].

The improved detection algorithm incorporates tracking, leading to more computation. Again, in order to decrease the computation time, the algorithm could be implemented on a GPU.

The occlusion algorithm also involves several processing steps which contribute to greater computation. The algorithm already consists of two steps: crop detection algorithm (the second contribution of this thesis) and edge detection. For edge detection algorithm and main vein detection, one of the critical steps is applying the Hough transform to detect main veins of the plant. The Hough transform algorithm can be computationally expensive when implemented sequentially, however, one possible solution is to optimize the algorithm
using parallel computing [190]. Another approach to decrease the computation time for the algorithm is to select (or vote) between 2 to 3 lines (main veins) as the strongest candidates, rather than detecting all lines.

For plant classification, particularly using deep learning approaches (AlexNet and GoogleNet), a further issue (apart from computation) that could arise in real-time is storage capacity for DNNS. As discussed in Chapter 7, AlexNet has 60 million parameters, while GoogleNet has approximately 7 million. One possible way to reduce parameter numbers is by choosing smaller filter sizes (e.g. 3x3) for convolution layers and max pooling layers. In addition, implementation of deep learning classifiers using GPUs is commonly done.

Finally, more extensive testing of the algorithms proposed in this thesis should be conducted.
References


References


References


