



การใช้เทคนิคประมวลผลภาพแบบไม่มีผู้สอนเพื่อตรวจจับการเจริญเติบโตที่ผิดปกติของต้นไม้

Using unsupervised image processing techniques to detect anomaly
growth of plant

จัดทำโดย

นางสาว ปวีณา	อินทสรสวรรค์	58102010811
--------------	--------------	-------------

นางสาว วาสนา	เจริญศิวาแพทย์	58102010824
--------------	----------------	-------------

อาจารย์ที่ปรึกษา

อาจารย์ ดร. ศิริสรพร เหล่าหะเกียรติ

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2561



คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงการ	การใช้เทคนิคประมวลผลภาพแบบไม่มีผู้สอนเพื่อตรวจจับการเจริญเติบโตที่ผิดปกติของต้นไม้ Using unsupervised image processing techniques to detect anomaly growth of plant
นิสิต	นางสาว ปวีณา อินทสรวรรค์ 58102010811 นางสาว วาสนา เจริญศิวาวาทย 58102010824
ปริญญา	วิทยาศาสตรบัณฑิต (วท.บ.)
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงการ	อาจารย์ ดร. ศิริสรพร เหล่าหะเกียรติ

ลงชื่อ

(อาจารย์ ดร. ศิริสรพร เหล่าหะเกียรติ)

อาจารย์ที่ปรึกษาโครงการ

บทคัดย่อ

การใช้ชีวิตในเมืองในปัจจุบันเต็มไปด้วยความเร่งรีบและมลภาวะซึ่งส่งผลเกี่ยวกับปัญหาด้านสุขภาพ วิธีการหนึ่งในการแก้ไขปัญหาด้านสุขภาพคือการปลูกผักเพื่อการปรุงเพื่อใช้ในการปรุงอาหารที่สะอาด อย่างไรก็ตามสวนผักที่ดูแลนั้นต้องการความใส่ใจเป็นอย่างมากในการสังเกตความผิดปกติของพืช เพื่อลดภาระในส่วนนี้ผู้วิจัยจึงเสนอระบบอัตโนมัติหลายระบบสำหรับตรวจจับความผิดปกติของพืช

ในงานวิจัยนี้เราใช้หลักการประมวลผลภาพเพื่อตรวจจับต้นไม้และใช้หลักการเรียนรู้แบบไม่มีผู้สอน (outlier detection) เพื่อตรวจจับต้นไม้ที่เติบโตผิดปกติ โดยทำการทดสอบกับต้นผักบุ้งจำนวน 13 ต้น

ผลลัพธ์จากการทดลองได้ตรวจพบว่ามีต้นผักบุ้ง 2 ต้นจาก 13 ต้นมีการเจริญเติบโตแบบผิดปกติ

Abstract

Living urban life nowadays is full of stress and pollutants, leading to unprecedented concerns over health problems. One way to address the health problem is to grow vegetables for clean cooking. However, a caring vegetable garden requires much effort in observing irregularities of the plants. To reduce the load in this respect, the researcher proposed several automatic systems for detecting abnormalities of the plants.

In this study, we used the image processing principle to detect the trees and used the unsupervised learning (outlier detection) principle to detect trees that are being grown abnormally. By testing with 13 Chinese morning glory trees

The results showed that there were 2 maize trees from 13 plants that were unusually grown.

กิตติกรรมประกาศ

การใช้เทคนิคประมวลผลภาพแบบไม่มีผู้สอนเพื่อตรวจจับการเจริญเติบโตที่ผิดปกติของต้นไม้ (Using unsupervised image processing techniques to detect anomaly growth of plant) สำเร็จลุล่วงได้ด้วยความกรุณาช่วยเหลือ ให้คำแนะนำและคำปรึกษาจาก ดร.ศิริสรพ เหล่าหะเกียรติ ที่ปรึกษาของโครงการ ที่กรุณาเสียสละเวลาเพื่อให้คำปรึกษาเสนอแนวทางในการแก้ปัญหาและแก้ไขข้อบกพร่องที่เกิดขึ้นระหว่างการทำงาน คณะผู้จัดทำกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณภาควิชาวิทยาการคอมพิวเตอร์ที่สนับสนุนอุปกรณ์ สถานที่ ในการทำโครงการ และ ขอขอบพระคุณอาจารย์ทุก ๆ ท่านในภาควิชาวิทยาการคอมพิวเตอร์ที่กรุณาให้ความรู้ ทำให้คณะผู้จัดทำสามารถนำความรู้ที่ได้มาประยุกต์ใช้ในการดำเนินโครงการ รวมทั้งให้คำปรึกษาและคำแนะนำที่เป็นประโยชน์ ทำให้คณะผู้จัดทำสามารถแก้ไขโครงการได้สมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณมหาวิทยาลัยศรีนครินทรวิโรฒที่เป็นแหล่งศึกษาหาความรู้ และเอื้อเพื่อสถานที่ สำหรับการดำเนินโครงการ รวมทั้งให้ประสบการณ์ในการทำกิจกรรมต่าง ๆ เพื่อที่จะนำไปประยุกต์ใช้ในการทำงานและการดำเนินชีวิตต่อไป

ขอขอบพระคุณบิดา มารดา เพื่อน รวมถึงครอบครัวของคณะผู้จัดทำที่ได้ให้การช่วยเหลือและสนับสนุนมาโดยตลอด รวมถึงให้ความห่วงใยและเป็นกำลังใจสำคัญในการทำโครงการเสมอมา

สุดท้ายขอขอบพระคุณผู้มีส่วนเกี่ยวข้องทุก ๆ ท่านที่ทำให้โครงการนี้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำหวังว่าโครงการการใช้เทคนิคประมวลผลภาพแบบไม่มีผู้สอนเพื่อตรวจจับการเจริญเติบโตที่ผิดปกติของต้นไม้ (Using unsupervised image processing techniques to detect anomaly growth of plant) จะเป็นประโยชน์ต่อผู้ที่สนใจศึกษา และสามารถเป็นส่วนหนึ่งสำหรับผู้ที่สนใจจะนำไปต่อยอดในบริบทต่าง ๆ ต่อไป

คณะผู้จัดทำ

สารบัญ

บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
บทที่ 1	1
บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2	3
องค์ความรู้ที่ใช้ในการทำวิจัย	3
2.1 องค์ความรู้เกี่ยวกับ Image Processing	3
2.2 องค์ความรู้เกี่ยวกับ Background Subtraction	4
2.3 องค์ความรู้เกี่ยวกับ Connecting component labeling	4
2.4 องค์ความรู้เกี่ยวกับ Unsupervised Learning	5
2.4.1 Clustering	5
2.5 องค์ความรู้เกี่ยวกับ Outlier Detection	6
2.6 องค์ความรู้เกี่ยวกับ Euclidean distance	6
บทที่ 3	7
การดำเนินงานวิจัย	7
3.1 วิธีการดำเนินงาน	7

3.1.1 การวางแผนระบบ	7
3.1.2 การวิเคราะห์และออกแบบระบบ	7
3.1.3 การปรับใช้ระบบจริง	7
3.1.4 การประเมินผล	7
3.2 แผนการดำเนินงาน	8
3.3 อุปกรณ์และเครื่องมือที่ใช้พัฒนา	9
3.3.1 Hardware	9
3.3.2 Software	9
3.3.3 Language	10
3.4 ภาพรวมการทำงานของระบบ	10
3.5 ขั้นตอนการทำงานของระบบ	11
3.5.1 ส่วนของการเตรียมการ	11
3.5.2 ส่วนของการทำงานของระบบ	11
บทที่ 4	19
ผลการดำเนินงาน	19
4.1 ผลลัพธ์จากการเก็บข้อมูล	19
4.1.1 ผลลัพธ์ที่ได้จากกระบวนการ Background Subtraction	19
4.1.2 ผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุ	21
4.1.3 ผลลัพธ์ที่ได้จากการนับจำนวนใบ	26
4.1.4 ข้อมูลที่เก็บใน csv ไฟล์	32
4.2 ผลลัพธ์ที่ได้จากการทดสอบ	34
บทที่ 5	37
สรุปผล อภิปรายผล และข้อเสนอแนะ	37

5.1 สรุปผลการดำเนินการ	37
5.2 ปัญหาและอุปสรรคจากการดำเนินงาน	38
5.3 ข้อเสนอแนะ	38
เอกสารอ้างอิง	39
ภาคผนวก	41
ภาคผนวก ก : การติดตั้งโปรแกรมที่ใช้พัฒนา	42
Notepad++	42
ภาคผนวก ข : การ Setup/Configure	45
การติดตั้ง OpenCV	45

สารบัญรูปภาพ

รูปที่ 2.1 Connecting component labeling	4
รูปที่ 2.2 กระบวนการ clustering	5
รูปที่ 2.6 Outlier Detection	6
รูปที่ 2.7 Euclidean distance	6
รูปที่ 3.1 Notepad++ Logo	9
รูปที่ 3.2 Google Colab Logo	10
รูปที่ 3.3 รูปแสดงกระบวนการทำ Background Subtraction	12
รูปที่ 3.4 รูปแสดงกระบวนการแปลงภาพเป็น grayscale	13
รูปที่ 3.5 รูปแสดงกระบวนการทำ Canny edge detection และ dilation + erosion	13
รูปที่ 3.6 รูปแสดงกระบวนการค้นหา contour	13
รูปที่ 3.7 รูปแสดงกระบวนการวนลูปเพื่อคำนวณขนาดของวัตถุ	14
รูปที่ 3.8 รูปแสดงกระบวนการ import library ที่เกี่ยวข้อง	15
รูปที่ 3.9 รูปแสดงกระบวนการทำ image filtering	15
รูปที่ 3.10 รูปแสดงกระบวนการแปลงภาพเป็น grayscale และ apply Otsu's thresholding	15
รูปที่ 3.11 รูปแสดงกระบวนการคำนวณ Euclidean distance และค้นหาจุด peak	15
รูปที่ 3.12 รูปแสดงกระบวนการ apply Watershed algorithm	15
รูปที่ 3.13 รูปแสดงกระบวนการเพื่อนับจำนวนวัตถุ	16
รูปที่ 3.13 รูปแสดงกระบวนการคำนวณหาจุด centroid	17
รูปที่ 3.14 รูปแสดงกระบวนการคำนวณ distance ระหว่างจุด centroid เทียบกับจุดอื่นๆ	17
รูปที่ 3.14 รูปแสดงผลลัพธ์จากกระบวนการ outlier detection	18

สารบัญตาราง

ตารางที่ 3.1 ตารางดำเนินการ	8
ตารางที่ 3.2 ตารางแสดงการทำ 3D Scatter plot	16
ตารางที่4.1 ตารางแสดงผลลัพธ์จากกระบวนการ Background Subtraction	20
ตารางที่4.2 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่1	21
ตารางที่4.3 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่2	21
ตารางที่4.4 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่3	22
ตารางที่4.5 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่4	22
ตารางที่4.6 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่5	23
ตารางที่4.7 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่6	23
ตารางที่4.8 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่7	24
ตารางที่4.9 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่8	24
ตารางที่4.10 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่9	25
ตารางที่4.11 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่10	25
ตารางที่4.12 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่1	26
ตารางที่4.13 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่2	27
ตารางที่4.14 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่3	27
ตารางที่4.15 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่4	28
ตารางที่4.16 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่5	28
ตารางที่4.17 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่6	29
ตารางที่4.17 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่7	29
ตารางที่4.18 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่8	30
ตารางที่4.19 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่9	30
ตารางที่4.20 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่10	31
ตารางที่4.21 ตารางแสดงข้อมูลใน csv ไฟล์	34
ตารางที่4.22 ตารางแสดงผลลัพธ์ที่ได้จากกระบวนการ Outlier Detection	36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ประเทศไทยเป็นประเทศแห่งการเกษตร ได้รับการยอมรับถึงคุณภาพที่ดีของผลผลิต เนื่องด้วยปัจจัยความเสี่ยงในการเจริญเติบโตของพืชทำให้เกษตรกรในไทยมีภาระมากขึ้น ทั้งด้านแมลงศัตรูพืช โรคต่าง ๆ ที่มีตามฤดูกาล แม้กระทั่งสภาพอากาศของธรรมชาติ ทำให้เกษตรกรต้องเสียรายได้ และทำให้ผลผลิตมีคุณภาพที่ไม่ดีพอต่อการจัดจำหน่าย

การศึกษามลการเจริญเติบโตของพืช ได้มีการศึกษาและทดลองเพื่อหาวิธีการที่ดีที่สุดในการปลูกพืช ซึ่งมีงานทดลองและวิจัยเกี่ยวข้องกับงานด้านนี้เป็นจำนวนมาก ซึ่ง ส่วนใหญ่เป็นการใช้อองค์ความรู้เกี่ยวกับ Supervised Learning ในการทดลองจำเป็นต้องมีฐานข้อมูล เพื่อนำมาเปรียบเทียบกับภาพที่นำมาใช้ในการประมวลผล เช่น ต้องการจะตรวจสอบโรคของต้นมะม่วง โดยใช้ใบของต้นมะม่วงที่เป็นโรคมাত্রตรวจสอบ จำเป็นที่จะต้องมีฐานข้อมูลของใบต้นมะม่วงที่เป็นโรคอยู่แล้วเพื่อใช้ในการตรวจสอบชนิดของโรคที่จะเกิดขึ้นได้ ทำให้ผู้วิจัยต้องทำการเก็บข้อมูลเพื่อทำฐานข้อมูลให้ครอบคลุมกับโรคที่มีอยู่ ซึ่งจะต้องใช้เวลามากเพื่อให้ได้ฐานข้อมูลที่มีคุณภาพ และสามารถตรวจสอบได้เพียงชนิดของโรค และชนิดของต้นไม้ที่มีในฐานข้อมูลเท่านั้น แต่ในงานวิจัยนี้ผู้จัดทำได้เล็งเห็นถึงข้อจำกัดของงานวิจัยเดิม ๆ ที่ทำมา ทำให้ผู้จัดทำนำองค์ความรู้เกี่ยวกับ Unsupervised Learning มาใช้ในงานวิจัย เพื่อลดปัญหาและข้อจำกัดที่เกิดขึ้นเมื่อใช้อองค์ประกอบ Supervised Learning โดยข้อดีของ Unsupervised Learning จะทำให้งานวิจัยสามารถดำเนินงานได้โดยไม่จำเป็นต้องมีฐานข้อมูลอยู่ก่อน เพื่อลดปัญหาระยะเวลาในการจัดเก็บข้อมูล และสามารถใช้ได้กับพืชได้หลากหลายชนิด

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อศึกษาการเขียนโปรแกรมเพื่อวิเคราะห์การเจริญเติบโตของพืช
2. เพื่อศึกษาเทคนิคและองค์ความรู้ที่เกี่ยวข้องในการวิเคราะห์การเจริญเติบโตของพืช
3. เพื่อศึกษาหลักการวิเคราะห์ภาพถ่ายโดยใช้ image processing
4. เพื่อศึกษาการใช้ feature ในการทำการแบ่งกลุ่ม
5. เพื่อศึกษาหลักการตรวจจับต้นไม้ที่มีความผิดปกติโดยใช้หลักการ Outlier Detection
6. เพื่อลดอัตราการตายของต้นไม้ โดยแจ้งเตือนผู้ใช้หากต้นไม้มีการเจริญเติบโตแบบผิดปกติ

1.3 ขอบเขตของโครงการ

1. สามารถตรวจสอบการเจริญเติบโตของต้นไม้ได้อย่างแม่นยำ
2. ใช้หลักการ image processing ในการวิเคราะห์การเจริญเติบโตของต้นไม้
3. ใช้หลักการ Unsupervised Learning (Outlier) ในการตรวจต้นไม้ที่มีการเจริญเติบโตผิดปกติ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถนำไปใช้กับการเกษตรได้อย่างมีประสิทธิภาพ
2. วิเคราะห์การเจริญเติบโตของพืชได้อย่างมีประสิทธิภาพ
3. ตรวจสอบการเจริญเติบโตของพืชได้อย่างแม่นยำ
4. สามารถตรวจจับต้นไม้ที่เจริญเติบโตแบบผิดปกติได้
5. ลดอัตราการตายของพืช

บทที่ 2

องค์ความรู้ที่ใช้ในการทำวิจัย

2.1 องค์ความรู้เกี่ยวกับ Image Processing

2.1.1 ความหมายของ image processing

การประมวลผลภาพ (Image Processing) หมายถึง การนำภาพมาประมวลผลหรือคิดคำนวณด้วยคอมพิวเตอร์ เพื่อให้ได้ข้อมูลที่เราต้องการทั้งในเชิงคุณภาพและปริมาณ โดยมีขั้นตอนต่าง ๆ ที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่เราสนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ

2.1.2 การทำงานของ image processing

จะใช้กล้องในการถ่ายภาพการเจริญเติบโตของพืชที่ระยะระยะ แล้วนำภาพที่ได้มาวิเคราะห์หา feature ของพืช เช่น ความสูง / สีของใบ / ความกว้างของลำต้น เป็นต้น จากนั้นจะนำ feature ที่ได้มาใช้ในการกระบวนการถัดไป

2.1.3 ประโยชน์ของ image processing

จะเห็นได้ว่าระบบเหล่านี้จำเป็นต้องมีการประมวลผลภาพจำนวนมาก และเป็นกระบวนการที่ต้องทำซ้ำ ๆ กันในรูปแบบเดิมเป็นส่วนใหญ่ ซึ่งงานในลักษณะเหล่านี้ หากให้มนุษย์วิเคราะห์เอง มักต้องใช้เวลามากและใช้แรงงานสูง อีกทั้งหากจำเป็นต้องวิเคราะห์ภาพเป็นจำนวนมาก ผู้วิเคราะห์ภาพเองอาจเกิดอาการล้า ส่งผลให้เกิดความผิดพลาดขึ้นได้ ดังนั้นคอมพิวเตอร์จึงมีบทบาทสำคัญในการทำหน้าที่เหล่านี้แทนมนุษย์ อีกทั้ง เป็นที่ทราบโดยทั่วกันว่าคอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมากมหาศาลได้ในเวลาอันสั้น จึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบต่าง ๆ ดังกล่าวข้างต้น

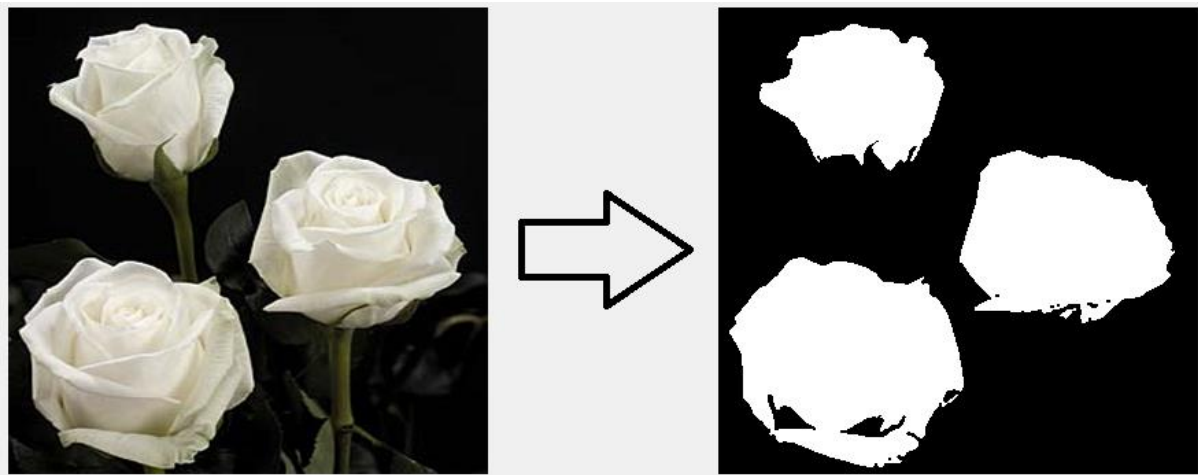
2.2 องค์ความรู้เกี่ยวกับ Background Subtraction

เป็นเทคนิคที่ใช้ในการหาความต่างของเฟรม foreground กับเฟรม background โดยประโยชน์ของการทำ background subtraction นั้นจะสามารถบอกได้ว่ามีวัตถุอะไรที่กำลังเคลื่อนไหวอยู่ในเฟรม foreground

$$|\text{foreground}(x,y) - \text{background}(x,y)|$$

2.3 องค์ความรู้เกี่ยวกับ Connecting component labeling

ใช้ในการวิเคราะห์ส่วนประกอบหรือบริเวณที่มีการเชื่อมติดกันและใช้ในการแยกบริเวณต่าง ๆ ออกจากกัน โดยทำการหา bounding box ของแต่ละ component ที่ดึงออกมาได้



รูปที่ 2.1 Connecting component labeling 1

2.4 องค์ความรู้เกี่ยวกับ Unsupervised Learning

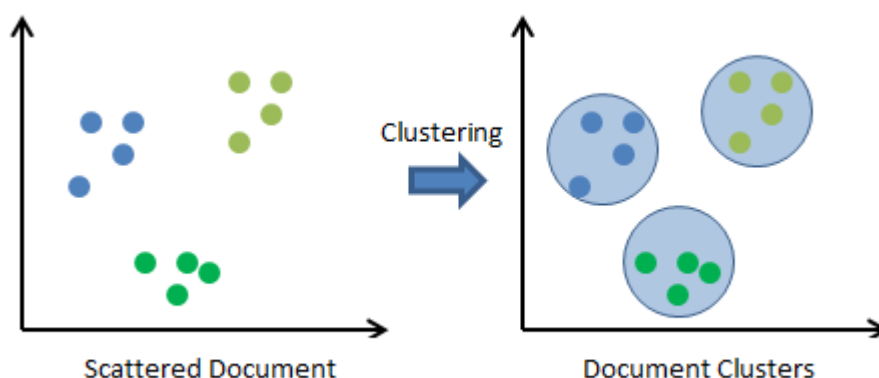
Unsupervised Learning หรือ การเรียนรู้แบบไม่มีผู้สอน ในการเรียนรู้แบบนี้ เราจะมีแต่ Inputs ให้ แล้วบอก Machine ว่าต้องการอะไร (เช่นการแบ่งกลุ่ม)

โดยงานวิจัยนี้จะใช้ Feature จากขั้นตอน Image Processing มาเป็นค่าอินพุต เพื่อต้องการที่จะใช้ feature ในการแบ่งกลุ่มข้อมูลด้วยวิธี clustering

2.4.1 Clustering

Clustering เป็นวิธีการที่พิจารณาข้อมูลแต่ละแถวเสมือนเป็นวัตถุ จะมีหลักการเหมือนกับการจำแนกประเภทข้อมูล คือทำการแบ่งข้อมูลออกเป็นคลัสเตอร์ โดยจะจัดให้ข้อมูลที่มีความคล้ายคลึงกันอยู่ในคลัสเตอร์เดียวกัน และข้อมูลที่อยู่ต่างคลัสเตอร์กันจะมีความคล้ายคลึงกันน้อยสุด ซึ่งความเหมือนหรือต่างกันสามารถเปรียบเทียบได้กับความใกล้ชิดกันของวัตถุใด ๆ โดยใช้ระยะทางเป็นตัวชี้วัด

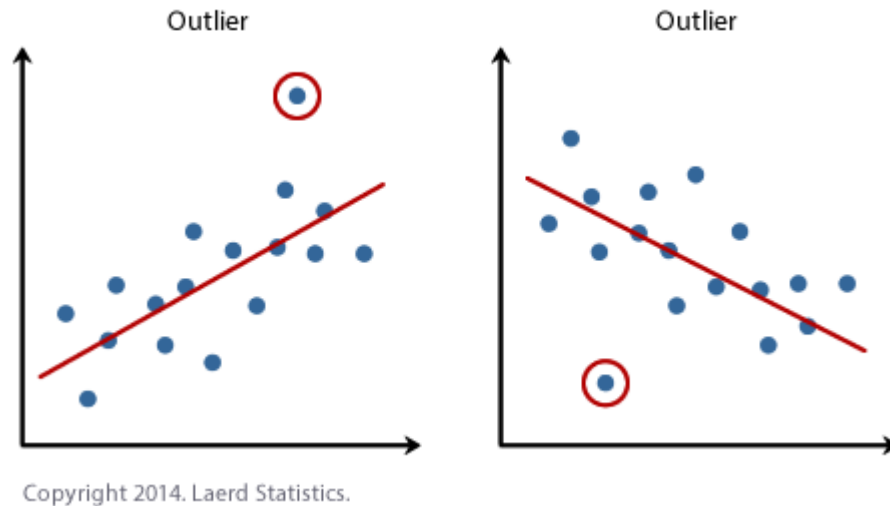
โดยจะเริ่มจากการพิจารณาข้อมูล feature ของพีช และทำการแบ่งข้อมูลโดยใช้ feature ออกเป็นคลัสเตอร์ คือ กลุ่มที่ต้นไม่มีการเจริญเติบโตแบบปกติ



รูปที่ 2.2 กระบวนการ clustering 2

2.5 องค์ความรู้เกี่ยวกับ Outlier Detection

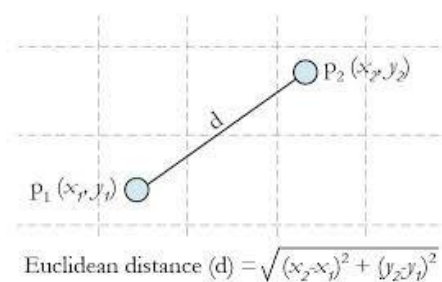
ค่าผิดปกติ (Outliers) เป็นข้อมูลที่มีค่าแยกออกจากกลุ่มหรือผิดแผกแตกต่างไปจากข้อมูลค่าอื่น ๆ เพื่อจะใช้เป็นตัวตรวจสอบหาต้นไม้มันที่มีการเจริญเติบโตแบบผิดปกติ



รูปที่ 2.6 Outlier Detection 3

2.6 องค์ความรู้เกี่ยวกับ Euclidean distance

ระยะทางแบบยูคลิดระหว่างจุดสองจุด p และ q คือความยาวของส่วนของเส้นตรง pq ถ้า $p = (p_1, p_2, \dots, p_n)$ และ $q = (q_1, q_2, \dots, q_n)$ ในระบบพิกัดคาร์ทีเซียน เป็นจุดสองจุดบนปริภูมิยูคลิด n มิติ ระยะทางระหว่างจุด p กับ q คำนวณได้จาก



รูปที่ 2.7 Euclidean distance 4

บทที่ 3

การดำเนินงานวิจัย

3.1 วิธีการดำเนินงาน

3.1.1 การวางแผนระบบ

- ศึกษาหลักการทฤษฎี องค์ความรู้ที่เกี่ยวข้อง และความต้องการของระบบ
- กำหนดวัตถุประสงค์ของโครงการโดยรวม

3.1.2 การวิเคราะห์และออกแบบระบบ

- วิเคราะห์และรวบรวมข้อมูลเพื่อกำหนดขอบเขต และข้อจำกัดของการทดลอง
- ออกแบบการทำงานของระบบ
- ออกแบบวิธีการทดสอบประสิทธิภาพของระบบ

3.1.3 การปรับใช้ระบบจริง

- ทำการเก็บข้อมูลภาพ จากกล้องถ่ายภาพ
- นำข้อมูลที่เก็บได้มาวิเคราะห์
- นำข้อมูลที่วิเคราะห์มาแสดงผล

3.1.4 การประเมินผล

- ทดสอบระบบที่พัฒนา
- ปรับปรุงและแก้ไขข้อผิดพลาดหลังการทดสอบ
- ทำการเปรียบเทียบและประเมินผล
- สรุปขั้นตอนการดำเนินการวิจัยทั้งหมด และจัดทำคู่มือการใช้งาน

3.2 แผนการดำเนินงาน

ขั้นตอน	เดือน									
	ปี 2561					ปี 2562				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1.การวางแผนโครงการ										
ประชุมเพื่อเลือกหัวข้อ										
ศึกษาหลักการและทฤษฎีที่เกี่ยวข้องกับงานวิจัย										
ศึกษาและออกแบบการพัฒนาระบบ										
ศึกษาเทคโนโลยีที่จะนำมาใช้										
2.การวิเคราะห์ระบบ										
วิเคราะห์ความเป็นไปได้ในการทำระบบ										
3. การออกแบบระบบ										
กำหนดวัตถุประสงค์และขอบเขตของระบบ										
4.การพัฒนาและทดสอบระบบ										
ทำการสร้างและพัฒนาระบบ										
ทำการทดสอบและปรับปรุงระบบ										
5.สรุปและประเมินผล										
ทำการปรับปรุงระบบอีกครั้ง										
ประเมินผลการใช้งานระบบ										
สรุปผลการดำเนินงาน										
จัดทำคู่มือ และเอกสารที่เกี่ยวข้อง										
นำเสนอระบบที่สมบูรณ์										

ตารางที่ 3.1 ตารางดำเนินการ 1

3.3 อุปกรณ์และเครื่องมือที่ใช้พัฒนา

3.3.1 Hardware

- Computer Notebook
 - Intel Core i7-6500U CPU @ 2.50 GHz 2.59 GHz
 - Windows 10 Pro (64-bit Operating System)
 - RAM 8 GB
- Wireless IP Camera

3.3.2 Software

- Notepad ++

เป็นโปรแกรมสำหรับการเปิด สร้าง และแก้ไข source code สำหรับนักพัฒนา โปรแกรมมีความสามารถสูง ทำงานได้รวดเร็ว ภายใต้การใช้งานทรัพยากรเครื่องต่ำ และไฟล์ติดตั้งที่มีขนาดเล็ก



รูปที่ 3.1 Notepad++ Logo 5

- Google Colab

Jupyter notebook ดัดแปลงที่รันอยู่บนคลาวด์ และไม่จำเป็นต้องติดตั้งโปรแกรมใด ๆ ก่อนใช้งาน ซึ่งทุกคนสามารถใช้งานได้ เพียงแค่มีบัญชี Google Drive เพื่อใช้ในการจัดเก็บตัวโค้ดเท่านั้น โดยภาษา Python เป็นภาษาหลักที่ใช้ในการเขียนและรันงานบนเว็บ Colab นี้



รูปที่ 3.2 Google Colab Logo 6

3.3.3 Language

- Python Programming Language

3.4 ภาพรวมการทำงานของระบบ

โครงการนี้ทำการศึกษาการเจริญเติบโตของต้นไม้โดยใช้เทคนิคการประมวลผลภาพเพื่อดึง feature เฉพาะของต้นไม้ออกมา และเก็บข้อมูล feature เฉพาะนี้ไว้ในไฟล์ .csv จากนั้นใช้เทคนิคประมวลผลภาพแบบไม่มีผู้สอน Clustering เพื่อแบ่งกลุ่มของข้อมูลโดยใช้ feature จากขั้นตอนการประมวลผลภาพในโครงการนี้ใช้ 3 feature คือ ความกว้างของลำต้น ความสูงของลำต้น และจำนวนใบ เท่านั้น จากนั้นใช้หลักการ Outlier เพื่อตรวจจับการเจริญเติบโตที่ผิดปกติของต้นไม้

3.5 ขั้นตอนการทำงานของระบบ

3.5.1 ส่วนของการเตรียมการ

ทำการติดตั้งกล้องจำนวน 1 ตัว เพื่อสังเกตดูการเจริญเติบโตของพืช โดยในโครงการนี้ใช้ต้นผักบุ้ง จำนวน 6 ต้น เป็นพืชทดสอบ

3.5.2 ส่วนของการทำงานของระบบ

แบ่งการทำงานออกเป็น 3 ส่วน ได้ดังนี้

1. ส่วนของการตรวจจับวัตถุจากวีดีโอ
2. ส่วนของ feature extraction ในโครงการนี้สนใจเพียง 3 feature คือ ความกว้าง ความสูง และ จำนวนใบ
3. ส่วนของการตรวจจับต้นไม้ที่มีการเจริญเติบโตแบบผิดปกติ

1. ส่วนของการตรวจจับวัตถุจากวิดีโอ

อ่านไฟล์วิดีโอเข้ามาแล้วทำ Background Subtraction เพื่อดึงต้นไม้ออกจากพื้นหลัง โดยใช้ฟังก์ชัน `cv2.createBackgroundSubtractorMOG()` จากนั้นจึงทำการแคปภาพที่ได้จากการทำ Background Subtraction ไว้ที่โฟลเดอร์ image

```
import cv2
import argparse

ap = argparse.ArgumentParser()
ap.add_argument("-v", "--video",
    help = "path to the (optional) video file")
args = vars(ap.parse_args())

if not args.get("video", False):
    cap = cv2.VideoCapture(0)
else:
    cap = cv2.VideoCapture(args["video"])
    fgbg = cv2.bgsegm.createBackgroundSubtractorMOG()

while True:
    ret, frame = cap.read()

    fgmask = fgbg.apply(frame)
    cv2.imshow('frame', fgmask)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

รูปที่ 3.3 รูปแสดงกระบวนการทำ Background Subtraction ๗

2. ส่วนของ feature extraction แบ่งได้เป็น 2 ส่วน ดังนี้

2.1 Feature ความกว้าง และ ความสูง

2.1.1 โหลดภาพ และ convert to grayscale และทำการ blur ภาพเล็กน้อย

```
# Load the image, convert it to grayscale, และเบลอเล็กน้อย
image = cv2.imread('BI_1_.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (7, 7), 0)
```

รูปที่ 3.4 รูปแสดงกระบวนการแปลงภาพเป็น grayscale 8

2.1.2 ทำการตรวจจับ edge detection, จากนั้นทำ dilation + erosion เพื่อปิดช่องว่างระหว่าง edge

```
# ทำการตรวจจับ edge detection, จากนั้นทำ dilation + erosion เพื่อปิดช่องว่างระหว่าง edge
edged = cv2.Canny(gray, 50, 100)
edged = cv2.dilate(edged, None, iterations=1)
edged = cv2.erode(edged, None, iterations=1)
```

รูปที่ 3.5 รูปแสดงกระบวนการทำ Canny edge detection และ dilation + erosion 9

2.1.3 หา contours ใน edge map

```
: # หา contours ใน edge map
cnts = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)

# sort the contours from left-to-right
(cnts, _) = contours.sort_contours(cnts)
pixelsPerMetric = None
```

รูปที่ 3.6 รูปแสดงกระบวนการค้นหา contour 10

2.1.4 วนลูปรอบ contours ที่ละอัน

```
def midpoint(ptA, ptB):
    return ((ptA[0] + ptB[0]) * 0.5, (ptA[1] + ptB[1]) * 0.5)

# วนลูปรอบ contours ทีละอัน
for c in cnts:
    # ไม่สนใจวัตถุที่มีขนาดเล็กกว่า 100
    if cv2.contourArea(c) < 400:
        continue

    # compute the rotated bounding box
    orig = image.copy()
    box = cv2.minAreaRect(c)
    box = cv2.cv.BoxPoints(box) if imutils.is_cv2() else cv2.boxPoints(box)
    box = np.array(box, dtype="int")

    # วาด box
    box = perspective.order_points(box)
    cv2.drawContours(orig, [box.astype("int")], -1, (132, 0, 255), 2)

    # คำนวณจุดกึ่งกลางระหว่าง จุดบนซ้าย กับ บนขวา และ ล่างซ้าย กับ ล่างขวา
    (tl, tr, br, bl) = box
    (tltrX, tltrY) = midpoint(tl, tr)
    (blbrX, blbrY) = midpoint(bl, br)

    # คำนวณจุดกึ่งกลางระหว่าง จุดบนซ้าย กับ ล่างซ้าย และ บนขวา กับ ล่างขวา
    (tlblX, tlblY) = midpoint(tl, bl)
    (trbrX, trbrY) = midpoint(tr, br)

    # วาดเส้นเชื่อมระหว่างจุด
    cv2.line(orig, (int(tltrX), int(tltrY)), (int(blbrX), int(blbrY)), (255, 26, 26), 2)
    cv2.line(orig, (int(tlblX), int(tlblY)), (int(trbrX), int(trbrY)), (255, 26, 26), 2)

    # คำนวณ Euclidean distance ระหว่างจุดกึ่งกลาง
    dA = dist.euclidean((tltrX, tltrY), (blbrX, blbrY))
    dB = dist.euclidean((tlblX, tlblY), (trbrX, trbrY))

    # if the pixels per metric has not been initialized, then compute it as the ratio of pixels to supplied metric
    if pixelsPerMetric is None:
        pixelsPerMetric = dB / 0.955

    # คำนวณขนาดของวัตถุ
    dimA = dA / pixelsPerMetric
    dimB = dB / pixelsPerMetric

    # เขียนขนาดที่วัดได้เป็นตัวเลข
    cv2.putText(orig, "{:.1f}in".format(dimA), (int(tltrX - 15), int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 128, 213), 2)
    cv2.putText(orig, "{:.1f}in".format(dimB), (int(trbrX + 10), int(trbrY)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 128, 213), 2)

    # show the output image
    cv2.imshow("Image", orig)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

รูปที่ 3.7 รูปแสดงกระบวนการวนลูปรอบเพื่อคำนวณขนาดของวัตถุ 11

2.1.5 เก็บค่าที่ได้ไว้ที่ไฟล์ plant.csv

2.2 Feature นับจำนวนใบ

2.2.1 Import library ที่เกี่ยวข้อง

```
from scipy.spatial import distance as dist
from imutils import perspective
from imutils import contours
from skimage.feature import peak_local_max
from skimage.morphology import watershed
from scipy import ndimage
import numpy as np
import argparse
import imutils
import cv2
```

รูปที่ 3.8 รูปแสดงกระบวนการ import library ที่เกี่ยวข้อง 12

2.2.2 โหลดภาพและ ทำ pyramid mean shift filtering เพื่อช่วยในขั้นตอน thresholding

```
image = cv2.imread('img/1_1.jpg')
shifted = cv2.pyrMeanShiftFiltering(image, 21, 51)
```

รูปที่ 3.9 รูปแสดงกระบวนการทำ image filtering 13

2.2.3 แปลงภาพเป็น grayscale และ apply Otsu's thresholding

```
gray = cv2.cvtColor(shifted, cv2.COLOR_BGR2GRAY)
thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]
```

รูปที่ 3.10 รูปแสดงกระบวนการแปลงภาพเป็น grayscale และ apply Otsu's thresholding 14

2.2.4 คำนวณ Euclidean distance จากทุก pixel ถึง pixel ที่ใกล้ที่สุด จากนั้นหาจุดสูงสุดใน distance map (จุด peak)

```
D = ndimage.distance_transform_edt(thresh)
localMax = peak_local_max(D, indices=False, min_distance=44, labels=thresh)
```

รูปที่ 3.11 รูปแสดงกระบวนการคำนวณ Euclidean distance และค้นหาจุด peak 15

2.2.5 ทำการวิเคราะห์ connected component กับ local peaks โดยใช้ 8-connectivity จากนั้นจึง apply Watershed algorithm

```
markers = ndimage.label(localMax, structure=np.ones((3, 3)))[0]
labels = watershed(-D, markers, mask=thresh)
print("{} leaves found".format(len(np.unique(labels)) - 1))
```

รูปที่ 3.12 รูปแสดงกระบวนการ apply Watershed algorithm 16

2.2.6 วนลูปรอบ unique labels แล้ว returned โดยใช้ Watershed algorithm

```
for label in np.unique(labels):
    # ถ้า label = 0 จะสรุปได้ว่าเป็นพื้นหลัง จะ ignore
    if label == 0:
        continue

    # otherwise, allocate memory for the label region and draw it on the mask
    mask = np.zeros(gray.shape, dtype="uint8")
    mask[labels == label] = 255

    # detect contours in the mask และ เลือกอันที่ใหญ่สุด
    cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[-2]
    c = max(cnts, key=cv2.contourArea)

    # draw a circle enclosing the object
    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 1)
    cv2.putText(image, "{}".format(label), (int(x) - 10, int(y)), cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)

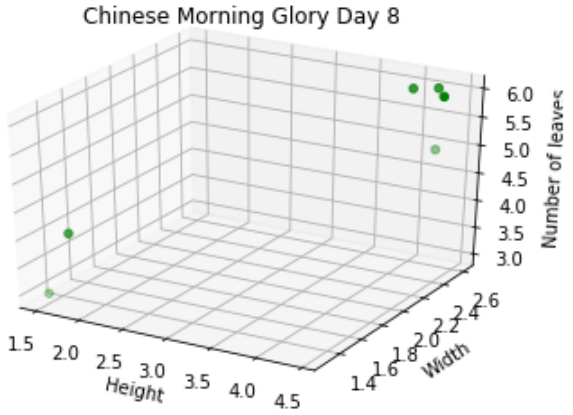
# show the output image
cv2.imshow("Output", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

รูปที่ 3.13 รูปแสดงกระบวนการเพื่อนับจำนวนวัตถุ 17

2.2.7 เก็บค่าที่ได้ไว้ที่ไฟล์ plant.csv

3. ส่วนของการตรวจจับต้นไม้มที่มีการเจริญเติบโตแบบผิดปกติโดยใช้หลักการ Outlier Detection

3.1 นำข้อมูลจากไฟล์ plant.csv มา plot ในรูปแบบ 3D Scatter plot

Python Code	Result
<pre>fig = plt.figure() ax = fig.add_subplot(111, projection='3d') x = day8['Height'] y = day8['Width'] z = day8['Leaf'] ax.scatter(x, y, z, c='g', marker='o') ax.set_title('Chinese Morning Glory Day 8') ax.set_xlabel('Height') ax.set_ylabel('Width') ax.set_zlabel('Number of leaves') plt.show()</pre>	

ตารางที่ 3.2 ตารางแสดงการทำ 3D Scatter plot 2

3.2 คำนวณ centroid

```
#centroid
length = y.shape[0]
sum_x = np.sum(y[:, 0])
sum_y = np.sum(y[:, 1])
sum_z = np.sum(y[:, 2])
a=sum_x/length
b=sum_y/length
c=sum_z/length
d=[]
d.extend([a,b,c])
print(d)

[2.767, 1.89, 5.833333333333333]
```

รูปที่ 3.13 รูปแสดงกระบวนการคำนวณหาจุด centroid 18

3.3 คำนวณหา distance ระหว่างจุด centroid เทียบกับจุดอื่น ๆ แล้วเลือก 2 จุดที่อยู่ใกล้เคียง centroid มากที่สุด

```
import math

centroid_1=d
dis_arr=[]

for item in y:
    distance = math.sqrt(((d[0]-item[0])**2)+((d[1]-item[1])**2)+((d[2]-item[2])**2))
    print('%.2f'%distance)
    dis_arr.append(float('%.2f'%distance))
#print(dis_arr)

2.40
1.53
1.50
2.62
3.33
0.71
```

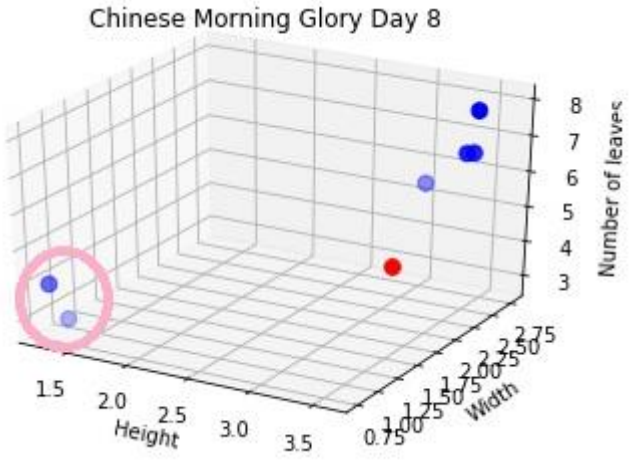
รูปที่ 3.14 รูปแสดงกระบวนการคำนวณ distance ระหว่างจุด centroid เทียบกับจุดอื่น ๆ 19

3.4 คำนวณ centroid ใหม่ระหว่างจุด 2 จุด จากขั้นตอนที่แล้ว

3.5 กำหนด threshold = 2.5

3.6 วนทำซ้ำตั้งแต่ขั้นตอนที่ 3.2 ใหม่

3.7 หากจุดใดมีระยะห่างจากจุดอื่นเกินค่า threshold ที่กำหนดจะบอกได้ว่าเป็น Outlier หรือต้นไม่มีการเจริญเติบโตที่ผิดปกติ

Python Code	Result
<pre>fig = plt.figure() ax = fig.add_subplot(111, projection='3d') x = day9['Height'] y = day9['Width'] z = day9['Leaf'] ax.scatter(x, y, z, s=60, c='b', marker='o') ax.scatter(d[0], d[0], d[0], s=60, c='r', marker='o') ax.set_title('Chinese Morning Glory Day 8') ax.set_xlabel('Height') ax.set_ylabel('Width') ax.set_zlabel('Number of leaves') plt.show()</pre>	

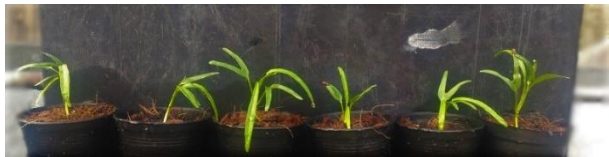
รูปที่ 3.14 รูปแสดงผลลัพธ์จากกระบวนการ outlier detection 20

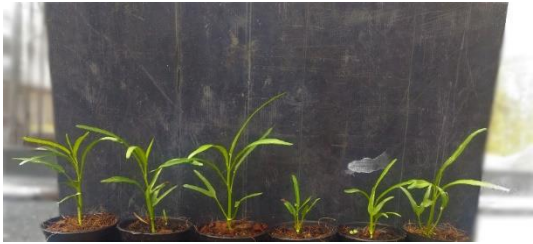
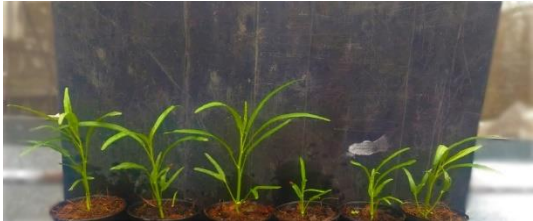
บทที่ 4

ผลการดำเนินงาน

4.1 ผลลัพธ์จากการเก็บข้อมูล

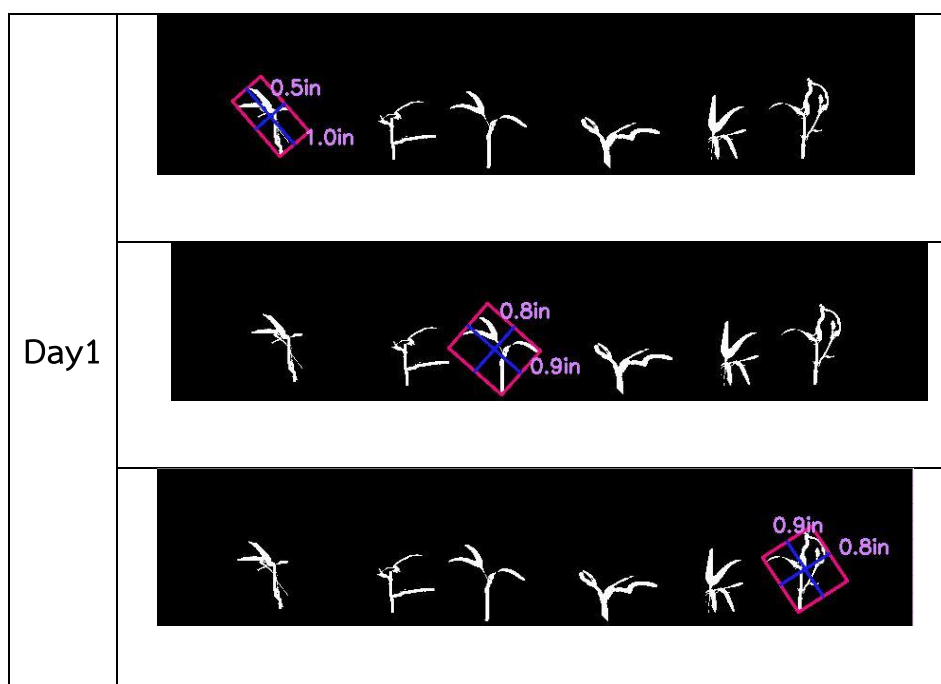
4.1.1 ผลลัพธ์ที่ได้จากกระบวนการ Background Subtraction

Day	Original	Background Subtraction
1		
2		
3		
4		
5		

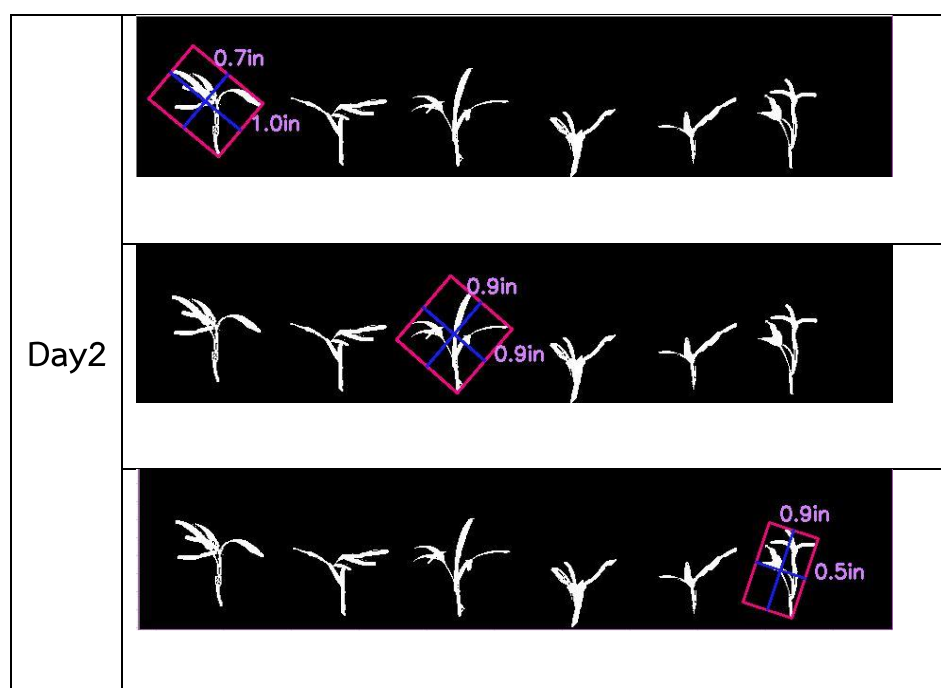
Day	Original	Background Subtraction
6		
7		
8		
9		
10		

ตารางที่ 4.1 ตารางแสดงผลลัพธ์จากกระบวนการ Background Subtraction 3

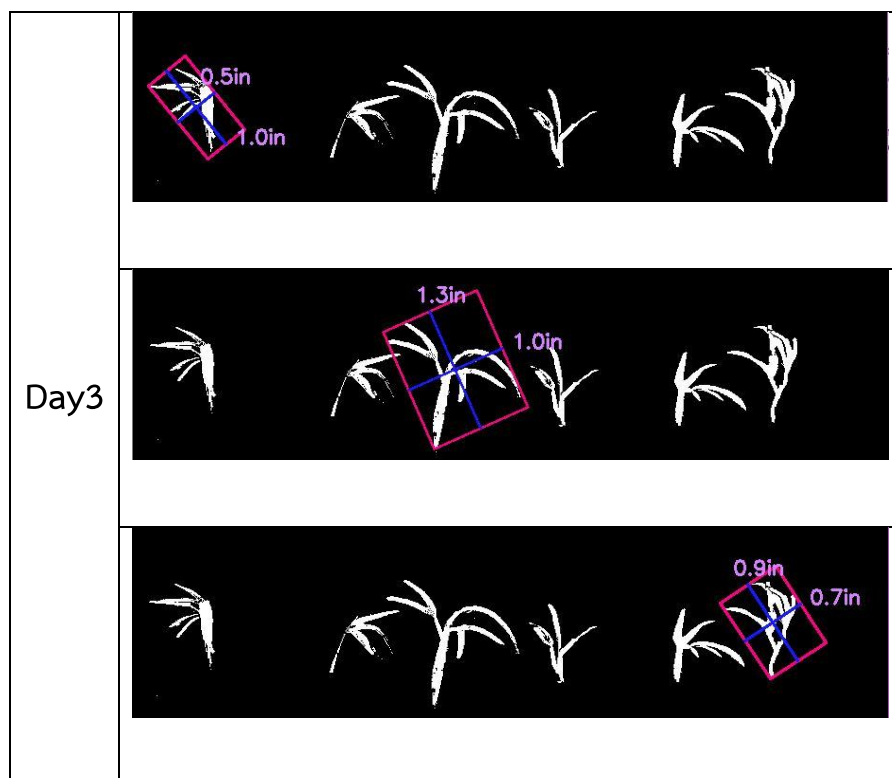
4.1.2 ผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุ



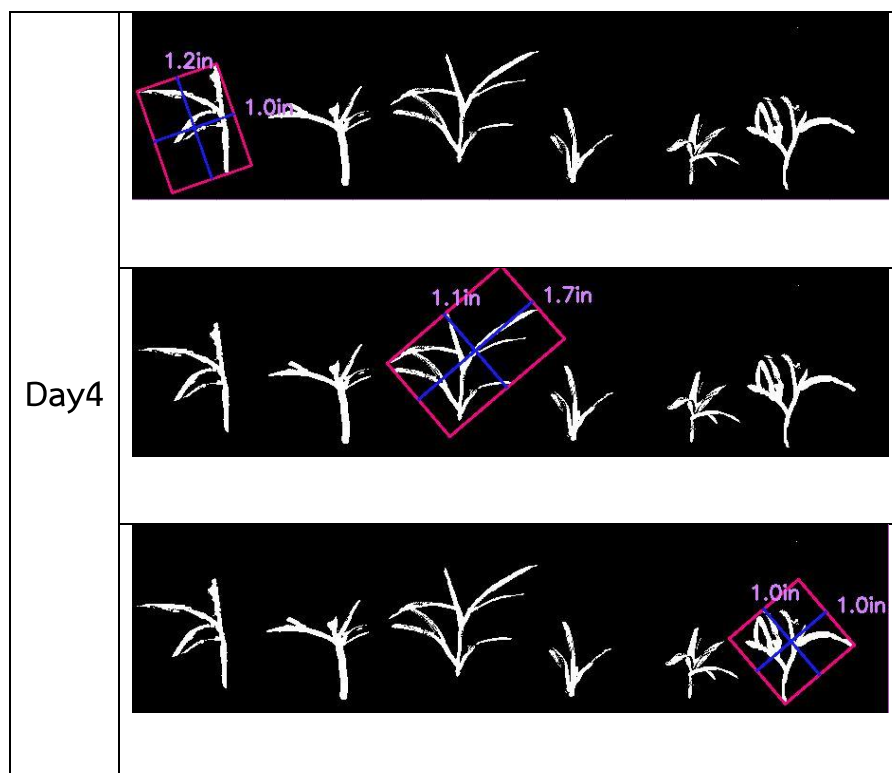
ตารางที่4.2 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่1 4



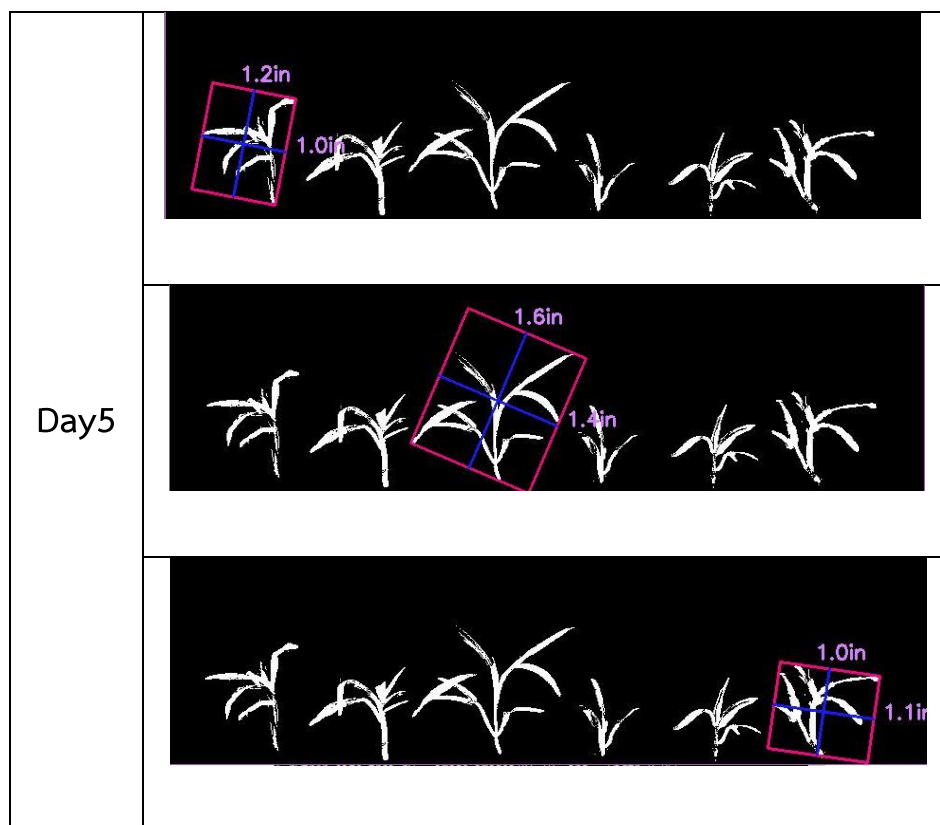
ตารางที่4.3 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่2 5



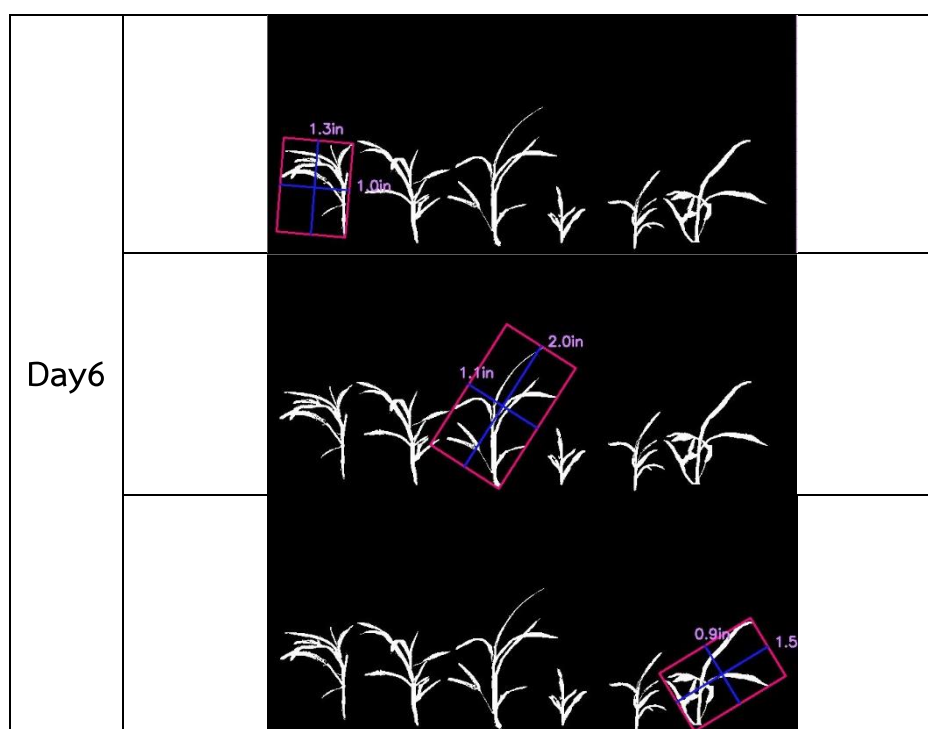
ตารางที่4.4 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่3 6



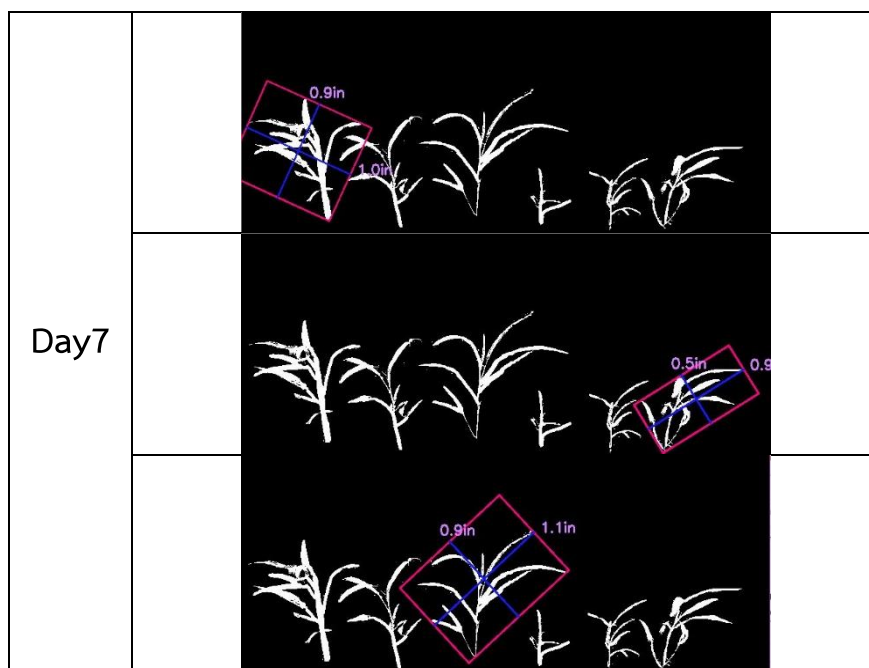
ตารางที่4.5 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่4 7



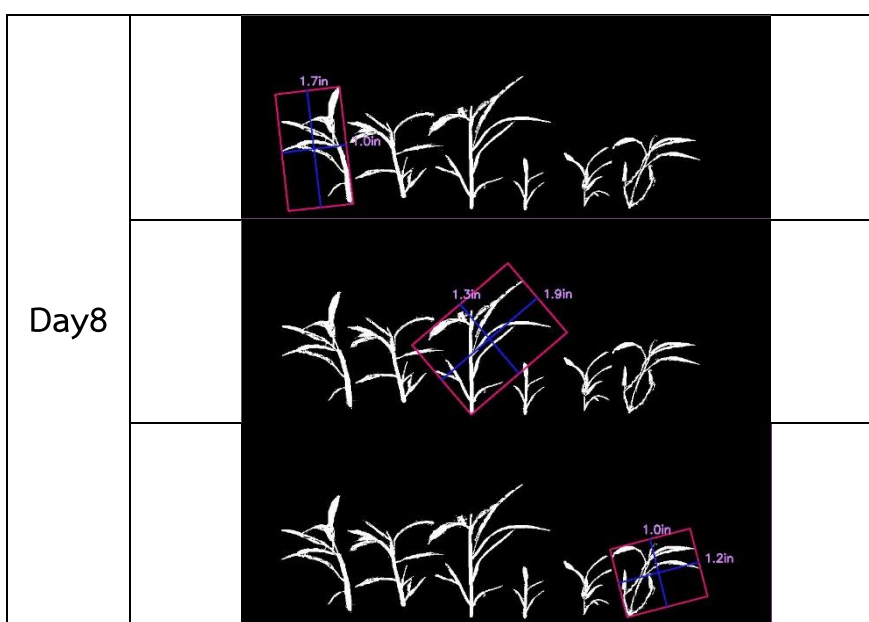
ตารางที่4.6 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่5 ๘



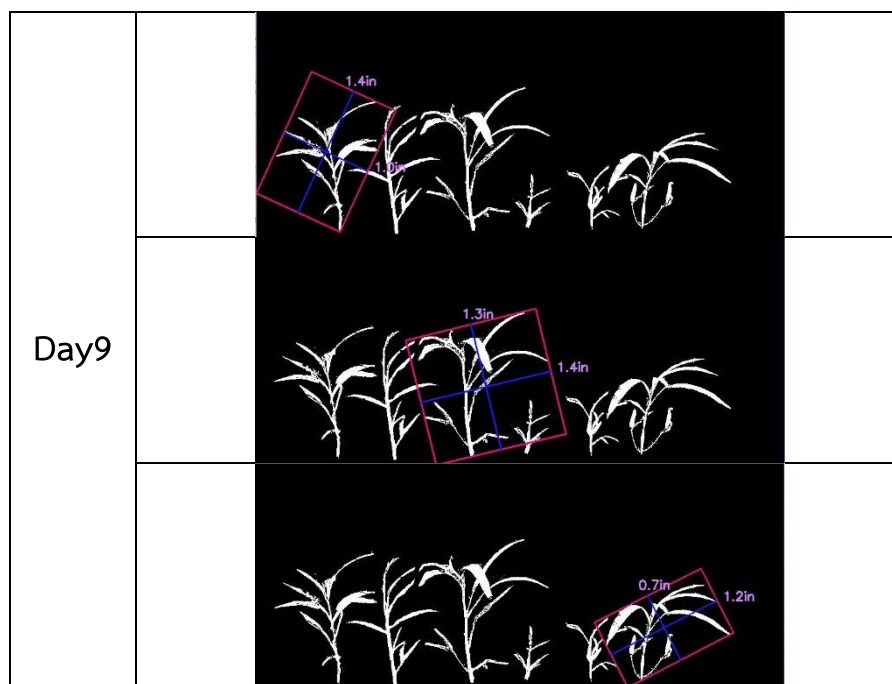
ตารางที่4.7 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่6 ๙



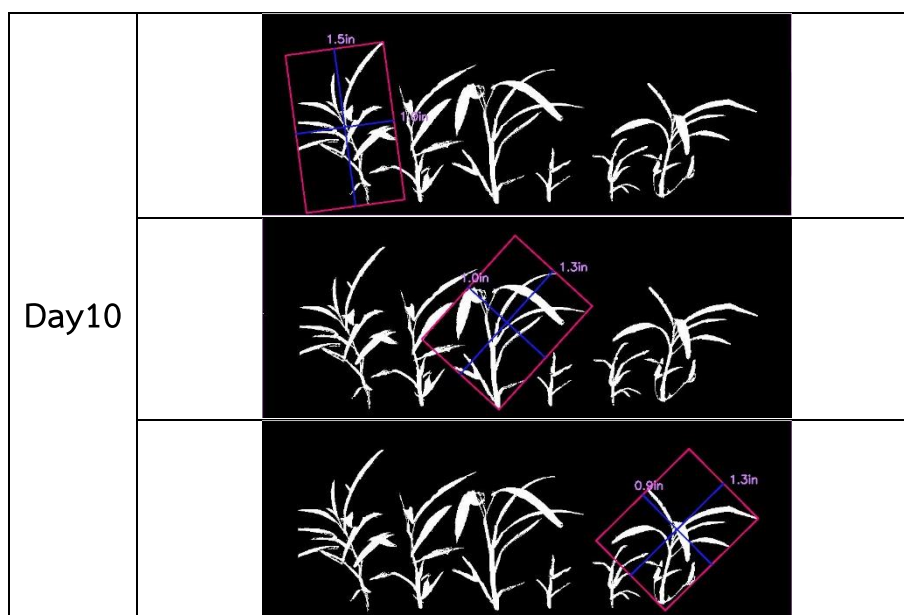
ตารางที่ 4.8 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่ 7 ¹⁰



ตารางที่ 4.9 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่ 8 ¹¹

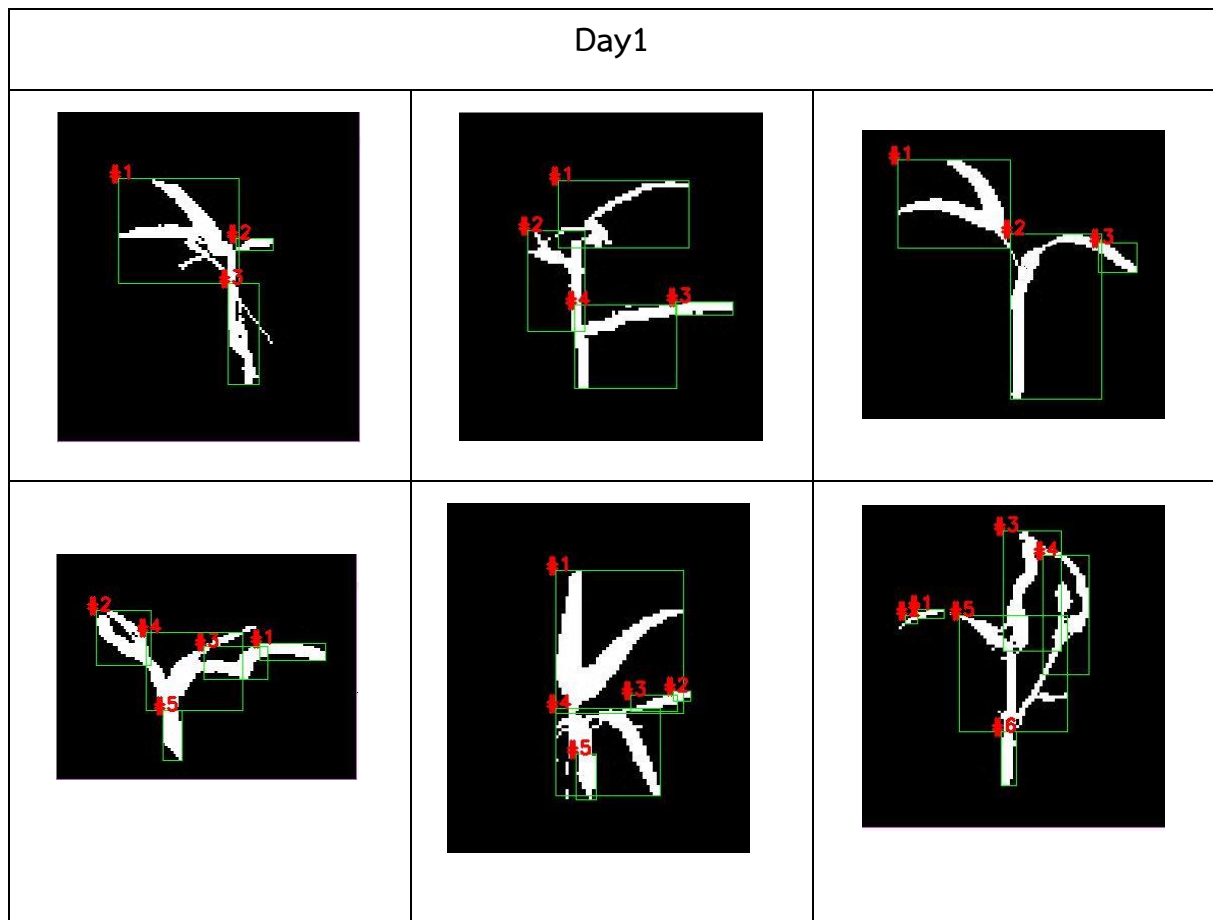


ตารางที่4.10 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่9 ¹²

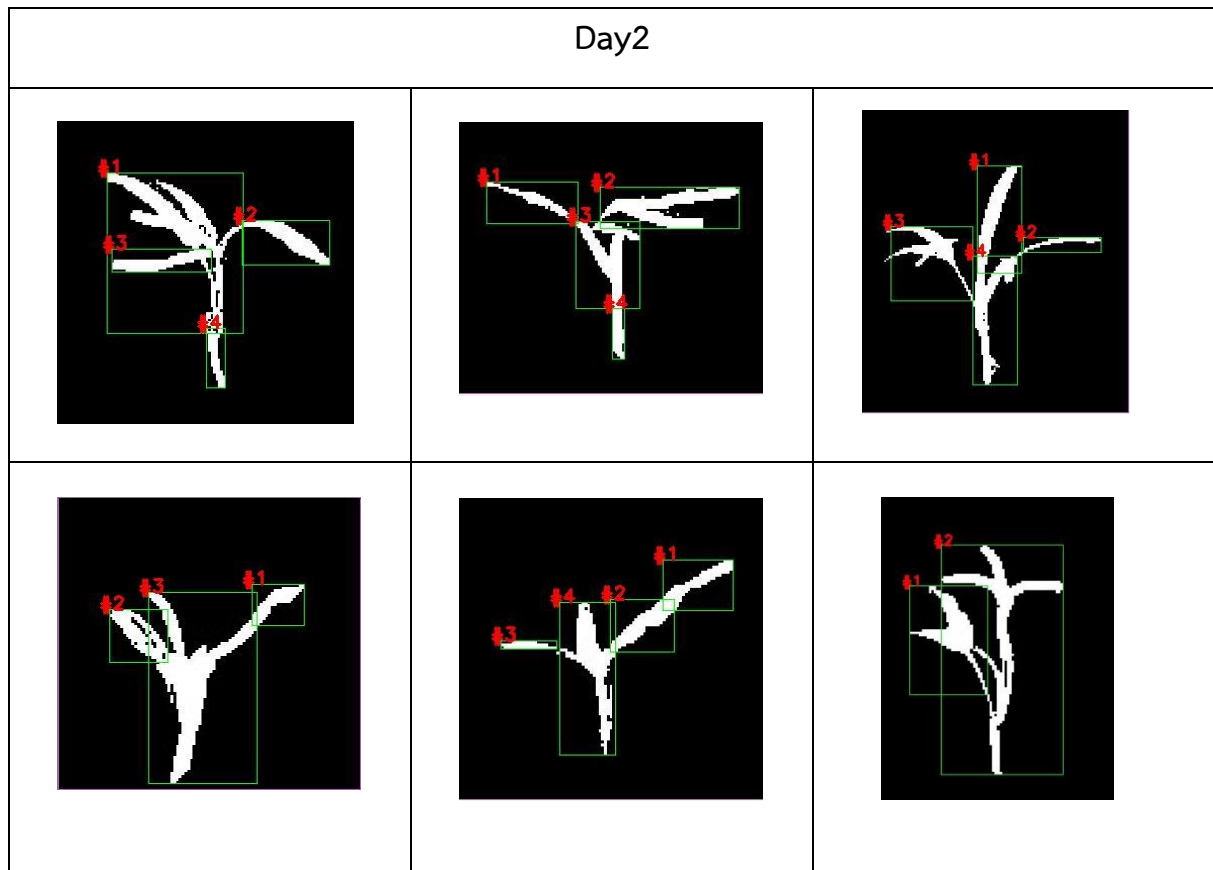


ตารางที่4.11 ตารางแสดงผลลัพธ์ที่ได้จากการวัดขนาดของวัตถุบางส่วนในวันที่10 ¹³

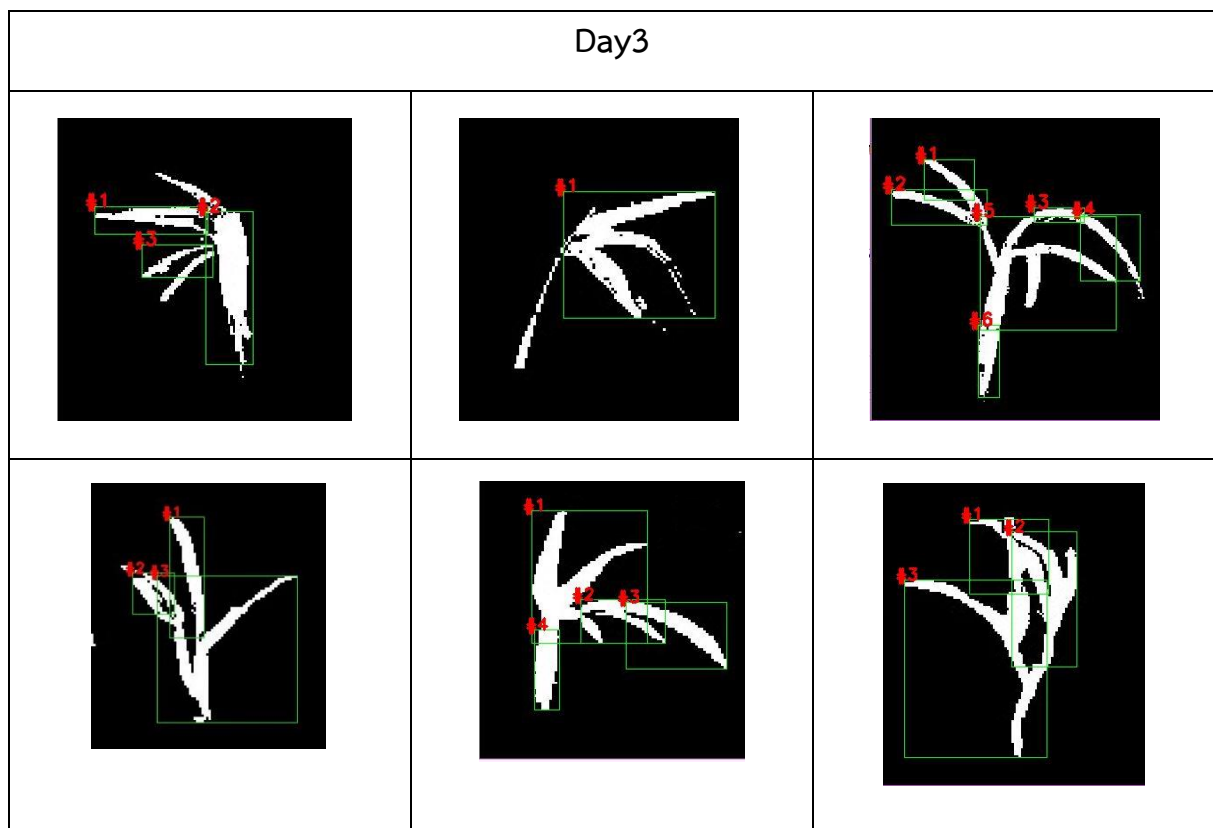
4.1.3 ผลลัพธ์ที่ได้จากการนับจำนวนใบ



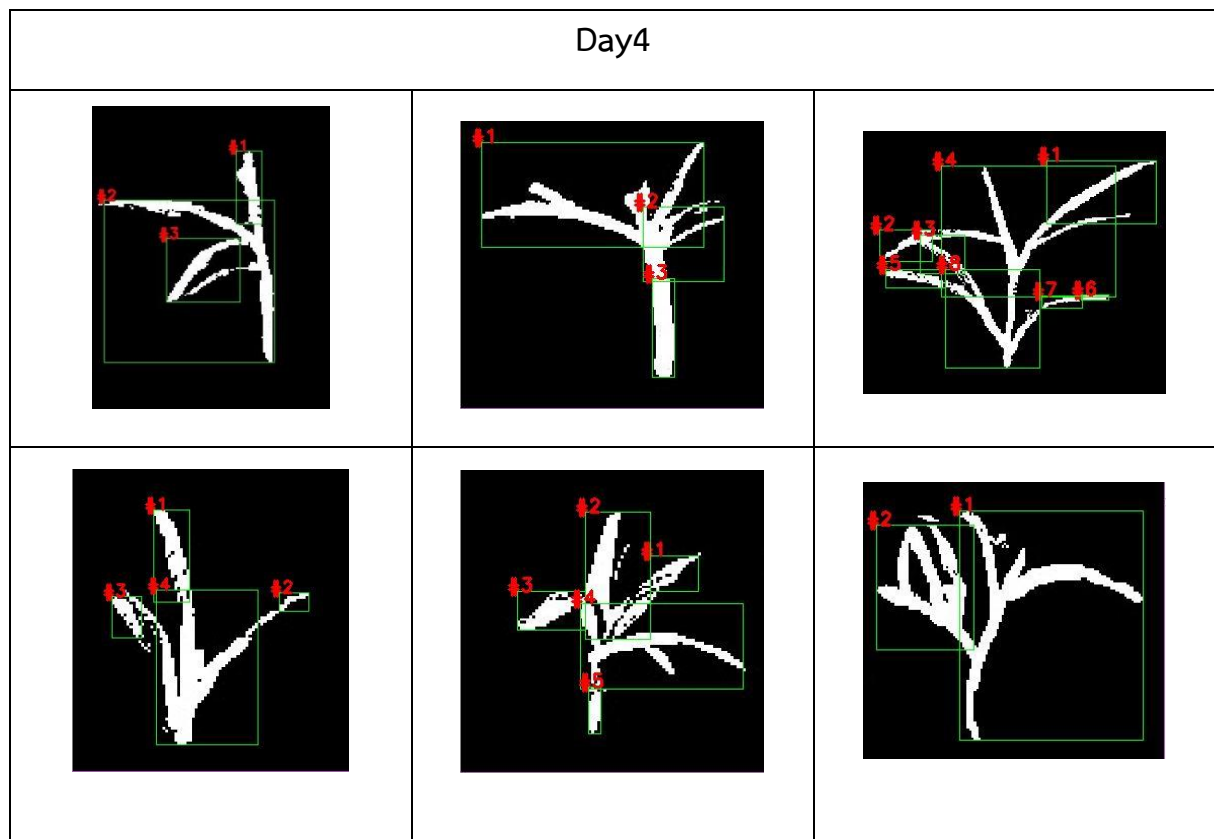
ตารางที่ 4.12 ตารางแสดงผลที่ได้จากการนับจำนวนใบไม้ในวันที่ 1¹⁴



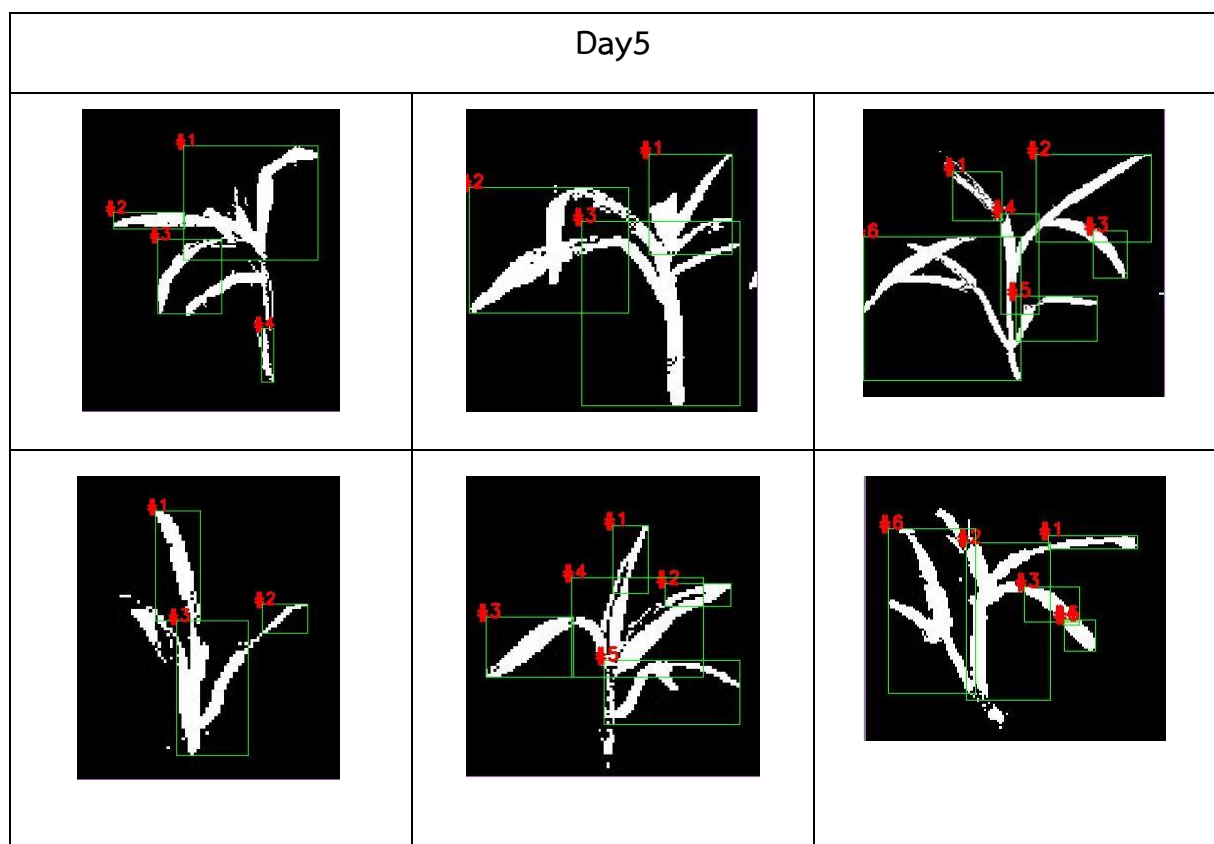
ตารางที่ 4.13 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่ 2 15



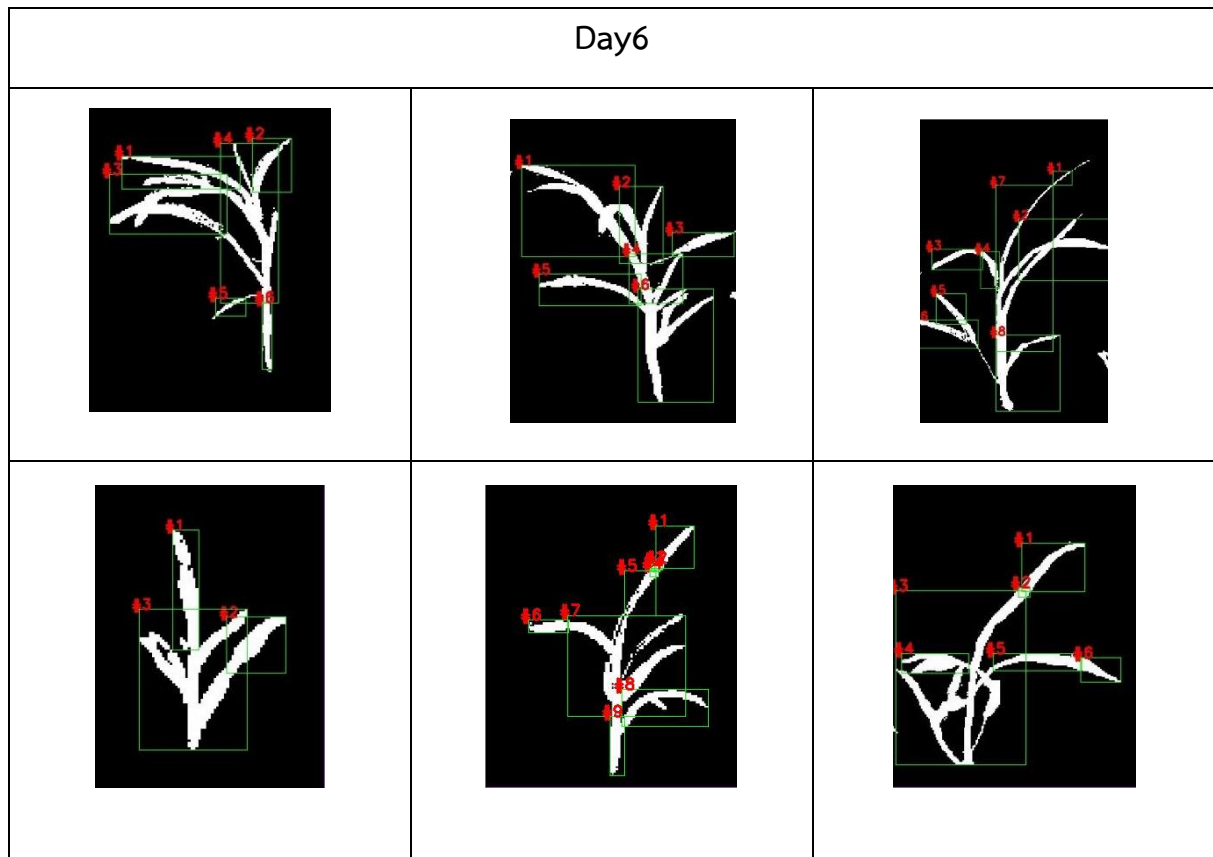
ตารางที่ 4.14 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่ 3 16



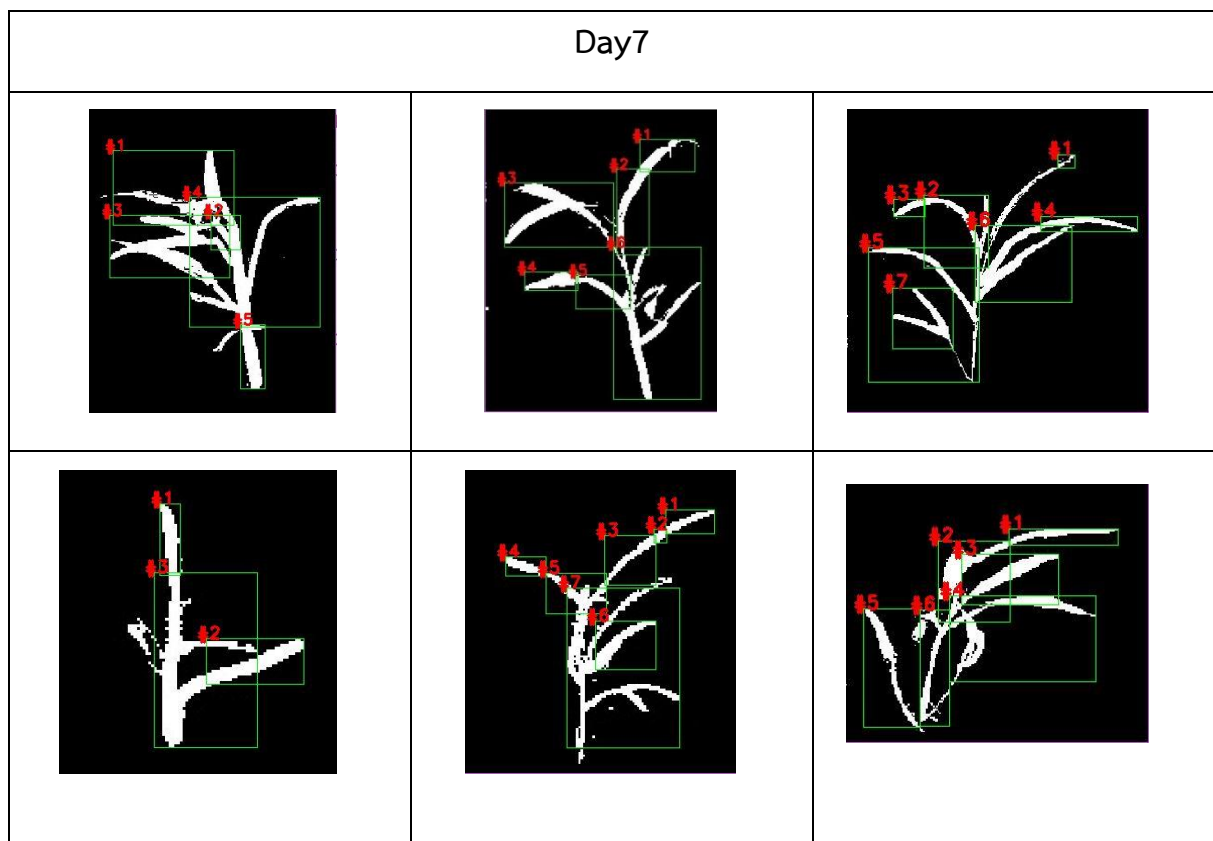
ตารางที่4.15 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่4 17



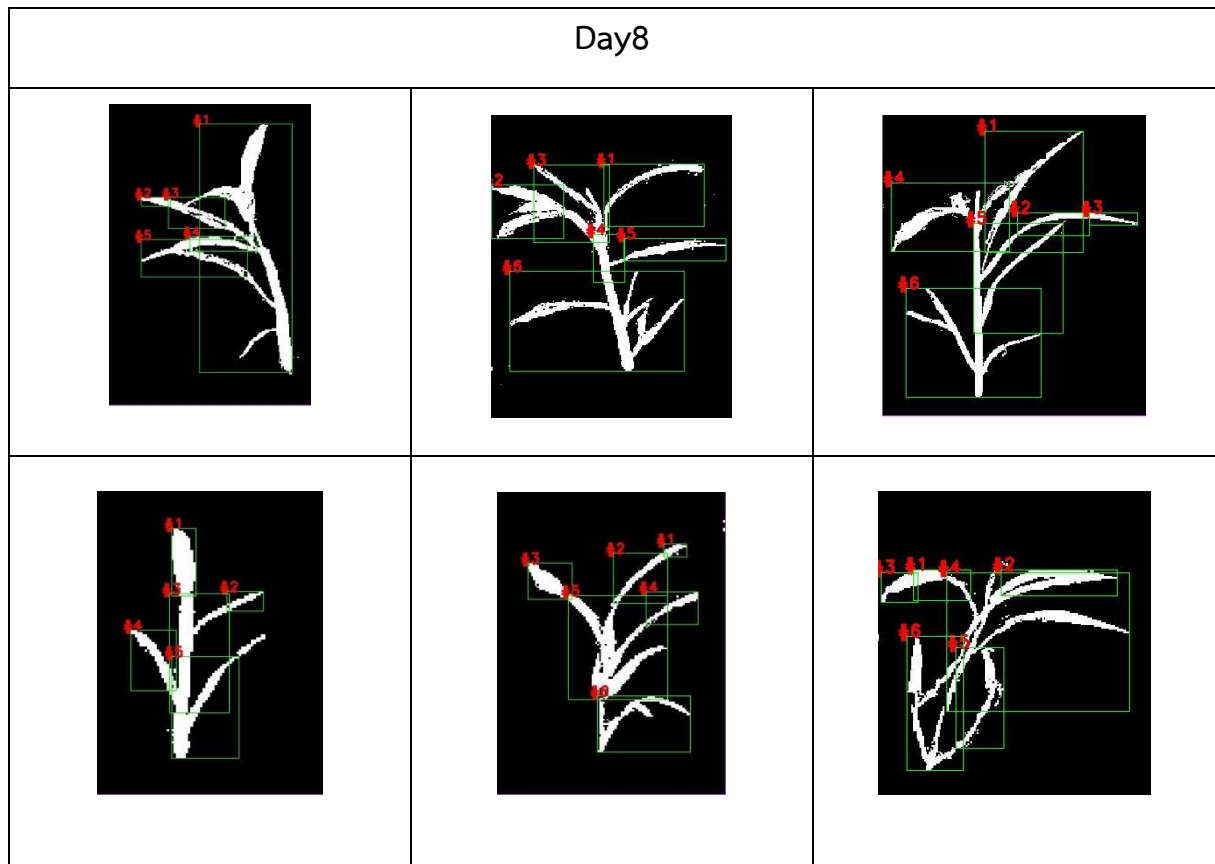
ตารางที่4.16 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่5 18



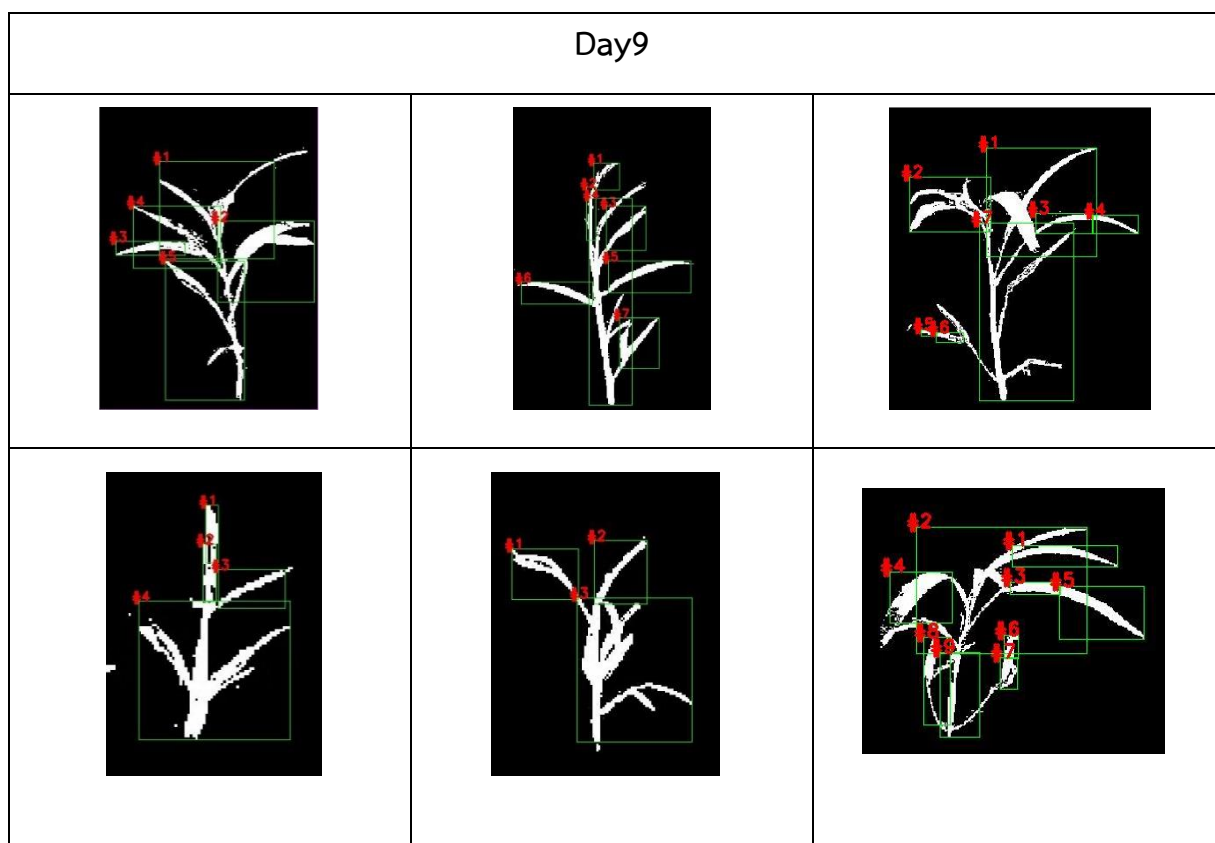
ตารางที่4.17 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่6 19



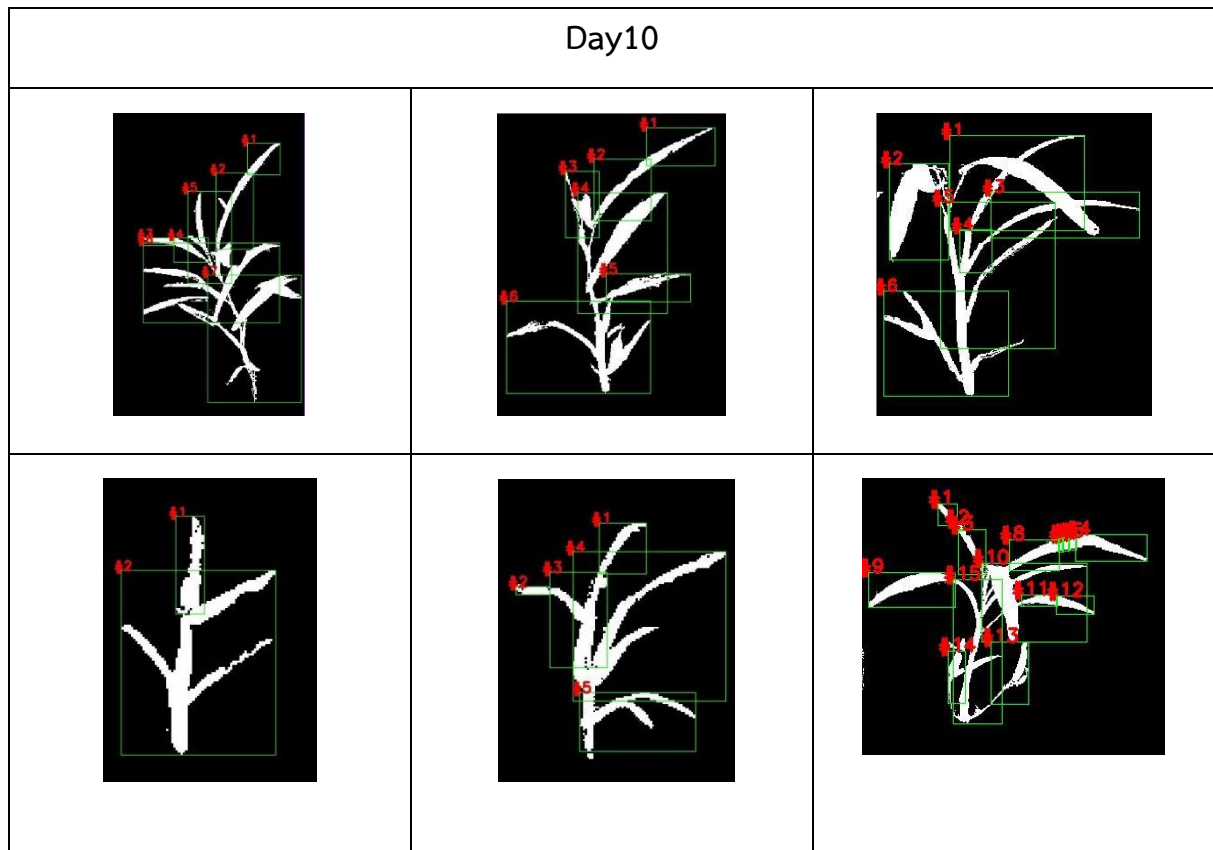
ตารางที่4.17 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่7 20



ตารางที่4.18 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่ 8 21



ตารางที่4.19 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่ 9 22



ตารางที่4.20 ตารางแสดงผลลัพธ์ที่ได้จากการนับจำนวนใบไม้ในวันที่10 23

4.1.4 ข้อมูลที่เก็บใน csv ไฟล์

Date	No	Width	Height	Leaf
1	1	1.27	2.54	3
1	2	1.578	1.878	4
1	3	1.132	2.286	3
1	4	1.232	1.778	2
1	5	1.27	2.286	2
1	6	1.232	2.286	3
2	1	1.778	2.54	4
2	2	2.04	2.278	4
2	3	2.086	2.286	4
2	4	1.27	2.286	3
2	5	1.524	2.032	4
2	6	1.27	2.286	3
3	1	1.27	1.74	3
3	2	1.27	1.778	3
3	3	1.54	2.202	5
3	4	1.27	1.778	3
3	5	1.278	1.778	3
3	6	1.478	2.286	3
4	1	2.54	3.048	3
4	2	2.54	3.048	3
4	3	2.794	3.318	5
4	4	1.524	1.778	3
4	5	1.678	1.778	4
4	6	2.54	2.74	3
5	1	2.54	2.54	4
5	2	2.486	2.486	3

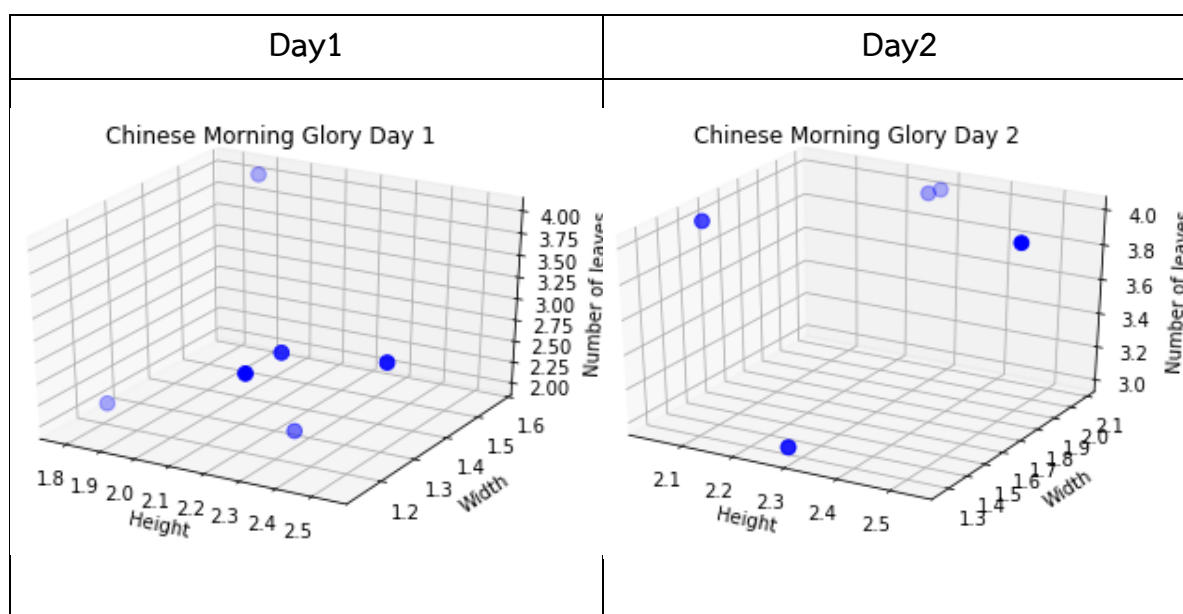
Date	No	Width	Height	Leaf
5	3	2.602	2.656	5
5	4	1.27	1.524	3
5	5	1.324	1.532	4
5	6	2.54	2.032	5
6	1	2.54	3.302	6
6	2	2.648	3.556	7
6	3	2.794	3.403	8
6	4	1.27	1.524	3
6	5	1.378	1.686	4
6	6	2.286	3.01	6
7	1	2.286	2.54	7
7	2	2.286	2.478	6
7	3	2.286	2.694	7
7	4	0.762	1.016	3
7	5	0.862	1.17	4
7	6	1.27	2.286	6
8	1	2.54	4.318	5
8	2	2.602	4.256	6
8	3	2.502	4.426	6
8	4	1.27	1.524	3
8	5	1.378	1.632	4
8	6	2.54	4.048	6
9	1	2.54	3.556	8
9	2	2.432	3.594	7
9	3	2.402	3.556	7
9	4	0.762	1.27	4
9	5	0.816	1.378	3

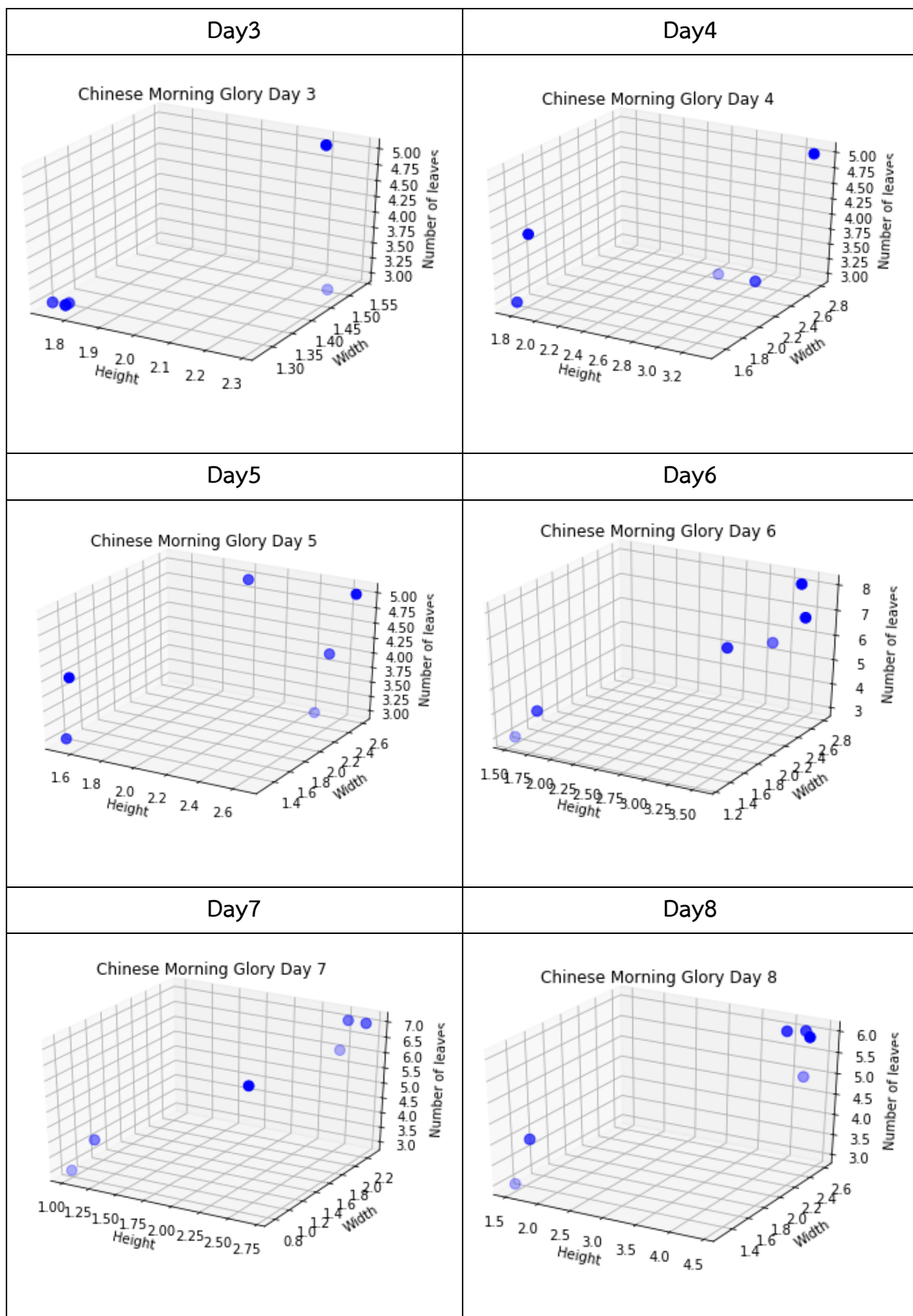
Date	No	Width	Height	Leaf
9	6	2.388	3.248	6
10	1	2.54	3.81	7
10	2	2.032	3.556	6
10	3	2.54	3.302	6
10	4	0.762	1.016	2
10	5	1.27	1.27	5
10	6	2.286	3.302	8

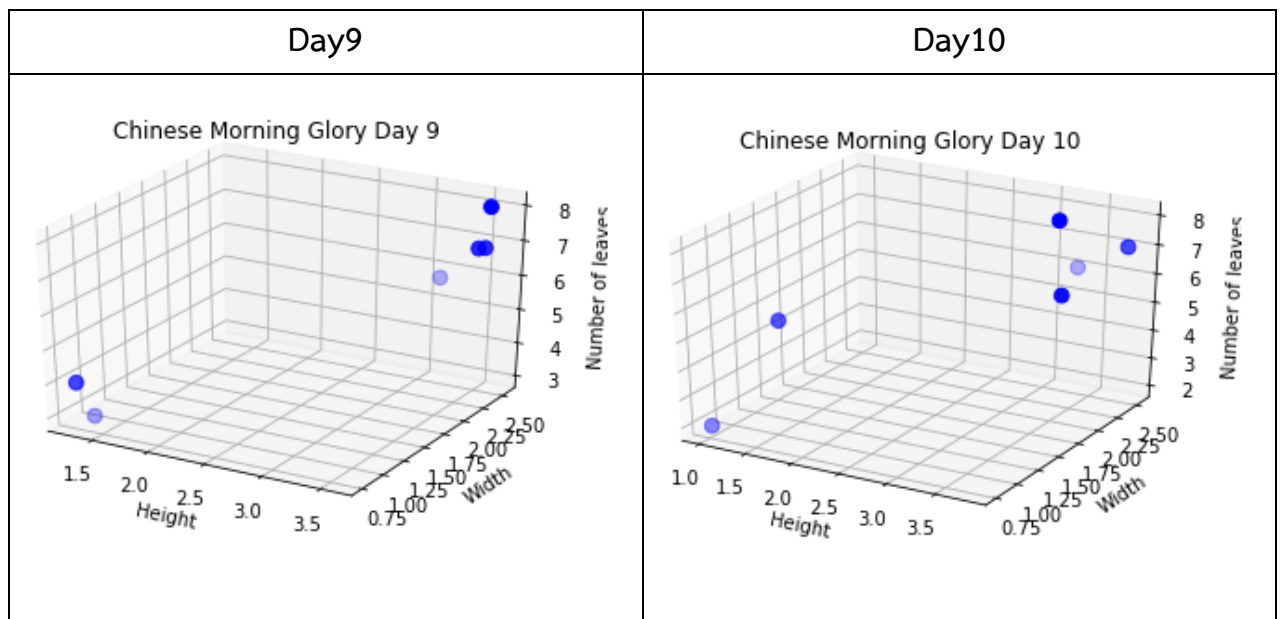
ตารางที่ 4.21 ตารางแสดงข้อมูลใน csv ไฟล์ 24

4.2 ผลลัพธ์ที่ได้จากการทดสอบ

จากการทดลองกับต้นผักบุ้งทั้งหมด 6 ต้น โดยในโครงการนี้จะพิจารณา 3 features คือ ความกว้างลำต้น , ความสูงลำต้น และ จำนวนใบ และใช้หลักการ Outlier Detection ในการตรวจสอบต้นไม้ที่มีการเจริญเติบโตแบบผิดปกติ ซึ่งผลการดำเนินงานตรวจพบว่ามีต้นผักบุ้ง 2 ต้นจาก 13 ต้นมีการเจริญเติบโตแบบผิดปกติ







ตารางที่ 4.22 ตารางแสดงผลพื้ที่ได้จากระบวนการ Outlier Detection 25

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการดำเนินการ

การศึกษามูลค่าการเจริญเติบโตของพืช ได้มีการศึกษาและทดลองเพื่อหาวิธีการที่ดีที่สุดในการปลูกพืช ซึ่งมีงานทดลองและวิจัยเกี่ยวข้องกับงานด้านนี้เป็นจำนวนมาก ซึ่งส่วนใหญ่เป็นการใช้องค์ความรู้เกี่ยวกับ Supervised Learning ในการทดลองจำเป็นต้องมีฐานข้อมูล เพื่อนำมาเปรียบเทียบกับภาพที่นำมาใช้ในการประมวลผล เช่น ต้องการจะตรวจสอบโรคของต้นมะม่วง โดยใช้ใบของต้นมะม่วงที่เป็นโรคมารตรวจสอบ จำเป็นที่จะต้องมีการมีฐานข้อมูลของใบต้นมะม่วงที่เป็นโรคอยู่แล้วเพื่อใช้ในการตรวจสอบชนิดของโรคที่จะเกิดขึ้นได้ ทำให้ผู้วิจัยต้องทำการเก็บข้อมูลเพื่อทำฐานข้อมูลให้ครอบคลุมกับโรคที่มีอยู่ ซึ่งจะต้องใช้เวลามากเพื่อให้ได้ฐานข้อมูลที่มีคุณภาพ และสามารถตรวจสอบได้เพียงชนิดของโรค และชนิดของต้นไม้ที่มีในฐานข้อมูลเท่านั้น ในโครงการนี้ได้นำองค์ความรู้เกี่ยวกับ Unsupervised Learning มาใช้ในงานวิจัย เพื่อลดปัญหาและข้อจำกัดที่เกิดขึ้นเมื่อใช้องค์ประกอบ Supervised Learning สถานที่ทดลองของการจัดทำโครงการนี้คือ คณะเกษตร มหาวิทยาลัยเกษตรศาสตร์ ผู้วิจัยทำการเก็บข้อมูลต้นผักบุ้งจาก IP Camera ในงานวิจัยนี้ใช้ต้นผักบุ้ง จำนวน 6 ต้น เป็นพืชทดสอบ ใช้ระยะเวลาในการเก็บข้อมูลการเจริญเติบโตทั้งสิ้น 21 วัน

จากการพัฒนาโครงการการใช้เทคนิคประมวลผลภาพแบบไม่มีผู้สอนเพื่อตรวจจับการเจริญเติบโตที่ผิดปกติของต้นไม้ โดยการทำงานจะแบ่งเป็น 3 ส่วน โดยส่วนที่1 คือการเก็บข้อมูลต้นผักบุ้งจาก IP Camera และสามารถวิเคราะห์หาต้นไม้ออกจากภาพ ส่วนที่2 นำข้อมูลที่ได้ไปวิเคราะห์และดึง feature ที่สำคัญออกมา ในโครงการนี้สนใจ 3 feature คือ ความกว้างลำต้น , ความสูงลำต้น และ จำนวนใบ จากนั้นเก็บข้อมูลที่ได้อัปโหลดไฟล์ .csv ซึ่งในส่วนนี้ได้พบว่าภาพในบางช่วงมีการซ้อนทับกันจึงทำให้เกิดความคลาดเคลื่อนในการดึง feature เล็กน้อย และส่วนที่3 นำข้อมูลที่ได้มาแสดงผลในรูปแบบกราฟ และใช้หลักการ Outlier Detection เพื่อตรวจสอบการเจริญเติบโตที่ผิดปกติของต้นไม้ โดยผลการดำเนินงานตรวจพบว่ามีต้นผักบุ้ง 2 ต้นจาก 13 ต้นมีการเจริญเติบโตแบบผิดปกติ

5.2 ปัญหาและอุปสรรคจากการดำเนินงาน

1. เนื่องจากสถานที่ที่ปลูกต้นไม้ไม่มีสัญญาณอินเทอร์เน็ตทำให้ไม่สามารถเก็บข้อมูลเป็นแบบเรียลไทม์ได้
2. ปัจจัยทางสภาพอากาศ และเมล็ดพันธุ์พืชทำให้การเพาะเมล็ดเกิดข้อผิดพลาด ต้นไม้เจริญเติบโตล่าช้าและไม่สามารถเจริญเติบโตได้ดีเท่าที่ควร
3. การเจริญเติบโตของต้นไม้ไม่สามารถควบคุมได้ ต้นไม้บางต้นซึ่งมีใบไม้ของแต่ละต้นซ้อนทับกัน จึงทำให้การตรวจนับจำนวนใบไม้ของแต่ละต้นเกิดข้อผิดพลาดจากความเป็นจริงเล็กน้อย

5.3 ข้อเสนอแนะ

ในอนาคตเราสามารถพัฒนาโครงการนี้ได้โดยการเพิ่ม feature ในการตรวจจับต้นไม้ได้มากขึ้น เช่น ตรวจจับสีของต้นไม้เพื่อดูความผิดปกติ หรืออาจจะเป็นโรคได้ ทำแอปพลิเคชันเพื่อให้ผู้ใช้สามารถตรวจสอบการเจริญเติบโตของต้นไม้ได้แบบเรียลไทม์โดยการติดตั้งไว้ในอุปกรณ์สื่อสาร เช่น โทรศัพท์มือถือ แท็บเล็ต ซึ่งส่งผลให้มีประสิทธิภาพมากยิ่งขึ้น

เอกสารอ้างอิง

- [1] . J. Jerrin and M. P. Manu, "Plant growth monitoring system, with dynamic user-interface," in *2016 IEEE Region 10 Humanitarian Technology Conference*, Agra, India, 2016.
- [2] Q. Li , C. Jinhai , B. Bettina and M. Stan , "Study on spike detection of cereal plants," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, Singapore, Singapore, 2014.
- [3] Z. Wenqiang , L. Wei , Y. Zhenyu and H. Jianda , "Height information acquisition method of seedling with machine vision," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, Shenyang, China, 2015. [1]
- [4] D. K. Sachin and P. A.B. , "Plant Disease Detection Using Image Processing," in *2015 International Conference on Computing Communication Control and Automation*, Pune, India, 2015.
- [5] "K-Means Clustering in OpenCV," 10 November 2014. [Online]. Available: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_ml/py_kmeans/py_kmeans_opencv/py_kmeans_opencv.html.
- [6] "Tracking multiple objects with OpenCV," 8 August 2018. [Online]. Available: <https://www.pyimagesearch.com/2018/08/06/tracking-multiple-objects-with-opencv/>.
- [7] "Measuring size of objects in an image with OpenCV," 28 March 2016. [Online]. Available: <https://www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/>.
- [8] "Determining object color with OpenCV," 15 February 2016. [Online]. Available: <https://www.pyimagesearch.com/2016/02/15/determining-object-color-with-opencv/>.
- [9] "Color spaces in OpenCV (C++ / Python)," 7 May 2017. [Online]. Available: <https://www.learnopencv.com/color-spaces-in-opencv-cpp-python/>.
- [1] "k-means clustering," wikipedia, 10 May 2017. [Online]. Available: <https://th.wikipedia.org/wiki/การแบ่งกลุ่มข้อมูลแบบเคมีน>.
- [0] แบ่งกลุ่มข้อมูลแบบเคมีน. [Accessed 3 March 2019].
- [1] "ระยะทางแบบยุคลิด," wikipedia, 21 September 2023. [Online]. Available: <https://th.wikipedia.org/wiki/ระยะทางแบบยุคลิด>.
- [1] ระยะทางแบบยุคลิด. [Accessed 3 March 2019].

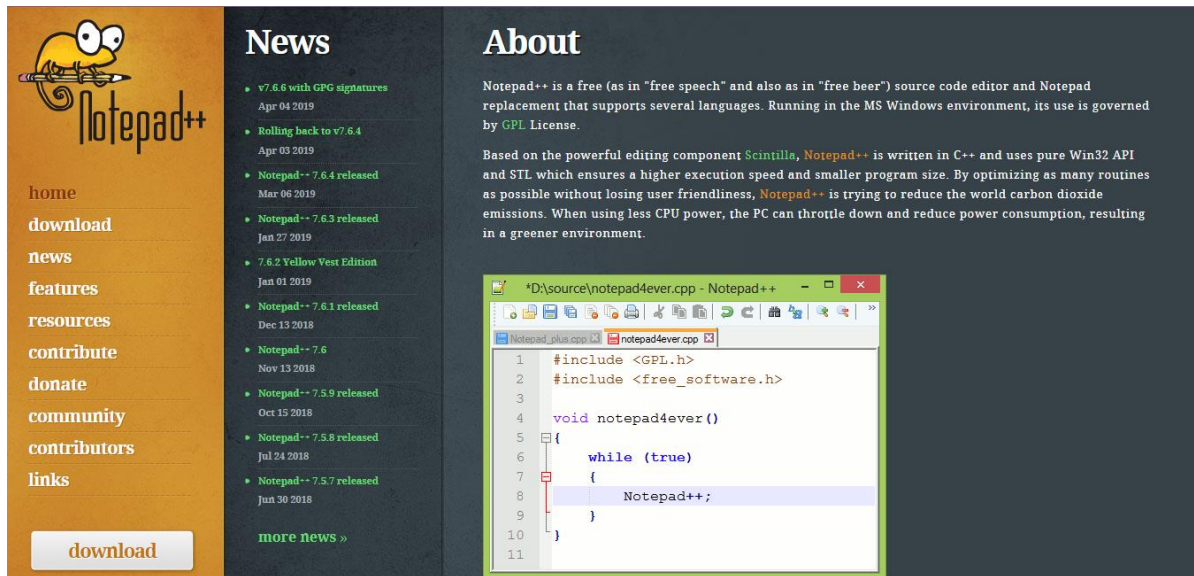
-
- [1] "numpy & matplotlib เบื้องต้น บทที่ ๔: การตัดแต่งแก้ไขอาเรย์," phyblas, 11 June 2016. [Online].
- [2] Available: <https://phyblas.hinaboshi.com/numa04>. [Accessed 4 March 2019].
- [1] "K-Means Clustering in OpenCV," morange, 15 August 2016. [Online]. Available:
- [3] https://wiki.morange.co.th/K-Means_Clustering#K-Means_Clustering_in_OpenCV. [Accessed 8 March 2019].
- [1] "Image Morphology," humancominteracg1.wixsite, [Online]. Available:
- [4] <https://humancominteracg1.wixsite.com/group1/w2-morphological>. [Accessed 17 March 2019].
- [1] "Background Subtraction," morange, 10 August 2016. [Online]. Available:
- [5] https://wiki.morange.co.th/Background_Subtraction. [Accessed 17 March 2019].
- [1] "Image Processing การหาขอบภาพ (Edge detection)," note005.blogspot, 28 August 2011. [Online].
- [6] Available: <http://note005.blogspot.com/2011/08/image-processing-edge-detection.html>. [Accessed 17 March 2019].
- [1] "OpenCV People Counter," pyimagesearch, 13 August 2018. [Online]. Available:
- [7] <https://www.pyimagesearch.com/2018/08/13/opencv-people-counter/>. [Accessed 8 April 2019].
- [1] "Contours in OpenCV," OpenCV, [Online]. Available: [https://opencv-python-](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#contours-getting-started)
- [8] [tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#contours-getting-started](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#contours-getting-started). [Accessed 13 April 2019].
- [1] "Image Filtering," OpenCV, 10 November 2014. [Online]. Available: [https://docs.opencv.org/3.0-](https://docs.opencv.org/3.0-beta/modules/imgproc/doc/filtering.html#pyrmeanshiftfiltering)
- [9] [beta/modules/imgproc/doc/filtering.html#pyrmeanshiftfiltering](https://docs.opencv.org/3.0-beta/modules/imgproc/doc/filtering.html#pyrmeanshiftfiltering). [Accessed 14 April 2019].
- [2] "Finding extreme points in contours with OpenCV," pyimagesearch, 11 April 2016. [Online]. Available:
- [0] <https://www.pyimagesearch.com/2016/04/11/finding-extreme-points-in-contours-with-opencv/>. [Accessed 14 April 2019].
- [2] "OpenCV Coin Detection Project," tremaineconsultinggroup, 27 June 2018. [Online]. Available:
- [1] <https://tremaineconsultinggroup.com/opencv-coin-detection-project/>. [Accessed 15 April 2019].
- [2] "Watershed OpenCV," pyimagesearch, 2 November 2015. [Online]. Available:
- [2] <https://www.pyimagesearch.com/2015/11/02/watershed-opencv/>. [Accessed 15 April 2019].
- [2] "Gesture recognition using OpenCV and Python," Github, 12 March 2018. [Online]. Available:
- [3] <https://github.com/avidLearnerInProgress/hand-gesture-recognition>. [Accessed 15 April 2019].

ภาคผนวก

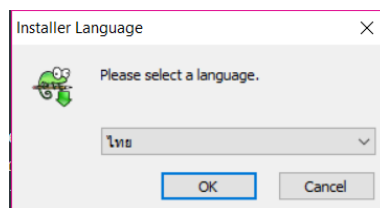
ภาคผนวก ก : การติดตั้งโปรแกรมที่ใช้พัฒนา

Notepad++

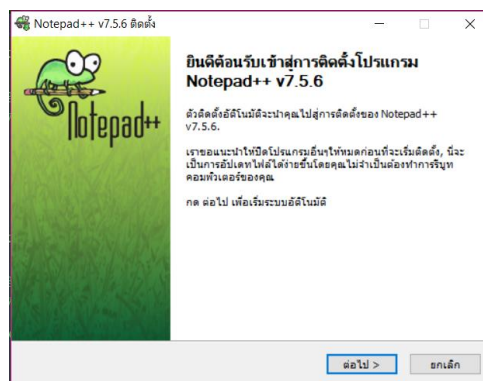
1. เข้าเว็บไซต์ <https://notepad-plus-plus.org/> เลือก Download และเลือกเวอร์ชันที่ต้องการเพื่อติดตั้ง Notepad++



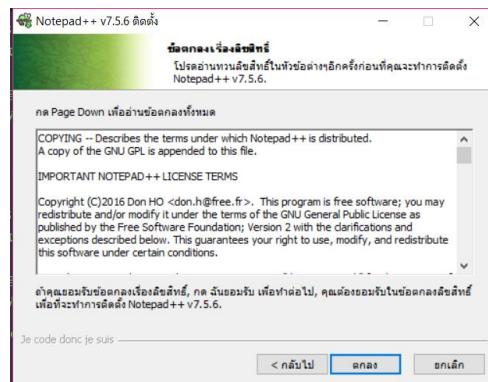
2. เปิดไฟล์ที่ดาวน์โหลดมา เลือกภาษาที่ต้องการติดตั้ง แล้วคลิก OK



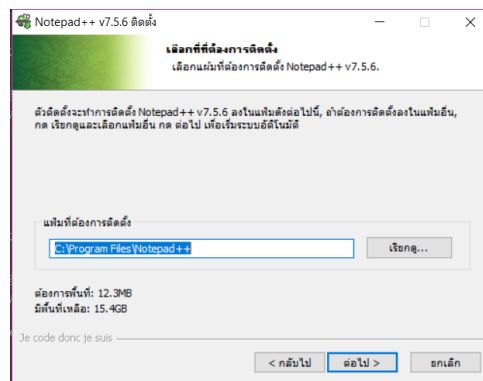
3. คลิก ต่อไป



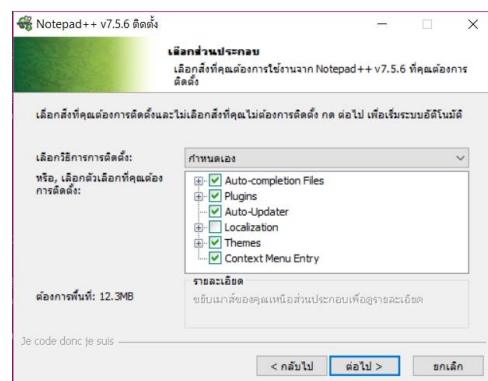
4. คลิก ตกลง



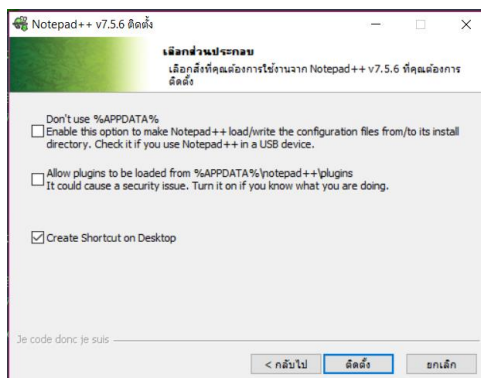
5. เลือกที่จัดเก็บไฟล์โปรแกรม แล้วคลิก ต่อไป



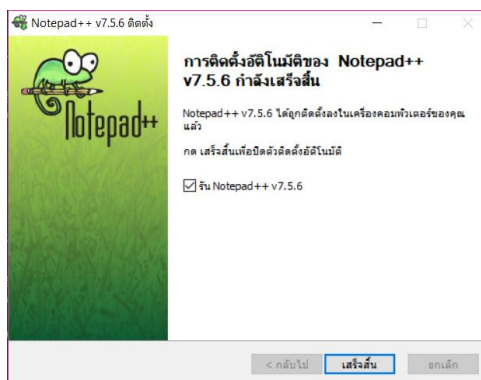
6. คลิก ต่อไป



7. ถ้าต้องการให้สร้าง shortcuts ที่ desktop ให้เลือกที่ Create Shortcut on Desktop แล้วคลิกติดตั้ง รอจนโปรแกรมเสร็จสิ้น



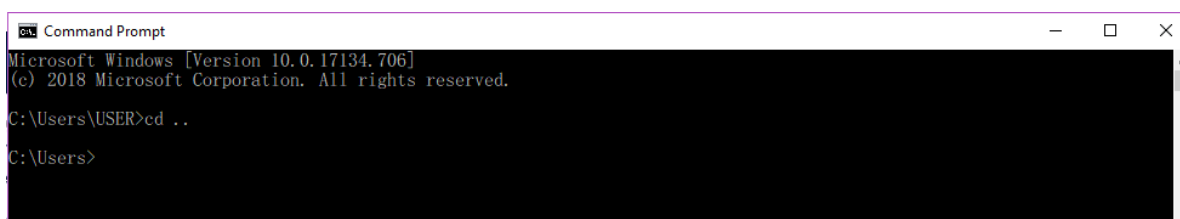
8. คลิก เสร็จสิ้น



ภาคผนวก ข : การ Setup/Configure

การติดตั้ง OpenCV

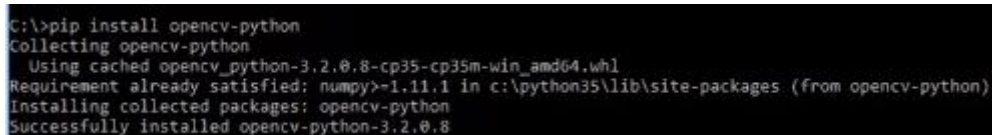
1. เปิด command prompt
2. Move to drive C://



```
Command Prompt
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\USER>cd ..
C:\Users>
```

3. พิมพ์ pip install opencv-python



```
C:\>pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-3.2.0.8-cp35-cp35m-win_amd64.whl
Requirement already satisfied: numpy>=1.11.1 in c:\python35\lib\site-packages (from opencv-python)
Installing collected packages: opencv-python
Successfully installed opencv-python-3.2.0.8
```

4. ทดสอบการใช้งานด้วยการพิมพ์คำว่า python แล้วตามด้วย import cv2



```
C:\>python
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import cv2
>>> _
```