



การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติ  
ด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ  
Development Algorithm for Automatic Extraction of fruit trees  
from UAV imagery

อุไรรัตน์ อุ่นเมือง

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

วิทยานิพนธ์ระดับปริญญาตรี เสนอภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม

คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร

เพื่อเป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

สาขาวิชาภูมิศาสตร์

พฤศจิกายน 2562

ลิขสิทธิ์เป็นของมหาวิทยาลัยนเรศวร

อาจารย์ที่ปรึกษา ประธานบริหารหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาภูมิศาสตร์ และ  
หัวหน้าภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและ  
สิ่งแวดล้อม ได้พิจารณาวิทยานิพนธ์ระดับปริญญาตรี เรื่อง “การพัฒนาอัลกอริทึมเพื่อการตรวจนับ  
จำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ” เห็นสมควรรับเป็นส่วนหนึ่ง  
ของการศึกษาหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาภูมิศาสตร์ ของมหาวิทยาลัยนเรศวร



.....  
(ผู้ช่วยศาสตราจารย์ ดร.นัฐพล มหาวิค)  
อาจารย์ที่ปรึกษา



.....  
(รองศาสตราจารย์พัฒนา ราชวงศ์)  
ประธานบริหารหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาภูมิศาสตร์



.....  
(รองศาสตราจารย์พัฒนา ราชวงศ์)  
หัวหน้าภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดีด้วยความอนุเคราะห์ช่วยเหลือเป็นอย่างดีจากคณาจารย์ และผู้ที่เกี่ยวข้องหลายท่านที่ช่วยเหลือในการลงพื้นที่และให้ข้อมูลอันเป็นประโยชน์ ขอขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.นัฐพล มหาวิค อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้สละเวลาอันมีค่ามาเป็นทีปรึกษา ให้คำแนะนำ ช่วยเหลือในเรื่องอุปกรณ์อากาศยานไร้คนขับ การดูแลเป็นอย่างดีในการดำเนินการลงพื้นที่ภาคสนาม ช่วยเหลือในการให้ความรู้ข้อมูล ข้อเสนอแนะอันเป็นประโยชน์ ตลอดจนตรวจสอบแก้ไขข้อบกพร่องของวิทยานิพนธ์ด้วยความเอาใจใส่ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างสมบูรณ์

ขอขอบพระคุณ เจ้าของสวนมะนาว จ.สุโขทัย จ.พิษณุโลก นายดำรงณ พรมมี และ นายประเสริฐ เรืองฉาย ที่ให้ความอนุเคราะห์ในการเข้าถ่ายพื้นที่ศึกษาโดย UAV นอกจากนี้ ขอขอบคุณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ให้ความอนุเคราะห์ซอฟต์แวร์ MATLAB ในการศึกษาครั้งนี้

ขอขอบพระคุณคณาจารย์สาขาวิชาภูมิศาสตร์ทุกท่านที่ให้คำปรึกษาชี้แนะแนวทางในการดำเนินงานวิจัย และเพื่อน ๆ สาขาวิชาภูมิศาสตร์ ที่ให้ความช่วยเหลือในเรื่องการลงพื้นที่เพื่อเก็บข้อมูล และให้กำลังใจจนทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์

ที่สำคัญขอกราบขอบพระคุณ บิดา มารดาและผู้ปกครอง ของผู้วิจัย ที่ให้การสนับสนุนในทุก ๆ ด้าน ไม่ว่าจะเป็นการส่งค่าเล่าเรียน ส่งกำลังใจให้ จนทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์

อุไรรัตน์ อุ้มเมือง

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

ชื่อเรื่อง	การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ
ผู้วิจัย	อุไรรัตน์ อุ่นเมือง
ที่ปรึกษาวิทยานิพนธ์	ผู้ช่วยศาสตราจารย์ ดร.นัฐพล มหาวีค
ประเภทสารนิพนธ์	วิทยานิพนธ์ วท.บ สาขาวิชาภูมิศาสตร์, มหาวิทยาลัยนเรศวร, 2562
คำสำคัญ	อากาศยานไร้คนขับ การสกัดขอบเขตต้นไม้ การประมวลผลภาพดิจิทัล

### บทคัดย่อ

จำนวนของต้นไม้ที่อยู่ในแปลงเกษตรนั้นเป็นข้อมูลที่สำคัญในการจัดการด้านเกษตรกรรม เนื่องจากจำนวนต้นไม้สามารถนำมาประมาณผลผลิตต่อพื้นที่เพาะปลูกได้ เพื่อเป็นข้อมูลช่วยตัดสินใจในการบริหารจัดการพื้นที่การเกษตรให้มีประสิทธิภาพมากยิ่งขึ้น ในการศึกษาครั้งนี้มุ่งที่จะพัฒนาอัลกอริทึมในโปรแกรม MATLAB เพื่อการตรวจนับจำนวนต้นไม้ผลแบบอัตโนมัติจากข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ คณะผู้วิจัยได้พัฒนา 3 เทคนิค คือ 1. การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ 2. การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ และ 3. การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ จากนั้นทำการตรวจสอบความถูกต้องของเทคนิค ในแต่ละเทคนิคกับข้อมูลที่ได้นับจริงจากการแปลภาพถ่ายด้วยสายตา ซึ่งได้ศึกษาพื้นที่แปลงเพาะปลูกสวนมะนาวใน อำเภอกีรีมาศ จังหวัดสุโขทัย โดยการศึกษาและพัฒนาอัลกอริทึมเบื้องต้นนั้นได้ใช้ข้อมูลภาพ Google map ของสวนไม้ผล 2 ชนิดคือ ส้มและทุเรียน จำนวนสวนละ 30 ภาพ จากการตรวจสอบความถูกต้องพบว่าร้อยละความถูกต้องของเทคนิค ที่ 1, 2 และ 3 มีค่าเท่ากับ 75%, 76% และ 88% ตามลำดับ และยังได้มีการศึกษาและพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ ซึ่งได้นำข้อมูลระดับความสูงเชิงเลข (Digital Surface Model หรือ DSM) เข้ามาใช้ในการพัฒนาอัลกอริทึม พบว่าร้อยละความถูกต้องของเทคนิคที่ 1, 2 และ 3 มีค่าเท่ากับ 74%, 79% และ 92% ตามลำดับ และยังพบว่าเมื่อนำข้อมูล DSM มาใช้ในการศึกษาส่งผลให้มีค่าความถูกต้องของการนับจำนวนต้นไม้ที่ได้จากอัลกอริทึมจะมีค่าสูงขึ้น

Copyright by Naresuan University

All rights reserved

**Title** DEVELOPMENT ALGORITHM FOR AUTOMATIC EXTRACTION OF FRUIT TREES FROM UAV IMAGERY

**Author** Urairat Unmueang

**Advisor** Assistant Professor Dr. Nattapon Mahavik

**Academic Paper** Thesis B.S. in Geography, Naresuan University, 2019

**Keywords** Unmanned Aerial Vehicle, Tree extraction, digital image processing

### ABSTRACT

Tree counts and sizes are important information to apply for crop yield estimation and agricultural planning. Therefore, implementation for automatic extraction of trees numbers from remotely sensed data is a challenging task. In this study, a novel approach and algorithm developed in MATLAB is proposed for the automatic extraction of citrus trees using unmanned aerial vehicle (UAV). Initially, preliminary testing by using Google image data have been done using of 2 fruit orchards, orange and durian with 30 images per orchard the research team 3 methods developed: 1. To count objects from its boundary 2. To examine the number of objects using the color its 3. To count the number of objects using its radius. To validate the performance of the developed extraction techniques, the ground truth images were interpreted. The accuracy of techniques 1, 2 and 3 has shown the absolute mean values of 75%, 76% and 88%, respectively. Improved accuracy results were found when the improved algorithm was included DSM (Digital Surface Model) generated from UAV data. It was found that the accuracy of techniques 1, 2 and 3 are 74%, 79% and 92%, respectively. It was found that the accuracy of counting the fruit tree obtained from the developed algorithm would be usually high incorporated with DSM data.

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved



## สารบัญ

บทที่	หน้า
1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ความสำคัญของการวิจัย.....	2
1.4 ขอบเขตของการวิจัย.....	2
1.4.1 ขอบเขตพื้นที่ศึกษา.....	2
1.4.2 ขอบเขตด้านเนื้อหาและเวลา.....	2
1.5 นิยามศัพท์เฉพาะ.....	4
1.6 สมมติฐานของการวิจัย.....	5
1.7 กรอบแนวคิดงานวิจัย.....	5
2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	7
2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง.....	7
2.1.1 หลักการโฟโตแกรมเมตรี.....	7
2.1.2 แนวคิดเรื่องการสำรวจ.....	12
2.1.3 แนวคิดเกี่ยวกับซอฟต์แวร์.....	14
2.1.4 แนวคิดเกี่ยวกับเทคนิควิธีการ.....	15
2.1.5 หลักการทางสถิติ.....	19
2.1.6 ลักษณะทางกายภาพของต้นมะนาว.....	19
2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง.....	20
3 วิธีดำเนินการวิจัย.....	23
3.1 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย.....	23
3.2 การเก็บรวบรวมข้อมูล.....	24
3.2.1 ข้อมูลภาพถ่ายทางอากาศจาก Google map.....	24
3.2.2 ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ.....	25

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## สารบัญ (ต่อ)

บทที่	หน้า
3.2.3 ข้อมูลภาคสนาม.....	27
3.3 การประมวลผลข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ.....	28
3.3.1 การประมวลผลภาพถ่ายทางอากาศด้วยโปรแกรม Pix4Dmapper.....	28
3.4 การวิเคราะห์ข้อมูล.....	33
3.4.1 การศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผลอัตโนมัติจาก ข้อมูล Google map.....	33
3.4.2 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติ ด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ.....	36
3.4.3 การตรวจสอบความถูกต้องของเทคนิคกับข้อมูลที่ได้นับจริงจาก การแปลภาพถ่ายด้วยสายตา.....	40
3.5 การเตรียมความพร้อม.....	40
3.5.1 การศึกษาอากาศยานไร้คนขับ.....	40
3.5.2 การศึกษาการประมวลผลข้อมูลภาพถ่ายทางอากาศจากอากาศ ยานไร้คนขับด้วยโปรแกรม Agisoft Photoscan และ Pix4Dmapper.....	42
3.5.3 การศึกษาเรื่อง Remote sensing ในโปรแกรม ArcMap.....	44
3.5.4 การศึกษาการใช้โปรแกรม MATLAB.....	45
4 ผลการวิจัย.....	46
4.1 การสร้างข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ.....	46
4.1.1 ผลการประมวลผลภาพถ่ายจากโปรแกรม Pix4Dmapper.....	46
4.2 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูล ภาพถ่ายดาวเทียมจาก Google map.....	49
4.2.1 วิธีการตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุด้วยข้อมูล ภาพถ่ายดาวเทียมจาก Google map.....	50
4.2.2 วิธีการตรวจนับจำนวนวัตถุจากค่าสีของวัตถุด้วยข้อมูลภาพถ่าย ดาวเทียมจาก Google map .....	50

## สารบัญ (ต่อ)

บทที่	หน้า
4.2.3	50
4.3	57
4.4	58
5	62
5.1	62
5.1.1	62
5.1.2	63
5.2	63
5.3	64
5.4	64
บรรณานุกรม	65
ภาคผนวก	68
ภาคผนวก ก	69
ภาคผนวก ข	74
ภาคผนวก ค	117
ประวัติผู้วิจัย	131

Copyright by Naresuan University

All rights reserved



## สารบัญตาราง

ตาราง	หน้า
3.1 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย.....	24
3.2 คุณสมบัติของอากาศยานไร้คนขับที่ทำการศึกษ.....	41
4.1 เพอร์เซ็นต์ความถูกต้องของการศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผล อัตโนมัติจากข้อมูล Google map.....	50
4.2 ผลการตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (ส้ม) ด้วยข้อมูลภาพถ่ายดาวเทียม จาก Google map.....	51
4.3 ผลการตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (ทุเรียน) ด้วยข้อมูลภาพถ่าย ดาวเทียมจาก Google map.....	52
4.4 ผลการตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (ส้ม) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map.....	53
4.5 ผลการตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (ทุเรียน) ด้วยข้อมูลภาพถ่ายดาวเทียม จาก Google map.....	54
4.6 ผลการตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (ส้ม) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map.....	55
4.7 ผลการตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (ทุเรียน) ด้วยข้อมูลภาพถ่ายดาวเทียม จาก Google map.....	56
4.8 เพอร์เซ็นต์ความถูกต้องของการศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผล อัตโนมัติจากข้อมูลภาพถ่ายทางอากาศด้วยอากาศยานไร้คนขับ.....	58

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## สารบัญภาพ

ภาพ	หน้า
1.1 แผนที่พื้นที่ศึกษา สวนมะนาว ตำบลสามพวง อำเภอคีรีมาศ จังหวัดสุโขทัย.....	3
1.2 แผนที่สวนมะนาว ตำบลท่าโพธิ์ อำเภอเมืองพิษณุโลก จังหวัดพิษณุโลก.....	3
1.3 กรอบแนวคิดงานวิจัย.....	6
2.1 ภาพถ่ายทางอากาศ: (a) ภาพตั้ง (b) ภาพเฉียงต่ำ และ (c) ภาพเฉียงสูง.....	8
2.2 ความสัมพันธ์ของขนาดพิกเซลของกล้องความสูงของ UAVs เหนือพื้นดินและความยาวโฟกัสของกล้องที่มีผลต่อ GSD .....	9
2.3 อากาศยานไร้คนขับชนิดปีกตรึง (Fixed Wing).....	10
2.4 อากาศยานไร้คนขับชนิดปีกหมุน (Multirotor).....	10
2.5 อากาศยานไร้คนขับชนิดปีกหมุนเดี่ยว (Single-Rotor Helicopter).....	11
2.6 อากาศยานไร้คนขับชนิดปีกตรึงชั้นลงแนวตั้ง (Fixed-Wing Hybrid).....	11
2.7 DJI Phantom 4 (DJI, 2018).....	13
2.8 Acer Nitro 5 AN515-51 (NITRO 5,2018).....	13
2.9 สมการเส้นตรงในรูปแบบเชิงขั้ว (Polar form) โดย $L$ คือ แนวเส้นของจุดต่างๆ ภายในรูป, $g$ คือ เวกเตอร์หน่วย (Unit Vector) ของแนวเส้น $L$ กับจุดอ้างอิง และ $P(x,y)$ คือ จุดต่างๆ ที่อยู่ในแนวเส้น $L$ .....	18
2.10 การหาวงกลมด้วยวิธีการ Hough Circle Transform: (ซ้าย) รูปภาพที่ต้องการหาวงกลมรัศมี $R$ ซึ่งมีจุดอยู่ 3 จุด และ (ขวา) วาดวงกลมรัศมี $R$ ใน Parametric space (accumulator array) โดยจุดที่ตัดกันมากที่สุดคือจุดที่มีค่า vote สูงสุด ซึ่งเป็นจุดศูนย์กลางของคำตอบ.....	18
2.11 การนับจำนวนต้นมะนาวอย่างอัตโนมัติโดยใช้ภาพจากอากาศยานไร้คนขับ (UAV)..	21
2.12 ระบบการแจกจ่ายต้นไม้ในแปลงปลูกการเกษตร.....	22
3.1 เครื่องมือที่ใช้ในการวิจัย.....	23
3.2 ข้อมูลภาพถ่ายทางอากาศจาก Google map: (a) สัม และ (b) ทุเรียน.....	25
3.3 ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ: (a) แอปพลิเคชัน DJI GS PRO, (b) การวางแผนแนวบินบนพื้นที่สวนมะนาวของจังหวัดสุโขทัย และ (c) การวางแผนแนวบินบนพื้นที่สวนมะนาวของจังหวัดพิษณุโลก.....	26
3.4 ขอบเขตสวนมะนาวและจำนวนต้นมะนาว: (a) สวนมะนาวของจังหวัดสุโขทัย และ (b) สวนมะนาวของจังหวัดพิษณุโลก.....	27
3.5 หน้าต่างสำหรับกำหนดค่า Project ในโปรแกรม Pix4Dmapper.....	28
3.6 การนำเข้าภาพถ่ายทางอากาศในโปรแกรม Pix4Dmappe.....	29
3.7 การกำหนดค่าพิกัดเป็น Auto Detected: WGS84/UTM Zone 47N.....	30
3.8 การกำหนดการประมวลผลเป็นแบบ 3D Maps .....	31

## สารบัญภาพ (ต่อ)

ภาพ	หน้า
3.9 การประมวลผลข้อมูลในโปรแกรม Pix4Dmappe.....	32
3.10 ขั้นตอนการประมวลผลข้อมูลในโปรแกรม Pix4Dmapper.....	32
3.11 สุ่มพื้นที่ตัวอย่างจากภาพถ่ายดาวเทียม Google map.....	33
3.12 การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุด้วยโปรแกรม MATLAB: (a) การเรียกภาพด้วยฟังก์ชัน imread, (b) การแปลงภาพสีเป็นภาพขาวดำด้วยฟังก์ชัน im2bw, (c) การหาขอบของวัตถุด้วยฟังก์ชัน canny, (d) การเพิ่มความหนาของเส้นขอบวัตถุด้วยฟังก์ชัน imdilate, (e) การเติมช่องว่างในวัตถุด้วยฟังก์ชัน holes, (f) การทดสอบการนับวัตถุ และ (g) การวนลูปเพื่อนับวัตถุ.....	34
3.13 การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุด้วยโปรแกรม MATLAB: (a) การเรียกภาพด้วยฟังก์ชัน imread, (b) การแสดงค่าสีของภาพด้วยฟังก์ชัน colorbar, (c) การกำหนดค่าสีตามวัตถุที่ต้องการนับ, (d) การแปลงภาพสีเป็นภาพขาวดำด้วยฟังก์ชัน im2bw, (e) การทดสอบการนับวัตถุ และ (f) การวนลูปเพื่อนับวัตถุ.....	35
3.14 การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุด้วยโปรแกรม MATLAB: (a) การเรียกภาพด้วยฟังก์ชัน imread, (b) การแปลงภาพสีเป็นภาพสีเทาด้วย rgb2gray และ (c) การกำหนดรัศมีต่ำสุด-สูงสุดของวัตถุ.....	36
3.15 การแปลงนามสกุลไฟล์ภาพจาก TFF เป็น PNG.....	37
3.16 ภาพถ่ายทางอากาศจากอากาศยานไร้คนขับทั้ง 4 ชั้นข้อมูล: (a) ภาพออร์โธสโตนมะนาวในจังหวัดสุโขทัย, (b) แบบจำลองพื้นผิวดิจิทัลสโตนมะนาวในจังหวัดสุโขทัย, (c) ภาพออร์โธสโตนมะนาวในจังหวัดพิษณุโลก และ (d) แบบจำลองพื้นผิวเชิงเลขสโตนมะนาวในจังหวัดพิษณุโลก.....	38
3.17 หน้าต่างแสดงผล GUI ในโปรแกรม MATLAB.....	39
3.18 ทดลองบังคับอากาศยานไร้คนขับ.....	42
3.19 การประมวลผลข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับด้วยโปรแกรม Agisoft Photoscan.....	43
3.20 การประมวลผลข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับด้วยโปรแกรม Pix4Dmapper.....	43
3.21 การจำแนกประเภทข้อมูลแบบควบคุม อำเภอคีรีมาศ จังหวัดสุโขทัย.....	44
3.22 การจำแนกประเภทข้อมูลแบบไม่ควบคุม อำเภอคีรีมาศ จังหวัดสุโขทัย.....	45
4.1 ภาพถ่ายจากอากาศยานไร้คนขับบนสโตนมะนาว ตำบลสามพวง อำเภอคีรีมาศ จังหวัดสุโขทัย.....	46
4.2 แบบจำลองความสูงพื้นผิวเชิงเลขของสโตนมะนาว ตำบลสามพวง อำเภอคีรีมาศ จังหวัดสุโขทัย.....	47

## สารบัญญภาพ (ต่อ)

ภาพ	หน้า
4.3 แผนที่ภาพถ่ายทางอากาศของสวนมะนาว ตำบลสามพวง อำเภอกีรีมาศ จังหวัดสุโขทัย.....	47
4.4 ภาพถ่ายจากอากาศยานไร้คนขับบนสวนมะนาว ตำบลท่าโพธิ์ อำเภอพิษณุโลก จังหวัดพิษณุโลก.....	48
4.5 แบบจำลองความสูงพื้นผิวเชิงเลขของสวนมะนาว ตำบลท่าโพธิ์ อำเภอพิษณุโลก จังหวัดพิษณุโลก.....	48
4.6 แผนที่ภาพถ่ายทางอากาศของสวนมะนาว ตำบลท่าโพธิ์ อำเภอพิษณุโลก จังหวัดพิษณุโลก.....	49
4.7 ภาพถ่ายทางอากาศจากอากาศยานไร้คนขับทั้ง 4 ชั้นข้อมูล: (a) ภาพออร์โธของสวนมะนาวในจังหวัดสุโขทัย, (b) แบบจำลองพื้นผิวดิจิตอลของสวนมะนาวในจังหวัดสุโขทัย, (c) ภาพออร์โธของสวนมะนาวในจังหวัดพิษณุโลก และ (d) แบบจำลองพื้นผิวเชิงเลขของสวนมะนาวในจังหวัดพิษณุโลก.....	57
4.8 GUI การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (RGB) สวนมะนาว จังหวัดสุโขทัย.....	59
4.9 GUI การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (DSM) สวนมะนาว จังหวัดสุโขทัย.....	59
4.10 GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (RGB) สวนมะนาว จังหวัดสุโขทัย....	60
4.11 GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (DSM) สวนมะนาว จังหวัดสุโขทัย...	60
4.12 GUI การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (RGB) สวนมะนาว จังหวัดสุโขทัย...	61
4.13 GUI การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (DSM) สวนมะนาว จังหวัดสุโขทัย..	61

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved



# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

อากาศยานไร้คนขับ (Unmanned Aerial Vehicle: UAV) เป็นเทคโนโลยีสมัยใหม่ที่กำลังได้รับความนิยมในงานด้านสำรวจ โดยสามารถบังคับด้วยสัญญาณวิทยุ ซึ่งมีระบบควบคุมและการสั่งการบินทั้งแบบอัตโนมัติและแบบกึ่งอัตโนมัติ สามารถควบคุมจากระยะไกลแทนนักบินจริง ในอดีตการนำอากาศยานไร้คนขับมาใช้ประโยชน์มักอยู่ในแวดวงที่จำกัด โดยมีการใช้ประโยชน์ในทางการทหารเท่านั้นเพราะมีค่าใช้จ่ายสูง แต่ในปัจจุบันเทคโนโลยีอากาศยานไร้คนขับได้มีการพัฒนาให้มีประสิทธิภาพมากขึ้นและมีราคาที่ถูกลง ทำให้มีการประยุกต์ใช้อากาศยานไร้คนขับในกิจกรรมต่างๆ อย่างหลากหลายมากยิ่งขึ้น (ธราววุฒิ บุญเหลือ, 2556) เช่น การวางแผนเพื่อพัฒนาเมืองและสภาพแวดล้อม การสำรวจทรัพยากรธรรมชาติ การสำรวจเพื่อวางแผนแก้ไขปัญหาอุตสาหกรรมและการเกษตรกรรม เป็นต้น นอกจากนี้ภาพถ่ายจากอากาศยานไร้คนขับยังสามารถสร้างผลิตภัณฑ์แบบจำลองความสูงด้วยหลักการทางโฟโตแกรมเมตรี ซึ่งสามารถนำไปประยุกต์ใช้ในการประมาณความค่าสูงของวัตถุ รวมถึงเป็นปัจจัยประกอบการตรวจนับจำนวนของวัตถุ ต้นไม้ อาคาร สิ่งปลูกสร้างได้

จำนวนของต้นไม้ที่มีอยู่ในแปลงเพาะปลูกเป็นข้อมูลสำคัญที่ใช้ในการจัดการด้านเกษตรกรรม เนื่องจากสามารถเป็นข้อมูลพื้นฐานที่ทำให้ทราบถึงจำนวนผลผลิตต่อพื้นที่ ในปัจจุบันมีเทคโนโลยีที่แพร่หลายในการนำมาใช้ตรวจนับจำนวนต้นไม้ ซึ่งส่วนใหญ่ใช้ข้อมูลภาพที่มาจากหลายแหล่ง ได้แก่ ภาพดาวเทียม ภาพถ่ายทางอากาศ และภาพจากอากาศยานไร้คนขับ เนื่องจากสามารถมองเห็นภาพรวมของพื้นที่ขนาดใหญ่ได้โดยไม่ต้องเข้าไปสำรวจในพื้นที่จริง (Gistda Academy, 2558) อีกทั้งยังช่วยประหยัดเวลาในการนับด้วยแรงงานซึ่งทำให้เสียเวลาและงบประมาณเป็นอย่างมาก โดยเฉพาะพืชสวนที่มีการปลูกพื้นจำนวนมาก เช่น สวนมะนาว สวนปาล์ม สวนทุเรียน เป็นต้น นอกจากนี้การเข้าสำรวจในบางพื้นที่มักพบว่าต้นไม้ขึ้นตามแนวทางเดิน ซึ่งทำให้ยากต่อการเข้าถึง อย่างไรก็ตามในปัจจุบันยังใช้การตรวจนับด้วยแรงงานคนเพราะยังไม่มีเทคโนโลยีที่เหมาะสมและถูกต้องในการตรวจนับโดยอัตโนมัติ จึงทำให้การตรวจนับจำนวนต้นไม้มีความยุ่งยากและใช้เวลาค่อนข้างมาก ทำให้เทคโนโลยีการสำรวจด้วยภาพจากอากาศยานไร้คนขับเป็นทางเลือกหนึ่งที่ถูกนำมาใช้ประโยชน์ ซึ่งหากมีการนำเทคโนโลยีการประมวลผลภาพมาพัฒนาเพื่อสร้างอัลกอริทึมในการตรวจนับแบบอัตโนมัติบนข้อมูลภาพถ่ายทางอากาศจะทำให้ประหยัดต้นทุนในการประมาณผลผลิตได้อย่างมหาศาล

การศึกษาค้นคว้าวิจัยมุ่งศึกษาและพัฒนาอัลกอริทึมในการตรวจนับจำนวนไม้ผลแบบอัตโนมัติ โดยประยุกต์ใช้ข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับที่บูรณาการระหว่างกระบวนการโฟโตแกรมเมตรี และเทคโนโลยีการประมวลผลภาพเชิงเลข รวมทั้งได้พัฒนา Graphic User Interface (GUI) เพื่อให้ง่ายต่อผู้ใช้งาน (Juthawut Chantharamalee, 2016)



## 1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและพัฒนาอัลกอริทึมในตรวจนับจำนวนไม้ผลแบบอัตโนมัติ
2. เพื่อประยุกต์ใช้ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับกับอัลกอริทึมในการตรวจนับจำนวนไม้ผลแบบอัตโนมัติ

## 1.3 ความสำคัญของการวิจัย

การตรวจนับจำนวนต้นไม้สามารถนำมาใช้ในการประมาณจำนวนผลผลิตต่อพื้นที่ เทคโนโลยีที่สามารถนำมาใช้ในการตรวจนับจำนวนต้นไม้คือ ภาพจากดาวเทียม ภาพถ่ายทางอากาศ และภาพจากอากาศยานไร้คนขับ เนื่องจากคุณลักษณะที่ทำให้สามารถมองเห็นภาพรวมของพื้นที่ขนาดใหญ่ได้โดยไม่ต้องเข้าไปสำรวจในพื้นที่จริง (Gistda Academy, 2558) และการลงพื้นที่สำรวจเพื่อตรวจนับจำนวนต้นไม้จำเป็นต้องทำการนับด้วยแรงงานคน จึงทำให้มีความยุ่งยากและใช้เวลาค่อนข้างมาก เพราะมีการปลูกเป็นจำนวนมาก อีกทั้งบางพื้นที่มีพืชชนิดอื่นๆ ขึ้นตามแนวทางเดินทำให้ยากต่อการเข้าถึง ดังนั้นการพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ จึงเป็นแนวทางหนึ่งในการประหยัดต้นทุนการสำรวจได้

## 1.4 ขอบเขตของการวิจัย

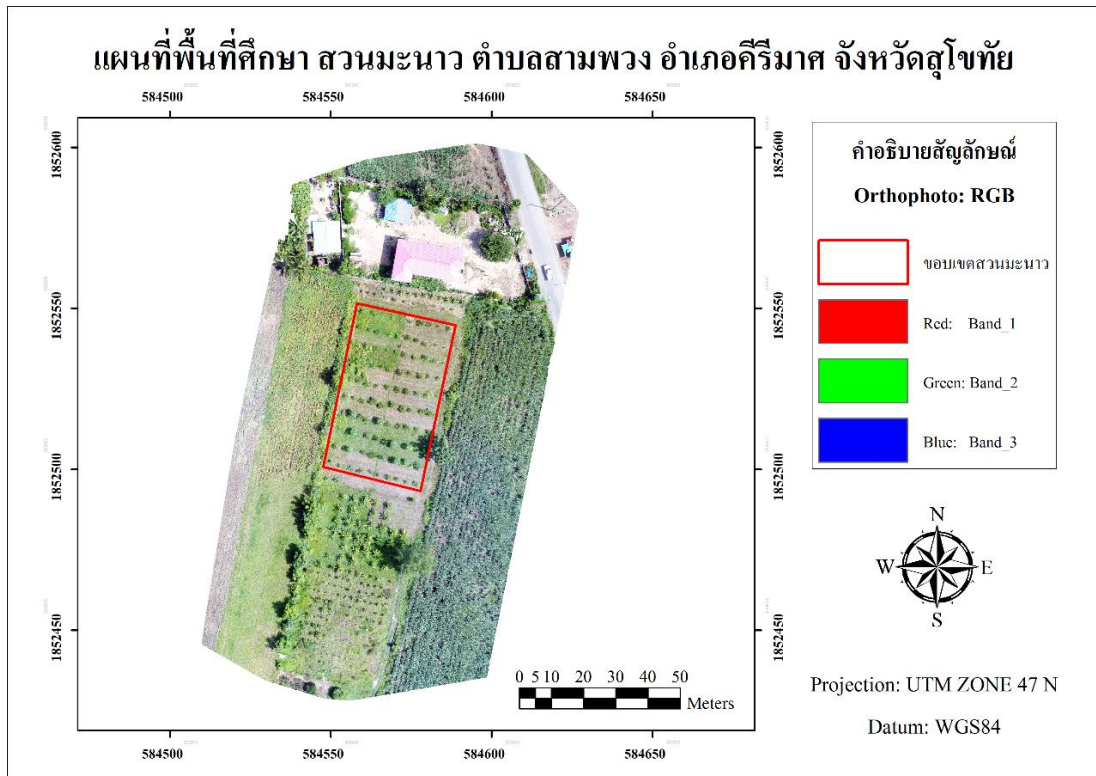
### 1.4.1 ขอบเขตพื้นที่ศึกษา

การศึกษาครั้งนี้มีพื้นที่ศึกษาจำนวน 2 แห่ง คือ

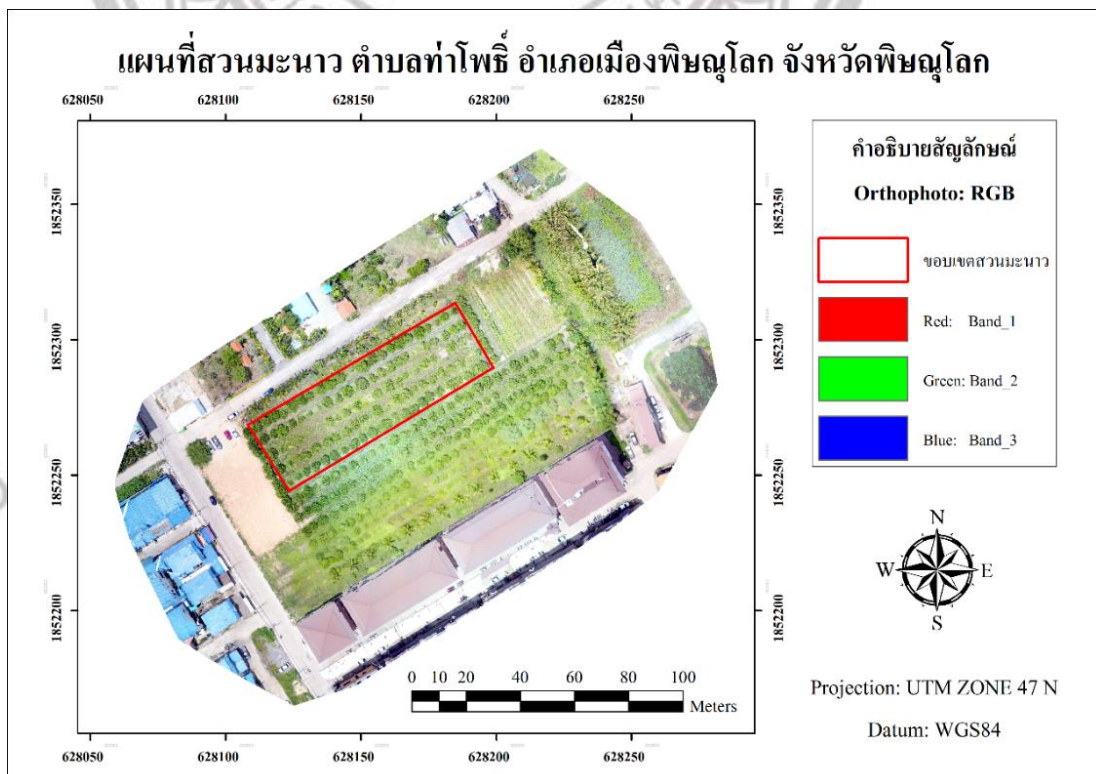
1. สวนมะนาว ตำบลสามพวง อำเภอศรีมหาโพธิ์ จังหวัดสุโขทัย โดยมีตำแหน่งทางภูมิศาสตร์ คือ ละติจูดที่ 16.754133 และลองจิจูดที่ 99.793430 มีพื้นที่ประมาณ 1 ไร่ อายุต้นมะนาวประมาณ 8 เดือน และมีต้นมะนาวจำนวน 94 ต้น ดังภาพ 1.1
2. สวนมะนาว ตำบลท่าโพธิ์ อำเภอเมืองพิษณุโลก จังหวัดพิษณุโลก โดยมีตำแหน่งทางภูมิศาสตร์ คือ ละติจูดที่ 16.750004 และลองจิจูดที่ 100.202362 มีพื้นที่ประมาณ 1 ไร่ อายุต้นมะนาวประมาณ 4 ปี และมีต้นมะนาวจำนวน 102 ต้น ดังภาพ 1.2

### 1.4.2 ขอบเขตด้านเนื้อหาและเวลา

ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ ยี่ห้อ DJI Phantom 4 Advanced โดยทำการบินถ่ายภาพสวนมะนาว จังหวัดสุโขทัย ในวันที่ 13 กันยายน พ.ศ. 2662 ตั้งแต่เวลา 10.00 - 10.18 น. และทำการบินถ่ายภาพสวนมะนาว จังหวัดพิษณุโลก ในวันที่ 16 กันยายน พ.ศ. 2662 ตั้งแต่เวลา 12.51 - 13.10 น. โดยทั้ง 2 วันที่ได้ทำการบินถ่ายภาพสภาพอากาศดี ไม่มีเมฆมารบกวน และมีแสงแดดที่แรง



ภาพ 1.1 แผนที่พื้นที่ศึกษา สวนมะนาว ตำบลสามพวง อำเภอกีรีมาศ จังหวัดสุโขทัย



ภาพ 1.2 แผนที่สวนมะนาว ตำบลท่าโพธิ์ อำเภอเมืองพิชัยโลก จังหวัดพิชัยโลก

### 1.5 นิยามศัพท์เฉพาะ

**การประมวลผลภาพ (Image Processing)** คือ การประยุกต์ใช้งานการประมวลผลบนสัญญาณ 2 มิติ เช่น ภาพนิ่ง (ภาพถ่าย) หรือภาพวีดิทัศน์ (วิดีโอ) และยังรวมถึงสัญญาณ 2 มิติอื่นๆ ที่ไม่ใช่ภาพ

**การประมวลผลภาพดิจิทัล (Digital Image Processing)** เป็นการกล่าวถึงเทคนิคและอัลกอริทึมต่างๆ ที่ใช้การประมวลผลภาพที่อยู่ในรูปแบบดิจิทัล (ภาพดิจิทัล) ภาพในที่นี้หมายถึงรวมถึงสัญญาณดิจิทัลใน 2 มิติ อื่นๆ โดยทั่วไปด้วย การประมวลผลภาพดิจิทัลเมื่อใช้อย่างกว้างๆ จะครอบคลุมถึงสัญญาณวิดีโอ (Video) หรือภาพเคลื่อนไหว ซึ่งจะเป็นชุดของภาพนิ่ง เรียกว่า เฟรม (Frame) หลายๆ ภาพต่อกันไปตามเวลา ซึ่งก็คือสัญญาณ 3 มิติ เมื่อนับเวลาเป็นมิติที่ 3 หรืออาจจะครอบคลุมถึงสัญญาณ 3 มิติ อื่นๆ เช่น ภาพ 3 มิติทางการแพทย์ หรืออาจจะมากกว่านั้น เช่น ภาพ 3 มิติ และหลายชนิด (Multimodal image)

**แบบจำลองพื้นผิวดิจิทัล (Digital Surface Model: DSM)** คือ การสร้างแบบจำลองความสูงของภูมิประเทศ และจัดเก็บข้อมูลให้อยู่ในรูปแบบตารางกริด หรือข้อมูลราสเตอร์ โดยรวมความสูงของสิ่งปกคลุมพื้นผิวทางกายภาพของโลก เช่น สิ่งปลูกสร้าง ต้นไม้ และพุ่มไม้ เป็นต้น (วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, 2561)

**ภาพออร์โธ (Orthophoto)** คือ ภาพของภูมิประเทศที่สร้างจากคู่ภาพที่ซ้อนทับกัน โดยมีการขจัดลักษณะเรขาคณิตแบบการฉายจากจุดศูนย์ทิวทัศน์ของภาพออกไป ภาพออร์โธสามารถนำมาใช้ได้ในลักษณะเดียวกับแผนที่ทางราบ (Planimetric Map) เนื่องจากมีมาตราส่วนคงที่ และหากนำเส้นชั้นความสูงมาซ้อนทับบนภาพออร์โธ จะทำให้ได้แผนที่ภาพออร์โธที่สามารถนำมาใช้เช่นเดียวกับแผนที่ภูมิประเทศ (Topographic Map) (วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, 2561)

**อากาศยานไร้คนขับ หรือยูเอวี (Unmanned Aerial Vehicle: UAV)** เป็นอากาศยานที่ไม่มีนักบินประจำการอยู่บนเครื่อง ไร้คนขับหรือนักบิน แต่สามารถควบคุมได้ อากาศยานไร้คนขับมีรูปร่าง ขนาด รูปแบบ และเอกลักษณ์ที่แตกต่างกันออกไป ตามหลักแล้วอากาศยานไร้คนขับก็คือโดรน (Drone) ซึ่งเป็นอากาศยานที่ควบคุมจากระยะไกล ใช้การควบคุมอัตโนมัติที่มีอยู่ 2 ลักษณะ คือ การควบคุมจากระยะไกล และการควบคุมแบบอัตโนมัติโดยใช้ระบบการบินด้วยตนเอง ซึ่งต้องอาศัยโปรแกรมคอมพิวเตอร์ที่มีระบบที่ซับซ้อนแล้วมีการติดตั้งไว้ในอากาศยาน โดยอาจกล่าวได้ว่า อากาศยานไร้คนขับ คือ เครื่องบินที่สามารถบินได้ด้วยระบบอัตโนมัติ โดยไม่ต้องใช้นักบินประจำการอยู่บนอากาศยาน (ประสพชัย ศีลาอ่อน, 2014)



### 1.6 สมมติฐานของการวิจัย

1. เทคนิคการตรวจนับจำนวนวัตถุจากระยะของวัตถุมีประสิทธิภาพในการตรวจนับจำนวนไม้ผลมากที่สุด
2. การใช้แบบจำลองพื้นผิวดิจิทัลในการตรวจนับจำนวนไม้ผลอัตโนมัติในแปลงปลูกการเกษตรมีความถูกต้องของจำนวนไม้ผลมากกว่าการใช้ภาพออร์โธ

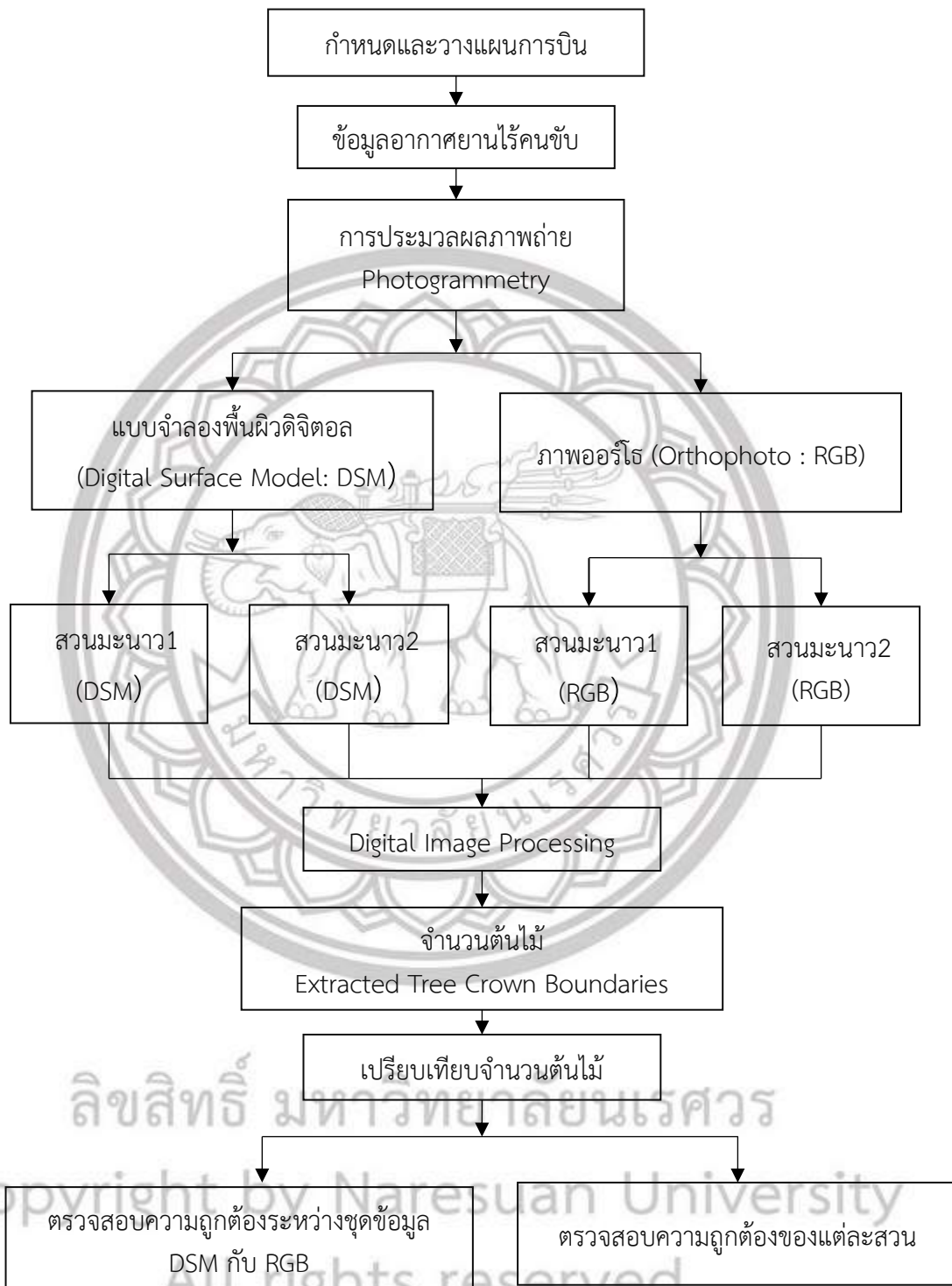
### 1.7 กรอบแนวคิดงานวิจัย

ผู้วิจัยได้ออกแบบกรอบแนวคิดงานวิจัยเพื่อใช้ในการพัฒนาอัลกอริทึมดังภาพ 1.3 ซึ่งได้มาจากการทดลองเบื้องต้นด้วยภาพจาก Google map เพื่อให้ทราบถึงแนวทางการพัฒนาโค้ดใน MATLAB ที่มุ่งพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนต้นของไม้ผลจากภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ โดยนำอัลกอริทึมที่พัฒนาขึ้นนี้นำมาทดสอบกับสวนมะนาวด้วยข้อมูลทั้งแบบจำลองพื้นผิวเชิงเลขและภาพออร์โธ จากนั้นทำการตรวจสอบความถูกต้องด้วยข้อมูลจากการแปลด้วยสายตา และข้อมูลจากการสำรวจภาคสนาม

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved



ภาพ 1.3 กรอบแนวคิดงานวิจัย



## บทที่ 2

### เอกสารและงานวิจัยที่เกี่ยวข้อง

จากการทบทวนเอกสารและงานวิจัยที่เกี่ยวข้อง เพื่อเป็นแนวคิดในการศึกษางานวิจัยเรื่อง การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ ผู้วิจัยได้ทำการรวบรวมหลักการ แนวคิด ทฤษฎี เอกสารและงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดของประเด็นต่างๆ ดังนี้

#### 2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

2.1.1 หลักการโฟโตแกรมเมตรี

2.1.2 แนวคิดเรื่องการสำรวจ

2.1.3 แนวคิดเกี่ยวกับซอฟต์แวร์

2.1.4 แนวคิดเกี่ยวกับเทคนิควิธีการ

2.1.5 หลักการทางสถิติ

2.1.6 ลักษณะทางกายภาพของต้นมะนาว

#### 2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

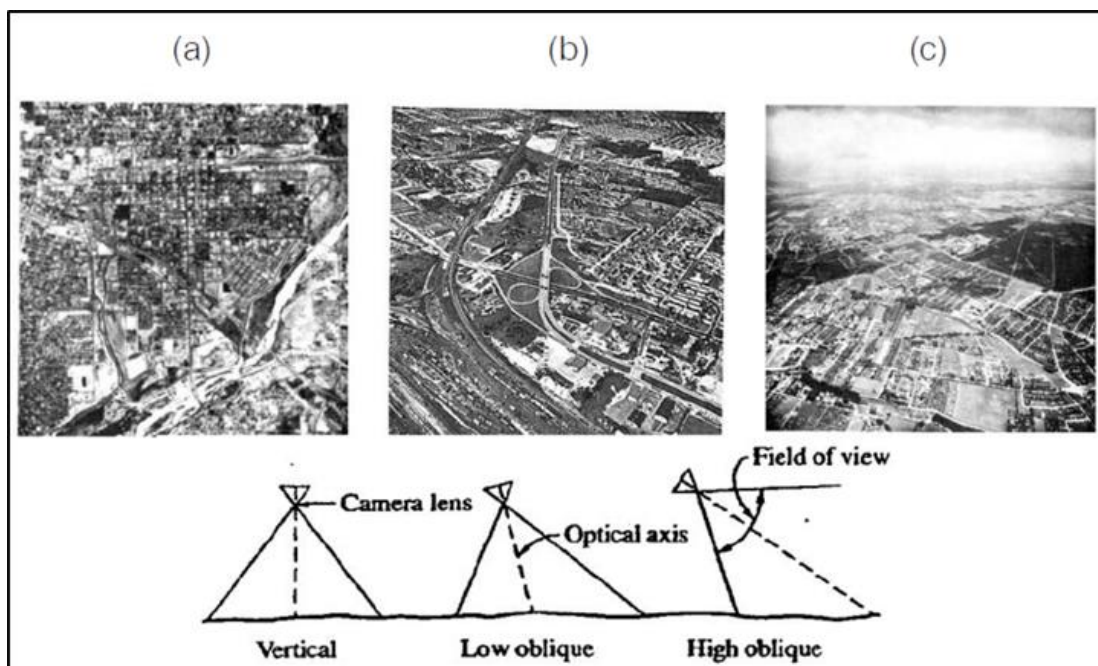
#### 2.1 แนวคิดทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 หลักการโฟโตแกรมเมตรี

##### 1. ภาพถ่ายทางอากาศ

ภาพถ่ายทางอากาศแบ่งออกเป็นภาพตั้งกับภาพเฉียง โดยภาพตั้ง (Vertical) ดังภาพ 2.1(a) คือ ภาพที่ถ่ายโดยให้แกนของกล้องอยู่ในแนวตั้ง ถ้าแกนของกล้องขณะที่ถ่ายอยู่ในแนวตั้ง ระนาบภาพจะขนานกับระนาบราบของพื้นหลักฐาน ภาพที่ได้จะเรียกว่า “ตั้งจริง (Truly Vertical)” ซึ่งการถ่ายภาพตั้งอาจจะมีอิทธิพลของสภาพอากาศทำให้เครื่องบินมีการเอียงในขณะที่ถ่ายภาพ ถ้าแกนของกล้องเอียงไปจากแนวตั้งเพียงเล็กน้อย ภาพที่ได้เรียกว่า “ภาพเอียง” ในงานส่วนใหญ่ความเอียงของแกนกล้องจะน้อยกว่า 1 องศา และไม่เกิน 3 องศา ส่วนภาพเฉียง (Oblique) คือ ภาพที่ถ่ายโดยให้แกนของกล้องเอียงออก ซึ่งแบ่งเป็นภาพเฉียงต่ำ (Low Oblique) ดังภาพ 2.1(b) จะไม่ปรากฏเส้นขอบฟ้า และภาพเอียงสูง (High Oblique) ดังภาพ 2.1(c) จะปรากฏแนวเส้นขอบฟ้า (กิตติศักดิ์ ศรีกลาง, 2559

All rights reserved

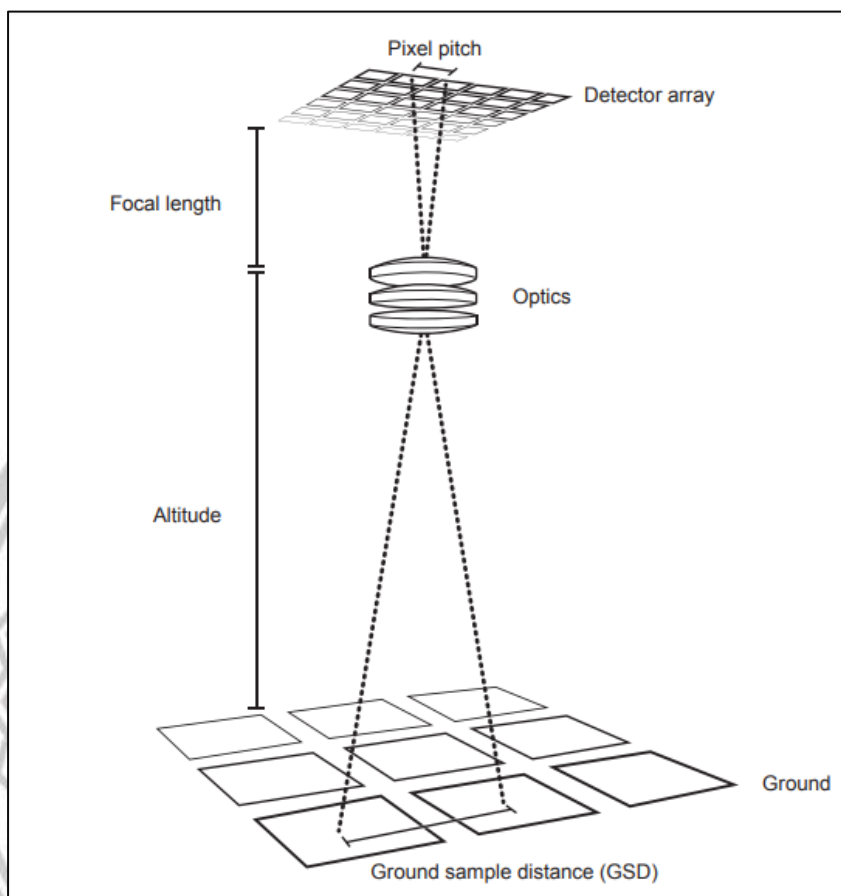


ภาพ 2.1 ภาพถ่ายทางอากาศ: (a) ภาพตั้ง (b) ภาพเฉียงต่ำ และ (c) ภาพเฉียงสูง  
ที่มา: กิตติศักดิ์ ศรีกลาง (2559)

## 2. การวางแผนการบิน

การวางแผนการบินที่ออกแบบอย่างถูกต้องเป็นสิ่งสำคัญสำหรับการสำรวจโฟโตแกรม พิกัดของแต่ละภาพควรได้รับการกำหนดอย่างแม่นยำเพื่อให้ได้พื้นที่ที่ต้องการ ขนาดตัวอย่าง และซ้อนทับกันระหว่างภาพถ่าย ความต้องการของสเตอริโอโฟโตมิเตอร์ คือ ทุกจุดบนพื้นดินจะถูกถ่ายภาพอย่างน้อยสองรูปแบบเมตริกด้วยการทับซ้อนกัน 50 เปอร์เซ็นต์ แต่ด้วยลักษณะของโลกไม่ได้เป็นพื้นราบและอาจมีอิทธิพลภายนอกมาส่งผลกระทบต่อเครื่องบินทำให้การซ้อนทับกัน 50 เปอร์เซ็นต์ไม่เพียงพอ เพื่อให้มั่นใจว่าเหมาะสมควรทับซ้อนกัน 60 เปอร์เซ็นต์ และ 25-30 เปอร์เซ็นต์ สำหรับการซ้อนทับด้านข้าง โดยการตรวจจับภาพด้วย UAV จำเป็นต้องซ้อนทับกันถึง 80 เปอร์เซ็นต์ เพื่อให้ได้ผลลัพธ์ที่ถูกต้องและเชื่อถือได้ ข้อควรพิจารณาในขณะที่ทำแผนการบินคือความละเอียดที่ต้องการบนพื้นดิน (Ground Sample Distance, GSD) โดย GSD คือ ระยะทางบนพื้นซึ่งหมายถึงหนึ่งพิกเซลบนรูปภาพ ค่า GSD ยิ่งน้อยความละเอียดเชิงพื้นที่ของภาพถ่ายและรายละเอียดที่มองเห็นได้จะชัดเจนยิ่งขึ้น ซึ่ง GSD ขึ้นอยู่กับขนาดพิกเซลของกล้องความสูงของ UAVs เหนือพื้นดิน และความยาวโฟกัสของกล้อง โดย GSD สามารถคำนวณได้จากสมการ (1)

$$GSD = (\text{pixel size} \times \text{height above ground level}) / \text{focal length} \quad (1)$$



ภาพ 2.2 ความสัมพันธ์ของขนาดพิกเซลของกล้องความสูงของ UAVs เหนือพื้นดินและความยาวโฟกัสของกล้องที่มีผลต่อ GSD

ที่มา: (Greenwood, 2015)

### 3. ประเภทของอากาศยานไร้คนขับสำหรับงานสำรวจด้วยภาพถ่าย

วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์ (2561) ได้อธิบายว่า อากาศยานไร้คนขับมีการแบ่งประเภทที่สามารถกำหนดรูปแบบการจัดได้หลายลักษณะ ทั้งนี้ขึ้นอยู่กับจุดประสงค์ในการนำไปใช้ คุณลักษณะเฉพาะของอากาศยานไร้คนขับถูกพัฒนาขึ้นสำหรับใช้งานในภารกิจใดภารกิจหนึ่งหรือสภาวะของภูมิประเทศในการนำไปใช้ นอกจากนี้ข้อพิจารณาดังกล่าวจะต้องคำนึงว่าอากาศยานไร้คนขับดังกล่าวนำไปใช้เป็นองค์กรใด และมีการใช้เพื่อจุดประสงค์หรือเหตุผลใด โดยแบ่งประเภทของอากาศยานไร้คนขับสำหรับงานสำรวจด้วยภาพถ่ายเป็น 4 ประเภท ดังนี้

1) อากาศยานไร้คนขับชนิดปีกตรึง (Fixed Wing) เป็นอากาศยานที่มีลักษณะคล้ายกับเครื่องบินทั่วไป ใช้ระยะเวลาการบินประมาณ 30-60 นาที สามารถบินครอบคลุมพื้นที่ได้มากกว่าอากาศยานไร้คนขับแบบปีกหมุน ในการลงจอดจะต้องอาศัยพื้นที่โล่งกว้างพอสมควร ดังภาพ 2.3

2) อากาศยานไร้คนขับชนิดปีกหมุน (Multirotor) เป็นอากาศยานขึ้นลงแนวดิ่งอาศัยการหมุนของใบพัดในการขึ้นลงและขับเคลื่อนไปในทิศทางต่างๆ ประกอบด้วยใบพัดจำนวน 3, 4, 6 และ 8 ใบพัด โดยทั่วไปใช้ระยะเวลาการบินประมาณ 10-20 นาที ดังภาพ 2.4

3) อากาศยานไร้คนขับชนิดปีกหมุนเดี่ยว (Single-Rotor Helicopter) เป็นอากาศยานที่มีรูปร่างและโครงสร้างคล้ายเฮลิคอปเตอร์ ไม่เหมือนอากาศยานไร้คนขับชนิดปีกหมุน มีใบพัดขนาดใหญ่เพียงใบเดียวที่ใช้ในการเคลื่อนที่และมีใบพัดขนาดเล็กอยู่บนหางของอากาศยานเพื่อควบคุมทิศทางการบิน ซึ่งจะมีประสิทธิภาพมากกว่าอากาศยานไร้คนขับชนิดปีกหมุนบางรุ่น ดังภาพ 2.5

4) อากาศยานไร้คนขับชนิดปีกตรึงขึ้นลงแนวตั้ง (Fixed-Wing Hybrid) เป็นอากาศยานที่ได้รับการพัฒนาขึ้นมาใหม่ มีลำตัวอากาศยานเป็นแบบอากาศยานไร้คนขับชนิดปีกตรึง แต่สามารถขึ้นลงแนวตั้งได้ ซึ่งเป็นการนำเอาข้อดีของประเภทอากาศยานไร้คนขับชนิดปีกตรึงมารวมกับอากาศยานไร้คนขับชนิดปีกหมุน ดังภาพ 2.6



ภาพ 2.3 อากาศยานไร้คนขับชนิดปีกตรึง (Fixed Wing)



ภาพ 2.4 อากาศยานไร้คนขับชนิดปีกหมุน (Multirotor)





ภาพ 2.5 อากาศยานไร้คนขับชนิดปีกหมุนเดี่ยว (Single-Rotor Helicopter)



ภาพ 2.6 อากาศยานไร้คนขับชนิดปีกตรึงขึ้นลงแนวตั้ง (Fixed-Wing Hybrid)

#### 4. ระดับความสูง

ระดับความสูงเป็นสิ่งสำคัญที่ควรพิจารณาเมื่อทำการบินเพื่อทำแผนที่จาก UAV ทั้งเพื่อการปฏิบัติและในความสนใจของการเล่นที่ปลอดภัยและถูกต้องตามกฎหมาย ยิ่งระดับความสูงเพิ่มมากขึ้น จะส่งผลให้ความละเอียดต่ำลง การถ่ายภาพในระดับสูงยังสามารถช่วยลดการบิดเบือนที่พบได้ในภาพของอาคารและวัตถุอื่นๆ บนพื้นดิน ในขณะที่ระดับความสูงต่ำจะลดค่า GSD และเพิ่มคุณภาพของภาพ (วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, 2561)

#### 5. จุดควบคุมภาคพื้นดิน ( Ground Control Point : GCP )

วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์ (2561) อธิบายว่าจุดควบคุมภาคพื้นดิน (Ground Control Point: GCP) หมายถึง จุดใดๆ ที่ทราบค่าพิกัดในระบบพิกัดภูมิประเทศ เป็นตัวกลางที่ทำให้สามารถจัดภาพให้มีความสัมพันธ์อ้างอิงกับพื้นภูมิประเทศ สามารถแบ่ง



ออกเป็นจุดควบคุมทางราบ (Horizontal Control Point) และจุดควบคุมทางตั้ง (Vertical Control Point) ซึ่งระบบพิกัดอ้างอิงในระบบพิกัดภูมิประเทศที่ใช้โดยทั่วไป มีดังนี้

- พิกัดทางราบ ได้แก่ ระบบพิกัดภูมิศาสตร์ (ละติจูด ลองจิจูด) และระบบพิกัดกริด UTM Easting, Northing (N, E)
- พิกัดทางตั้ง ได้แก่ ระดับสูงเหนือพื้นอ้างอิง เช่น ระดับทะเลปานกลาง (Mean Sea Level: MSL)

#### 6. แบบจำลองพื้นผิวดิจิทัล (Digital Surface Model: DSM)

วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์ (2561) อธิบายว่าแบบจำลองพื้นผิวดิจิทัล (Digital Surface Model: DSM) คือ การสร้างแบบจำลองความสูงของภูมิประเทศ และจัดเก็บข้อมูลให้อยู่ในรูปแบบตารางกริดหรือข้อมูลราสเตอร์ โดยรวมความสูงของสิ่งปกคลุมพื้นผิวทางกายภาพของโลกด้วย เช่น สิ่งปลูกสร้าง ต้นไม้ และพุ่มไม้ เป็นต้น

#### 7. ภาพออร์โธ (Orthophoto)

ภาพออร์โธ (Orthophoto) คือ ภาพของภูมิประเทศที่สร้างจากคู่ภาพที่ซ้อนทับกันโดยมีการขจัดลักษณะเรขาคณิตแบบการฉายจากจุดศูนย์ทิวทัศน์ของภาพออกไป ภาพออร์โธสามารถนำมาใช้ได้ ในลักษณะเดียวกับแผนที่ทางราบ (Planimetric Map) เนื่องจากมีมาตราส่วนคงที่ และถ้าหากนำเส้นชั้นความสูงมาซ้อนทับบนภาพออร์โธ จะได้เป็นแผนที่ภาพออร์โธที่สามารถนำมาใช้ได้เช่นเดียวกับแผนที่ภูมิประเทศ (Topographic Map) (วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, 2561)

### 2.1.2 แนวคิดเรื่องการสำรวจ

#### 1. เครื่องมือและอุปกรณ์ในการตรวจนับไม้ผล

- แผนที่สภาพภูมิประเทศบริเวณพื้นที่ที่ทำการศึกษา ภาพถ่ายดาวเทียม หรือภาพถ่ายทางอากาศ
- เครื่องจับพิกัดทางภูมิศาสตร์ (GPS) และเข็มทิศ (Compass)
- อุปกรณ์อื่นๆ ได้แก่ กล้องถ่ายภาพ สมุดจดบันทึก และเครื่องเขียน

#### 2. อากาศยานไร้คนขับ DJI Phantom 4

คุณสมบัติ:

- น้ำหนัก (รวมแบตเตอรี่และใบพัด) 1,380 กรัม
- ช่วงความถูกต้องของ Hover แนวตั้ง  $\pm 0.1$  เมตร (พร้อมด้วยตำแหน่งวิสัยทัศน์) และ  $\pm 0.5$  เมตร (พร้อมตำแหน่ง GPS) แนวนอน  $\pm 0.3$  เมตร (ด้วยตำแหน่งวิสัยทัศน์) และ  $\pm 1.5$  เมตร (พร้อมตำแหน่ง GPS)
- ระบบระบุตำแหน่งดาวเทียม: GPS/GLONASS
- ความละเอียดวิดีโอ UHD: 4,096x2,160 พิกเซล 24/25, 3,840x2,160 พิกเซล 24/25/30 FHD: 1,920x1,080 พิกเซล 24/25/30/48/50/60
- ความละเอียดภาพนิ่ง: 20 Mega-Pixel
- ส่งสัญญาณภาพ/ข้อมูลการบิน และปริมาณแบตเตอรี่
- ขากล่องกิมบอล 3 แกน (ไม่สามารถถอดก้างได้)
- ความเร็วขั้วเตอร์อิเล็กทรอนิกส์ 8-1/8,000 วินาที

- ขนาดรูปภาพ 4,000 × 3,000
- ความจุสูงสุด 64 กิกะไบต์
- แบตเตอรี่ Lipo ความจุ 5,350 มิลลิแอมแปร์ 15.2 โวลต์ (บินได้นานสูงสุด 20-25 นาที) ชาร์จแบตเตอรี่ประมาณ 1 ชั่วโมง
- ควบคุมและรับภาพได้ไกลรอบตัวสูงสุด 2-7 กิโลเมตร ผ่านความถี่ 2.4 จิกะเฮิรตซ์ และ 5.8 จิกะเฮิรตซ์
- วิทยุควบคุมได้ทั้งโทรนและกล่องรับสัญญาณภาพซัดระดับ HD สู่ Smart Phone และ Tablet (iOS/Android) ด้วยสาย USB



ภาพ 2.7 DJI Phantom 4 (DJI, 2018)

### 3. คอมพิวเตอร์โน้ตบุ๊ก Acer Nitro 5 AN515-51

คุณสมบัติ:

- Intel Core i7-7700HQ (2.80 - 3.80 จิกะเฮิรตซ์)
- NVIDIA GeForce GTX 1050Ti (4 กิกะไบต์ GDDR5)
- 8 กิกะไบต์ DDR4
- 1 TB 5400 RPM + 128 กิกะไบต์ SSD
- 15.6 นิ้ว (1,920x1,080) Full HD IPS



ภาพ 2.8 Acer Nitro 5 AN515-51 (NITRO 5,2018)

### 2.1.3 แนวคิดเกี่ยวกับซอฟต์แวร์

#### 1. Agisoft Photoscan

**Agisoft PhotoScan** เป็นโปรแกรมในการสร้างแบบจำลอง 3 มิติขั้นสูงที่ใช้ภาพเพื่อสร้างแบบจำลอง 3 มิติ ที่มีคุณภาพจากภาพถ่ายอากาศยานไร้คนขับ เป็นเทคโนโลยีที่มีประสิทธิภาพทั้งในสภาพที่มีจุดควบคุมภาคพื้นดินและไม่มีจุดควบคุมภาคพื้นดิน ภาพถ่ายนั้นสามารถนำมาทำภาพ 3 มิติ โดยวัตถุที่จะสร้างนั้นจะต้องปรากฏในภาพถ่ายอย่างน้อยสองรูป การจัดตำแหน่งภาพและการสร้างโมเดล 3D เป็นไปอย่างอัตโนมัติ โดยการประมวลผลภาพด้วย Agisoft PhotoScan ประกอบด้วยขั้นตอนหลักดังนี้

- (1) การนำเข้าภาพถ่ายลงใน Agisoft PhotoScan โดยก่อนที่จะเริ่มดำเนินการจำเป็นต้องระบุว่าจะใช้ภาพแบบใด ซึ่งไม่ต้องนำเข้าภาพใน Agisoft PhotoScan หากไม่มีความจำเป็น
- (2) การตรวจสอบภาพ โดยเลือกเฉพาะภาพที่จำเป็นต่อจุดมุ่งหมายของผู้ปฏิบัติ
- (3) การจัดแนวภาพถ่าย เมื่อภาพถ่ายถูกนำเข้าสู่อะกิสต์โฟโตสแกน แล้วจำเป็นต้องมีการจัดแนว โดยในขั้นตอนนี้ Agisoft PhotoScan จะค้นหาตำแหน่งกล้องและการวางแนวสำหรับแต่ละภาพถ่ายและสร้างรูปแบบข้อมูลจุดความสูงหนาแน่นแบบเบาบาง
- (4) การสร้าง Point Cloud หรือข้อมูลจุดความสูงหนาแน่น
- (5) การสร้าง Mesh หรือแบบจำลองพื้นผิวเชิงเลข
- (6) การสร้าง Texture หรือเนื้อภาพ
- (7) การสร้างแบบจำลองความสูงเชิงเลข หรือ DEM
- (8) การสร้างแผนที่ภาพถ่าย หรือ Orthophoto
- (9) การสร้างรายงานการประมวลผล
- (10) การส่งออกผลลัพธ์เพื่อนำไปใช้งานอื่นๆ

คุณภาพของภาพ (Image quality) ข้อมูลภาพที่ไม่ดีส่งผลทำให้การจัดตำแหน่งไม่ดี การแก้ไขที่แนะนำคือให้ปิดภาพที่มีค่าคุณภาพต่ำกว่า 0.5 หน่วย หากต้องการยกเลิกภาพถ่ายใดให้ใช้ปุ่มปิดใช้งานจากแถบเครื่องมือรูปภาพ โดย Agisoft Photoscan จะประมาณการคุณภาพของภาพสำหรับแต่ละรูปแบบการนำเข้า และค่าของพารามิเตอร์ถูกคำนวณตามระดับความคมชัดของส่วนที่โฟกัสที่สุดของภาพ (Agisoft, 2017)

#### 2. Pix4d Mapper

**Pix4D Mapper** เป็นโปรแกรมการทำแผนที่ขั้นสูงที่สามารถประมวลผลสร้างโมเดล 3 มิติ จากภาพถ่ายทางอากาศแบบอัตโนมัติ โดยใช้เทมเพลตมาตรฐานสำหรับการประมวลผลอัตโนมัติหรือการตั้งค่าแบบกำหนดเองเพื่อการควบคุมคุณภาพข้อมูล Point Cloud, DSMs และ Orthomosaic และยังสามารถส่งออกข้อมูลแผนที่แบบดิจิทัล โมเดล 3 มิติ และรายละเอียดต่างๆ ที่จำเป็นได้ ซึ่งผลลัพธ์ที่ได้มีดังนี้ Full-color point cloud, Orthomosaic, Digital surface model (DSM), 3D textured mesh, Index map และ Thermal maps (Pix4D, 2018)

### 3. MATLAB

MATLAB เป็นภาษาคอมพิวเตอร์ระดับสูงที่มาพร้อมกับสภาพแวดล้อมการทำงานเชิงโต้ตอบ ซึ่งสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วเร็วกว่าภาษาคอมพิวเตอร์สมัยก่อน แสดงผลกราฟฟิก และเขียนแอฟพลิเคชัน ทำให้สามารถคำนวณผลลัพธ์ พัฒนาอัลกอริทึม สร้างแบบจำลอง และแอฟพลิเคชันได้ง่ายและรวดเร็วมาก

ข้อมูลภาพดิจิทัล เป็นเซตของค่าตัวเลขที่แทนระดับความเข้มของสัญญาณที่เก็บอยู่ในรูปของเมทริกซ์ เนื่องจากโปรแกรม MATLAB รองรับข้อมูลพื้นฐานเป็นอะเรย์หลายมิติ รวมถึงเมทริกซ์ และตัวดำเนินการต่างๆ ทำให้มีความเหมาะสมที่จะใช้โปรแกรมนี้ในการประมวลผลภาพ

#### 2.1.4 แนวคิดเกี่ยวกับเทคนิควิธีการ

##### 1. ภาษา Python

**Python** เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรม ซึ่งภาษา Python พัฒนาขึ้นโดย Guido van Rossum ณ National Research Institute for Mathematics and Computer Science ประเทศเนเธอร์แลนด์ และถูกเผยแพร่ครั้งแรกในปี ค.ศ. 1991 โดย Python เป็นภาษาแบบ Interpreter ที่ถูกออกแบบโดยมีวัตถุประสงค์ที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอื่นๆ อย่าง C++ และ Java ซึ่งภาษา Python ถูกกำหนดให้มีโครงสร้างที่ทำให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมขนาดเล็กไปจนถึงโปรแกรมขนาดใหญ่ (สุชาติ คุ่มมะณี, 2558)

##### 2. Computer Vision

**Computer Vision** คือการประมวลผลภาพโดยคอมพิวเตอร์ กล่าวคือเป็นการจัดสภาพแวดล้อมทั้งทางด้านซอฟต์แวร์และฮาร์ดแวร์ของคอมพิวเตอร์ให้พร้อมในการรับรู้ข้อมูลภาพทั้งภาพนิ่งและภาพเคลื่อนไหว จากนั้นนำมาวิเคราะห์ภาพที่ได้ในด้านต่างๆ ด้วยคอมพิวเตอร์ เปรียบได้กับดวงตาของมนุษย์เมื่อมองภาพ ทำให้สามารถนำภาพนั้นมาวิเคราะห์ด้วยสมองว่าเป็นภาพอะไร คุณลักษณะเด่นของภาพมีอะไรบ้าง (ชุตีสันต์ เกติวิบูลย์เวช, 2554)

##### 3. Edge detection

**การหาขอบภาพ (Edge Detection)** เป็นการใช้หลักการหาความชันของความเข้มสี (Intensity) เนื่องจากที่ขอบรูปจะเป็นบริเวณมีความแตกต่างของสีมาก ซึ่งหากหาความชันของค่า Intensity ก็จะได้ความชันมาก แต่หากรูปนั้นไม่มีขอบ ค่า Intensity บริเวณนั้นจะใกล้เคียงกันหรือเท่ากัน ทำให้ไม่มีความชัน (มหศักดิ์ เกตุฉำ, 2561)

##### 4. Canny

ขั้นตอนการค้นหาขอบภาพโดยวิธีของ **Canny** ประกอบด้วย 4 ขั้นตอน โดยเริ่มต้นจากการปรับภาพให้เรียบ (Smoothing) ด้วยตัวกรองเกาส์เซียนเพื่อกำจัดสัญญาณรบกวน หลังจากนั้นหาอนุพันธ์อันดับหนึ่ง คำนวณค่าขนาดและทิศทางของเกรเดียนต์ นำค่าที่ได้มาคำนวณค่าของ Non-maxima Suppression กับค่าขนาดของเกรเดียนต์เพื่อทำให้ได้ขอบที่บางลง และในขั้นตอนสุดท้ายใช้



การกำหนดจุดอ้างอิงสองระดับ (Double Thresholding) เพื่อระบุค่าของพิกเซลที่เป็นขอบและช่วยเชื่อมต่อขอบ (มหศักดิ์ เกตุฉ่ำ, 2561)

#### 5. การดึงวัตถุพื้นหน้าออกจากพื้นหลังโดยใช้ Threshold

**กระบวนการ Segmentation** เป็นกระบวนการแยกวัตถุ หรือองค์ประกอบต่างๆ ออกจากภาพอินพุท โดยวิธีการ Intensity Thresholding เป็นการแยกแยะวัตถุจากฉากหลังโดยดูจาก Intensity ของพิกเซลเป็นหลัก ข้อดีของของวิธีการนี้คือมีขั้นตอนในการทำงานที่ง่ายไม่ซับซ้อนและทำงานได้รวดเร็ว แต่ข้อเสียคือไม่สามารถใช้กับภาพที่มีสัญญาณรบกวนมาก หรือภาพที่มีความสว่างไม่สม่ำเสมอ เทคนิคการทำขีดแบ่ง (Thresholding Techniques) เป็นการพิจารณาว่าจุดภาพใดควรเป็นจุดขาวหรือจุดดำ ซึ่งทำได้โดยการเปรียบเทียบระหว่างจุดภาพเริ่มต้นกับค่าคงที่ค่าหนึ่ง เรียกว่า “ค่าขีดแบ่ง” ซึ่งเป็นค่าความเข้มแสงค่าหนึ่งที่ใช้แยกแยะประเภทของจุดภาพ เทคนิคนี้ใช้กันมากในกรณีที่มีข้อมูลภาพมีลักษณะแตกต่างกันระหว่างวัตถุและพื้นหลัง สำหรับการตัดพื้นหลังออกจากพื้นหน้าโดยใช้วิธี Threshold ที่มีระดับความเข้มอยู่ระหว่างกลุ่มทั้งสองของ Histogram ซึ่งค่า Threshold ที่ได้จะอยู่ระหว่าง 0-255 เท่านั้น Threshold จะถูกนำไปใช้เพื่อเปรียบเทียบค่าของแต่ละพิกเซล หากค่า  $f(x, y)$  น้อยกว่า Threshold จุดพิกเซลนั้นจะถูกปรับให้เป็นสีดำหรือเป็นส่วนของวัตถุ และหากค่า  $f(x, y)$

#### 6. Segmentation

**กระบวนการ Segmentation** เป็นกระบวนการแยกวัตถุ หรือองค์ประกอบต่างๆ ออกจากภาพอินพุท โดยจะใช้สีของ Marker ซึ่งมีความเด่นชัดจากสีของพื้นหลังในการ Segmentation (NEXTSOFTWAREHOUSE, 2014) มีขั้นตอนดังนี้

- (1) ปรับปรุงคุณภาพของภาพ โดยใช้เทคนิคการเบลอภาพ
- (2) แปลงปริภูมิสีจากปริภูมิ RGB เป็นปริภูมิ HSV
- (3) ตั้งค่า Threshold ของค่า H, S และ V ให้สอดคล้องกับสีของ Marker
- (4) กำหนดให้ค่าของจุดสีที่เป็นองค์ประกอบที่ต้องการพิจารณามีค่าเป็นสีขาว (255) และกำหนดให้ค่าของจุดสีที่ไม่ใช่องค์ประกอบที่ต้องการพิจารณามีค่าเป็นสีดำ (0)

#### 7. Graphical User Interface

**Graphical User Interface GUI** (อ่านว่า จียูไอ หรือ กูยู) คือ การติดต่อกับผู้ใช้โดยใช้ภาพสัญลักษณ์ เป็นการออกแบบส่วนของโปรแกรมคอมพิวเตอร์ให้มีการโต้ตอบกับผู้ใช้ โดยการใช้ Icon รูปภาพ และสัญลักษณ์อื่นๆ เพื่อแทนลักษณะต่างๆ ของโปรแกรมแทนที่ผู้ใช้จะพิมพ์คำสั่งต่างๆ ในการทำงาน ทำให้ผู้ใช้งานสามารถทำงานได้ง่ายและรวดเร็วขึ้น ไม่จำเป็นต้องจดจำคำสั่งต่างๆ ของโปรแกรมมากนัก ถือเป็นวิธีการให้ความสะดวกแก่ผู้ใช้คอมพิวเตอร์ให้ติดต่อสื่อสารกับระบบโดยผ่านทางภาพ เช่น ใช้เมาส์กดเลือก Icon แทนการพิมพ์คำสั่งเหมือนแต่ก่อน โดยเฉพาะอย่างยิ่งในบางโปรแกรมที่มีคำสั่งมากๆ เช่น โปรแกรม AutoCAD ที่ใช้ในการวาดแบบ ซึ่งจะมีคำสั่งต่างๆ ที่ใช้ในการสร้างรูปมากมาย ผู้ใช้สามารถใช้เมาส์ (Mouse) เลือกคำสั่งที่ต้องการจะวาดจาก Icons ที่ปรากฏในโปรแกรมและใช้งานได้เลย โดยไม่ต้องพิมพ์คำสั่งต่างๆ ทางแป้นพิมพ์ ทำให้เกิดความรวดเร็วใน



การทำงาน และไม่ต้องเสียเวลาในการเรียนรู้และจดจำคำสั่งที่ต้องการมากนัก เพียงดูจาก Icons ที่ปรากฏในโปรแกรมก็สามารถใช้งานได้ทันที

ตัวอย่างโปรแกรมที่ช่วยออกแบบโปรแกรมที่ใช้ GUI เช่น Microsoft Visual Basic เป็นต้น ตัวอย่างของซอฟต์แวร์ที่ใช้ GUI ในการใช้งาน เช่น ระบบปฏิบัติการ Windows และซอฟต์แวร์ที่ใช้งานบน Windows ทั้งหมด เช่น Microsoft Office, Adobe Photoshop และอื่นๆ รวมทั้งระบบปฏิบัติการ Mac OS X และ Unix และโปรแกรมที่ใช้งานบน Mac OS X และ Unix ด้วย นอกจากนี้ระบบปฏิบัติการของอุปกรณ์พกพา เช่น iOS, Android, Symbian, BlackBerry และ Windows Phone ก็ใช้ระบบ GUI เพื่อการสื่อสารระหว่างผู้ใหญ่และอุปกรณ์นั้นๆ (นฤชาติ ฦ ตะกั่วทุ่ง, 2557) ซึ่งอาจกล่าวได้ว่าประโยชน์ของ GUI คือ มีความสวยงาม น่าใช้งาน และใช้งานได้ง่าย รวดเร็ว อีกทั้งไม่จำเป็นต้องมีความรู้ด้านการเขียนคำสั่งหรือ Code ภาษาที่คอมพิวเตอร์เข้าใจ

#### 8. Hough Transform

**Hough Transform** เป็นวิธีการที่นิยมใช้ในการวิเคราะห์หาวัตถุภายในภาพที่มีรูปทรงเรขาคณิต (Parametric Shape) ที่สามารถสร้างรูปร่างได้จากสมการได้ เช่น วงกลม เส้นตรง โดยวิธีการประมวลผลเพื่อหารูปทรงวัตถุดังกล่าวภายในภาพด้วย Hough Transform มีหลายวิธีคือ Hough Line Transform และ Hough Circle Transform ส่วนการวิเคราะห์หาวัตถุรูปทรงทั่วไปด้วย Hough Transform ที่ไม่ใช่รูปทรงเรขาคณิต (Arbitrary Shape) สามารถใช้วิธีการ Generalized Hough Transform และ Modified Generalized Hough Transform ในการวิเคราะห์หาตำแหน่งวัตถุต่างๆ (กิตติวัฒน์ นิเมเกิดผล, 2549)

การคำนวณหามุมเส้นตรงภายในภาพด้วยวิธีการ Hough Line ทำได้โดยการสร้าง Accumulator array (A) ของ  $m, c$  จากสมการ (3) หรือสมการ (4) จากนั้นทำการวาดเส้นโดยการ Sampling ค่า  $m$  หรือ  $c$  ของทุกจุดสีดำ (X,Y) ภายในภาพ ซึ่งตำแหน่งที่มีค่าสูงสุดของ A คือ ตำแหน่งมุมของเส้นตรงต่างๆ ที่พบจากรูปที่นำมาวิเคราะห์ สามารถแสดงสมการได้ ดังนี้

$$\text{สมการเส้นตรง} \quad y = mx + c \quad (3)$$

$$\text{หรือ} \quad c = y - mx \quad (4)$$

สมการเส้นตรงดังกล่าวจะเกิดปัญหาเมื่อเส้นเป็นแนวตั้งเนื่องจากค่า  $m, b$  จะเพิ่มสูงมาก ซึ่งสามารถแก้ไขได้โดยใช้การประมวลผลจากสมการ (5) โดย  $p$  คือ ระยะที่สั้นที่สุดจากจุดอ้างอิงถึงจุดบนแนวเส้น  $(x,y)$  และ  $\theta$  เป็นมุมของเส้นนั้นๆ กับแกน  $x$  โดยมีหลักการทำงานเช่นเดียวกับการใช้สมการ (3) ซึ่งสามารถหาค่า  $p$  ได้จากการ Sampling ค่า  $\theta$  ดังนี้

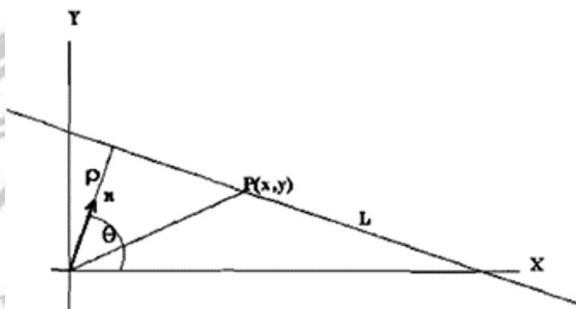
$$\text{Polar } p = x - \cos \theta + y \sin \theta \quad (5)$$

การคำนวณหาตำแหน่งวงกลมในภาพด้วยวิธีการ Hough circle มีหลักการเช่นเดียวกับ Hough Line ซึ่งทำได้โดยการสร้าง Accumulator array (A) ของรัศมีที่ต้องการหา ( $r$ ) แล้วทำ

การวาดวงกลมใน A ด้วยรัศมี  $r$  ของทุกจุดสีดำ (X,Y) ภายในภาพ ซึ่งตำแหน่งที่มีค่า vote สูงของ A คือ ตำแหน่งที่พบวงกลมรัศมีดังกล่าว โดยสามารถแสดงการคำนวณดังสมการ (6) และสมการ (7)

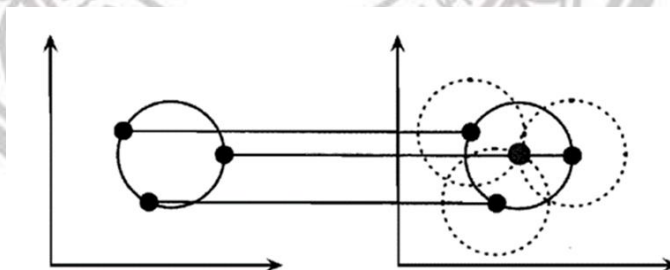
$$\text{จากสมการเส้นตรง} \quad (x - a)^2 + (y - b)^2 = r^2 \quad (6)$$

$$\text{หรือ} \quad x = a + r \cdot \cos \theta, y = b + r \cdot \sin \theta \quad (7)$$



ภาพ 2.9 สมการเส้นตรงในรูปแบบเชิงขั้ว (Polar form) โดย  $L$  คือ แนวเส้นของจุดต่างๆ ภายในรูป,  $k$  คือ เวกเตอร์หน่วย (Unit Vector) ของแนวเส้น  $L$  กับจุดอ้างอิง และ  $P(x,y)$  คือ จุดต่างๆ ที่อยู่ในแนวเส้น  $L$

ที่มา: กิตติวัฒน์ นิมเกิดผล (2549)



ภาพ 2.10 การหาวงกลมด้วยวิธีการ Hough Circle Transform: (ซ้าย) รูปภาพที่ต้องการหาวงกลมรัศมี  $R$  ซึ่งมีจุดอยู่ 3 จุด และ (ขวา) วาดวงกลมรัศมี  $R$  ใน Parametric space (accumulator array) โดยจุดที่ตัดกันมากที่สุดคือจุดที่มีค่า vote สูงสุด ซึ่งเป็นจุดศูนย์กลางของคำตอบ

ที่มา: กิตติวัฒน์ นิมเกิดผล (2549)

All rights reserved

### 2.1.5 หลักการทางสถิติ

การทดสอบความถูกต้องในการประมาณค่าโดยใช้วิธีทางสถิติด้วยการหาค่าความคลาดเคลื่อนเฉลี่ยสัมบูรณ์ (Mean Absolute Error: MAE) ดังสมการ (8) และค่ารากที่สองของความคลาดเคลื่อนยกกำลังสองเฉลี่ย (Root Mean Square Error: RMSE) ดังสมการ (9) (วิภารัตน์ อัมพะวัน, 2561)

$$MAE = \frac{1}{n} \sum_{i=1}^n |Estimated\ value - Observed\ value| \quad (8)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [Estimated\ value - Observed\ value]^2} \quad (9)$$

โดยที่ Estimated value คือ จำนวนไม้ผลจากการตรวจนับแบบอัตโนมัติ  
Observed value คือ จำนวนไม้ผลจากการตรวจนับจริง  
MAE คือ ความคลาดเคลื่อนเฉลี่ยสัมบูรณ์  
RMSE คือ ค่ารากที่สองของความคลาดเคลื่อนยกกำลังสองเฉลี่ย  
n คือ จำนวนข้อมูลทั้งหมด

### 2.1.6 ลักษณะทางกายภาพของต้นมะนาว

คำว่า “มะนาว” ในภาษาอังกฤษคนไทยมักจะใช้คำว่า “Lemon” ซึ่งจริงๆ แล้วคำว่า Lemon จะหมายถึงผลส้มที่มีหัวท้ายมน ต่างจากมะนาวที่มีผลกลมๆ อย่างที่เรารู้จัก ดังนั้นควรเรียกมะนาวว่า “Lime” ตามชื่อที่ถูกต้อง ส่วนลักษณะพฤกษศาสตร์ของมะนาวมักจะขึ้นกับพันธุ์ของมะนาว โดยส่วนใหญ่มะนาวจะมีลักษณะเป็นไม้พุ่มหรือไม้ยืนต้นขนาดเล็ก สูงตั้งแต่ 0.5-5 เมตร แล้วแต่พันธุ์ (พันธุ์ที่เตี้ยที่สุดที่มักพบในประเทศไทยคือพันธุ์ “มะนาวเตี้ย” มีความสูงประมาณ 0.5-1 เมตร เท่านั้น) ใบกว้าง 3-9 เซนติเมตร ยาวประมาณ 4-8 เซนติเมตร (พันธุ์ที่ใบใหญ่ที่สุดคือพันธุ์ “มะนาวพม่า” ที่กว้างได้ถึง 9 เซนติเมตร)

- ชื่อวิทยาศาสตร์: Citrus aurantifolia Swing
- ชื่อวงศ์: Rutaceae
- ชื่อสามัญ: Lime, Common Lime
- ลักษณะทางพฤกษศาสตร์: ต้น: เป็นไม้พุ่มสูง 2-4 เมตร เปลือกลำต้นมีสีเทาปนน้ำตาล กิ่งอ่อนมีสีเขียวอ่อน เมื่อแก่สีเข้ม บนลำต้นและกิ่งก้านจะมีหนามแข็งแหลม ส่วนใหญ่เกิดที่ซอกใบ
- ใบ: เป็นใบประกอบชนิดมีใบย่อยใบเดี่ยว เรียงสลับ มีสีเขียวอ่อนรูปร่างยาวรีหรือรูปไข่ ปลายใบมีลักษณะแหลมขอบใบ
- ดอก: เป็นดอกเดี่ยวหรือช่อดอก เกิดบริเวณซอกใบและปลายกิ่งกลีบดอกสีขาว กลีบเลี้ยงสีเขียวอ่อนเกสรตัวผู้อยู่เป็นกลุ่ม เกสรตัวเมียรูปคล้ายทรงกระบอก
- ผล: ผลสดรูปกลมและรูปยาวรี หรือรูปไข่ มีขนาดกว้างยาวประมาณ 3-12 เซนติเมตร ผิวเปลือกมีลักษณะขรุขระ และมีต่อมน้ำมันที่ผิว
- เมล็ด: ขนาดเล็กคล้ายรูปไข่ ด้านปลายหัวจะแหลม ภายในเมล็ดมีเนื้อเยื่อสีขาว

## 2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

Dilek et al. (2018) ได้ศึกษาการนับจำนวนต้นมะนาวแบบอัตโนมัติโดยใช้ภาพจากอากาศยานไร้คนขับ (UAV) ด้วยกระบวนการโฟโตแกรมเมตรีเพื่อให้ได้แบบจำลองพื้นผิวดิจิทัล (DSM) และภาพ orthomosaic (RGB) จากนั้นนำภาพที่ได้มาหาขอบเขตทรงพุ่มของต้นมะนาวด้วยหลักการ Circular Hough Transform และใช้โปรแกรม OpenCV-Python ในการหาขอบเขตทรงพุ่มของต้นมะนาวเพื่อจะนับจำนวนต้นมะนาวจากขอบเขตทรงพุ่มของต้นมะนาว ในการศึกษานี้ได้นับจำนวนต้นส้มแบบอัตโนมัติจำนวน 3 ส่วน และแบ่งข้อมูลเป็น 3 ชุด ได้แก่ ชุดที่ 1 (DSM) ชุดที่ 2 (DSM)+(RGB) และชุดที่ 3 (RGB) พบว่าผลลัพธ์ที่ได้มีความถูกต้องของการวิเคราะห์เกิน 80% และผลลัพธ์ที่แม่นยำที่สุดคือชุดที่ 1 แม้ว่าชุดที่ 2 และชุดที่ 3 จะให้ค่าความถูกต้องต่ำเมื่อเปรียบเทียบกับชุดที่ 1 แต่ความสามารถในการแยกสัญญาณยังคงสูงและสามารถใช้งานได้โดยเฉพาะอย่างยิ่งการนับต้นไม้และการตรวจจับตำแหน่งต้นไม้ ดังภาพ 2.11

Gistda Academy (2015) ได้ประยุกต์ใช้ข้อมูลระยะไกลในการบริหารจัดการการเกษตร โดยมีการพัฒนาวิธีการแจงนับต้นไม้อินภาพถ่ายขึ้น ซึ่งในขั้นต้นของการพัฒนาได้ใช้ภาพของแปลงปลูกต้นปาล์มน้ำมันเนื่องจากต้นปาล์มน้ำมันถือเป็นพืชเศรษฐกิจที่สำคัญของประเทศไทย โดยข้อมูลหลักที่ใช้เป็นภาพถ่ายดาวเทียมความละเอียดสูงแบบหลายช่วงคลื่น (High Resolution Multi-spectral Satellite Image) นอกจากนี้ข้อมูลสำคัญที่ใช้ในกระบวนการแจงนับต้นไม้คือ ขอบเขตของแปลงปลูก ซึ่งสามารถใช้ Shape file ในฐานข้อมูล หรือใช้มีอระบุงขอบเขตของแปลงปลูก การทำงานของระบบที่พัฒนาขึ้นอยู่บนหลักการของสถิติเชิงพื้นที่ (Spatial statistics) และการประมวลผลภาพถ่ายดิจิทัล (Digital Image Processing) เพื่อทดสอบประสิทธิภาพของวิธีการที่พัฒนาขึ้นได้มีการใช้ภาพของแปลงปลูกปาล์มน้ำมันบริเวณจังหวัดพังงา และจังหวัดภูเก็ต ซึ่งถ่ายภาพโดยใช้ดาวเทียม Quickbird และ WorldView-2 จากการทดลองพบว่าผลการแจงนับมีความถูกต้องมากกว่าร้อยละ 90 เมื่อเปรียบเทียบกับการใช้สายตามนุษย์ในการแจงนับ โดยจุดเด่นของวิธีการที่พัฒนาขึ้นคือสามารถนำไปใช้กับภาพถ่ายจากตัวรับรู้ชนิดอื่น เช่น ภาพถ่ายทางอากาศ ดังภาพ 2.12

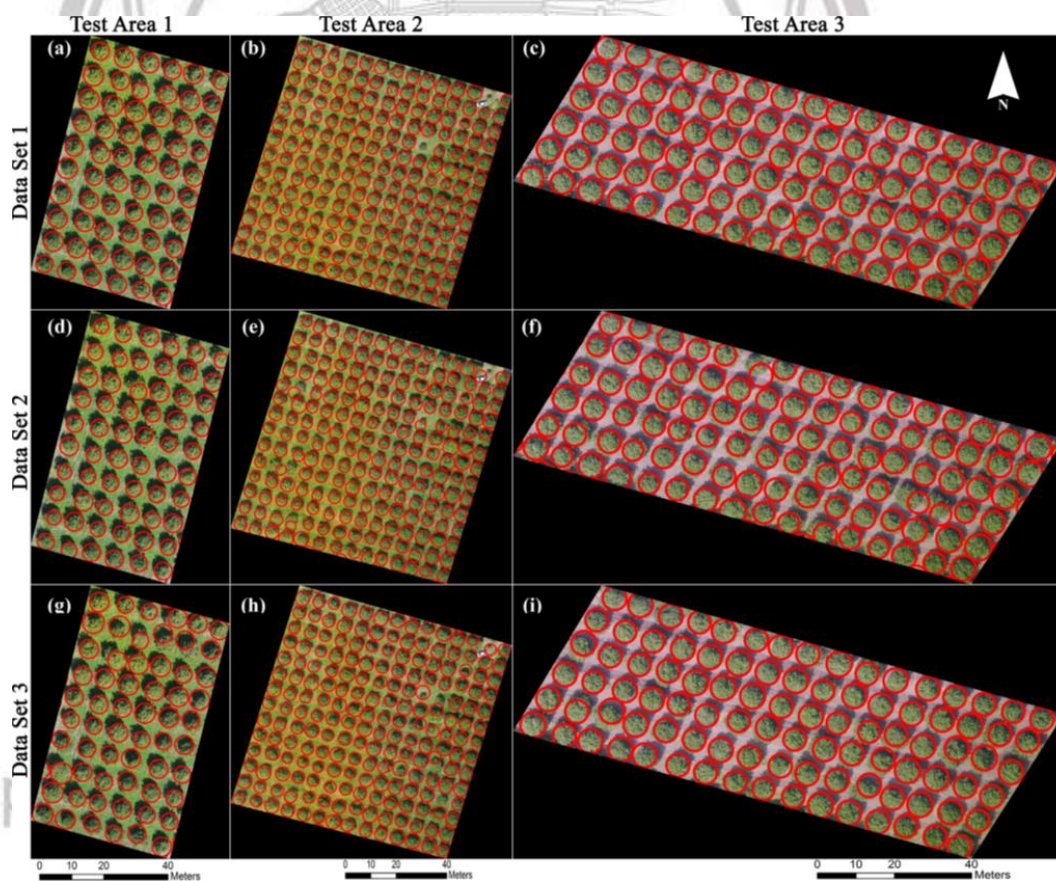
Maciel et al. (2018) ได้ทำการตรวจจับต้นส้มแบบอัตโนมัติจากภาพ UAV โดยใช้เครือข่ายประสาทเทียม (Convolutional Neural Networks: CNN) ซึ่งเป็นการจัดการกับปัญหาในการตรวจจับต้นส้มในส่วนที่มีความหนาแน่นสูง เนื่องจากวิธีการตรวจนับที่เหมาะสมสำหรับพื้นที่ที่มีการกระจายของต้นไม้มาก่อนหน้านี้ไม่เหมาะสมสำหรับข้อมูลของงานวิจัยนี้ ดังนั้นในการศึกษาครั้งนี้จึงเสนอวิธีการใหม่ในการตรวจนับคือ Convolutional Neural Networks (CNN) ซึ่งแบ่งออกเป็นสามขั้นตอน ได้แก่ (1) การใช้ CNN สำหรับอนุมาณจุดกึ่งกลางของแถวต้นไม้ (2) การแบ่งส่วนพื้นที่ความน่าจะเป็นของเส้นตรง และ (3) การใช้ CNN เพื่อจำแนกขอบเขตของต้นไม้ พบว่าการทดสอบมีค่าความแม่นยำโดยรวมที่ 94%

Emad et al. (2018) ได้ทำการตรวจนับจำนวนต้นปาล์มจากภาพถ่ายดาวเทียมที่มีความละเอียดเชิงพื้นที่สูง ซึ่งสามารถมองเห็นลักษณะของต้นปาล์มได้อย่างชัดเจน พบว่าต้นปาล์มมีลักษณะการเพาะปลูกที่เป็นแนว ในขณะที่ต้นไม้อื่นๆ มีรูปร่างไม่สม่ำเสมอส่วนใหญ่เป็นเพราะมีกิ่งก้านที่บิดเบี้ยว ต้นปาล์มประกอบด้วยลำต้นยาวหัวใหญ่และมีทรงพุ่มที่ขนาดใหญ่เกือบจะเป็นวงกลมและรูปทรงกลมของมุมมองด้านบนเป็นคุณสมบัติพิเศษที่สามารถใช้ในการออกแบบโปรแกรมที่

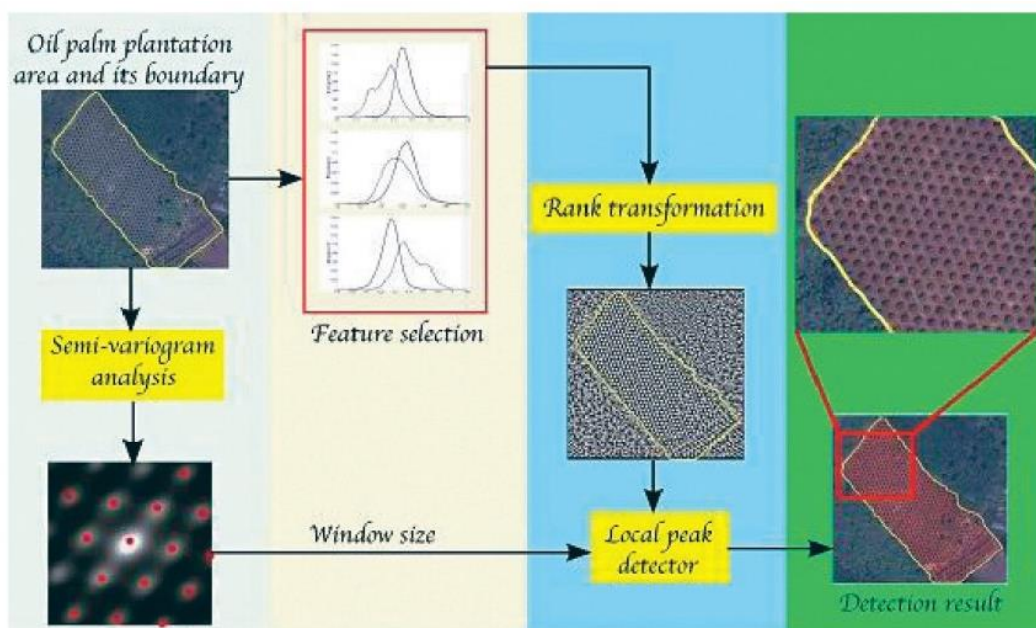


แยกรูปร่างของต้นปาล์ม จากนั้นจะนับจำนวนต้นปาล์มที่ปรากฏบนภาพถ่ายดาวเทียม โดยวิธีนี้มีประโยชน์ในการนับจำนวนต้นปาล์มเพื่อการคำนวณการเกษตรหรือสิ่งแวดล้อม นอกจากนี้ยังสามารถนำไปใช้กับภาพถ่ายดาวเทียมความละเอียดสูงเช่น Quick Bird ได้เนื่องจากความละเอียดของภาพอยู่ที่ 0.6 เมตร และยังสามารถนำไปประยุกต์ใช้กับภาพถ่ายทางอากาศได้ ซึ่งภาพถ่ายดาวเทียมสามารถเข้าถึงได้ฟรีจาก Google Earth explorer

คริส เอื้อไพบูลย์ชัย (2016) ได้ศึกษาการคัดขนาดผลลำไยโดยใช้การแปลงฮัฟ โดยได้นำเสนอการคัดขนาดผลลำไยโดยใช้การประมวลผลภาพดิจิทัล ซึ่งขั้นตอนแรกจะเป็นการกรองสัญญาณความถี่ต่ำผ่านและการตรวจหาขอบภาพแบบแคนนี่เพื่อลดข้อมูลที่ไม่ต้องการ จากนั้นทำการแปลงฮัฟ สำหรับตรวจหาวงกลมโดยไม่กำหนดรัศมีเพื่อตรวจหาภาพผลลำไย และขั้นตอนสุดท้ายจะเป็นการดึงลักษณะสำคัญด้านขนาดของผลลำไยและทำการคัดแยกประเภท โดยผลการทดลองแสดงให้เห็นว่าวิธีการที่นำเสนอสามารถคัดขนาดของผลลำไยได้อย่างมีประสิทธิภาพ



ภาพ 2.11 การนับจำนวนต้นมะนาวอย่างอัตโนมัติโดยใช้ภาพจากอากาศยานไร้คนขับ (UAV)  
ที่มา: Dilek (2018)



ภาพ 2.12 ระบบการแจกแจงต้นไม้ในแปลงปลูกการเกษตร  
ที่มา: Gistda Academy (2015)

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## บทที่ 3

### วิธีดำเนินการวิจัย

การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ โดยบูรณาการระหว่างกระบวนการโปรแกรมเมตริกและเทคโนโลยีการประมวลผลภาพเชิงเลข รวมทั้งได้พัฒนา Graphic User Interface (GUI) เพื่อให้ง่ายต่อผู้ใช้งาน มีวิธีดำเนินการวิจัยดังนี้

- 3.1 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย
- 3.2 การเก็บรวบรวมข้อมูล
- 3.3 การประมวลผลข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ
- 3.4 การวิเคราะห์ข้อมูล
- 3.5 การเตรียมความพร้อม

#### 3.1 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย



ภาพ 3.1 เครื่องมือที่ใช้ในการวิจัย

Copyright by Naresuan University

All rights reserved



ตาราง 1.1 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย

ลำดับ	ข้อมูล/อุปกรณ์	ประเภท	แหล่งที่มา/คุณสมบัติ
1	ภาพถ่ายทางอากาศ	Raster	- DJI Phantom 4 Advanced
2	DJI Phantom 4 Advanced	Multicopter	- น้ำหนัก (รวมแบตเตอรี่และใบพัด) 1,380 กรัม - ความละเอียดภาพนิ่ง 20 Mega-Pixel - แบตเตอรี่ Lipo ความจุ 5,350 mAH - ระบุตำแหน่งดาวเทียม GPS/GLONASS
3	ขอบเขตข้อมูลจังหวัด อำเภอลำปาง	Vector	- กรมการปกครอง
4	คอมพิวเตอร์โน้ตบุ๊ก Acer Nitro 5 AN515-51	Hardware	- Intel Core i7-7700HQ - NVIDIA GeForce GTX 1050Ti (4GB GDDR5) - 8 GB DDR4 - 1 TB 5400 RPM + 128 GB SSD
5	Agisoft Photoscan	Software	- โปรแกรมสร้างแผนที่จากข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ
6	Pix4Dmapper	Software	- โปรแกรมสร้างแผนที่จากข้อมูลภาพถ่ายจากอากาศยานไร้คนขับรุ่นทดลองใช้
7	ArcGIS	Software	- โปรแกรมสารสนเทศทางภูมิศาสตร์
8	MATLAB	Software	- โปรแกรมคำนวณทางคณิตศาสตร์

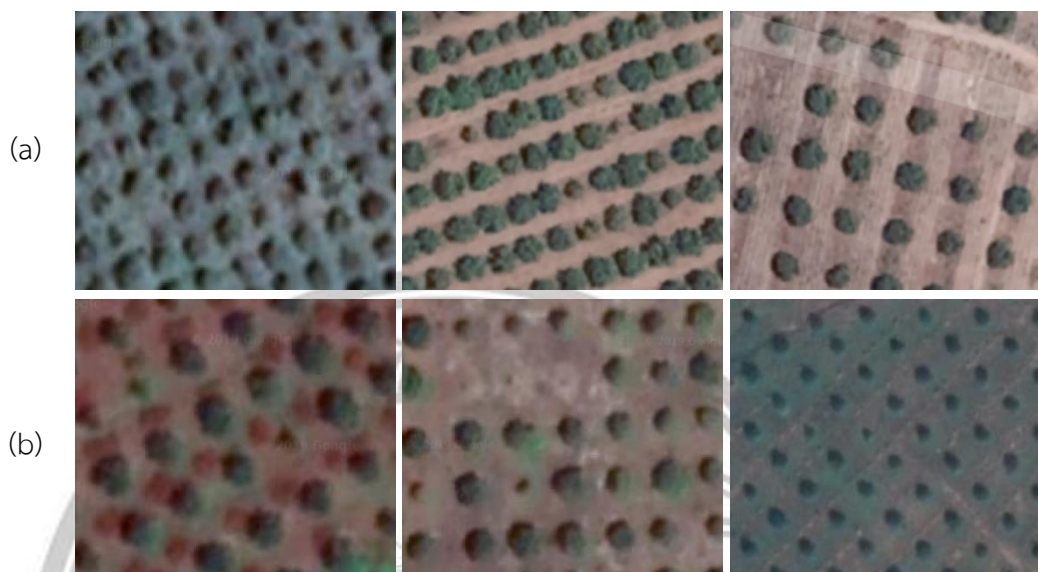
### 3.2 การเก็บรวบรวมข้อมูล

ผู้วิจัยได้ทำการเก็บรวบรวมข้อมูลที่ใช้ในการศึกษาเพื่อเปรียบเทียบผลของการพัฒนาอัลกอริทึม ประกอบด้วยข้อมูลจาก 2 แหล่ง ได้แก่ ข้อมูลภาพถ่ายทางอากาศจาก Google map และข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ โดยใช้วิธีการแปลด้วยสายตาเพื่อจำแนกจำนวนต้นไม้ที่จะนำมาตรวจสอบความถูกต้องของอัลกอริทึมที่ผู้วิจัยได้พัฒนาขึ้น และรวบรวมข้อมูลจากภาคสนาม

#### 3.2.1 ข้อมูลภาพถ่ายทางอากาศจาก Google map

ผู้วิจัยได้ทำการเลือกพื้นที่ตัวอย่างจาก Google map โดยเลือกพืชสวนที่มีลักษณะเป็นทรงพุ่มคล้ายกับทรงพุ่มของต้นมะนาว เพื่อนำภาพไปทดลองกับอัลกอริทึมเบื้องต้นที่ได้พัฒนาใน MATLAB ก่อนที่จะทดลองกับข้อมูลจริง ซึ่งผู้วิจัยได้เลือกพืชมา 2 ชนิด คือ ส้ม ดังภาพ 3.2(a) และทุเรียนดังภาพ 3.2(b) โดยแบ่งเป็นชนิดละ 30 สวน เพื่อทดสอบความสามารถในการนับจำนวนต้นไม้ของอัลกอริทึมที่ได้พัฒนาขึ้นมา





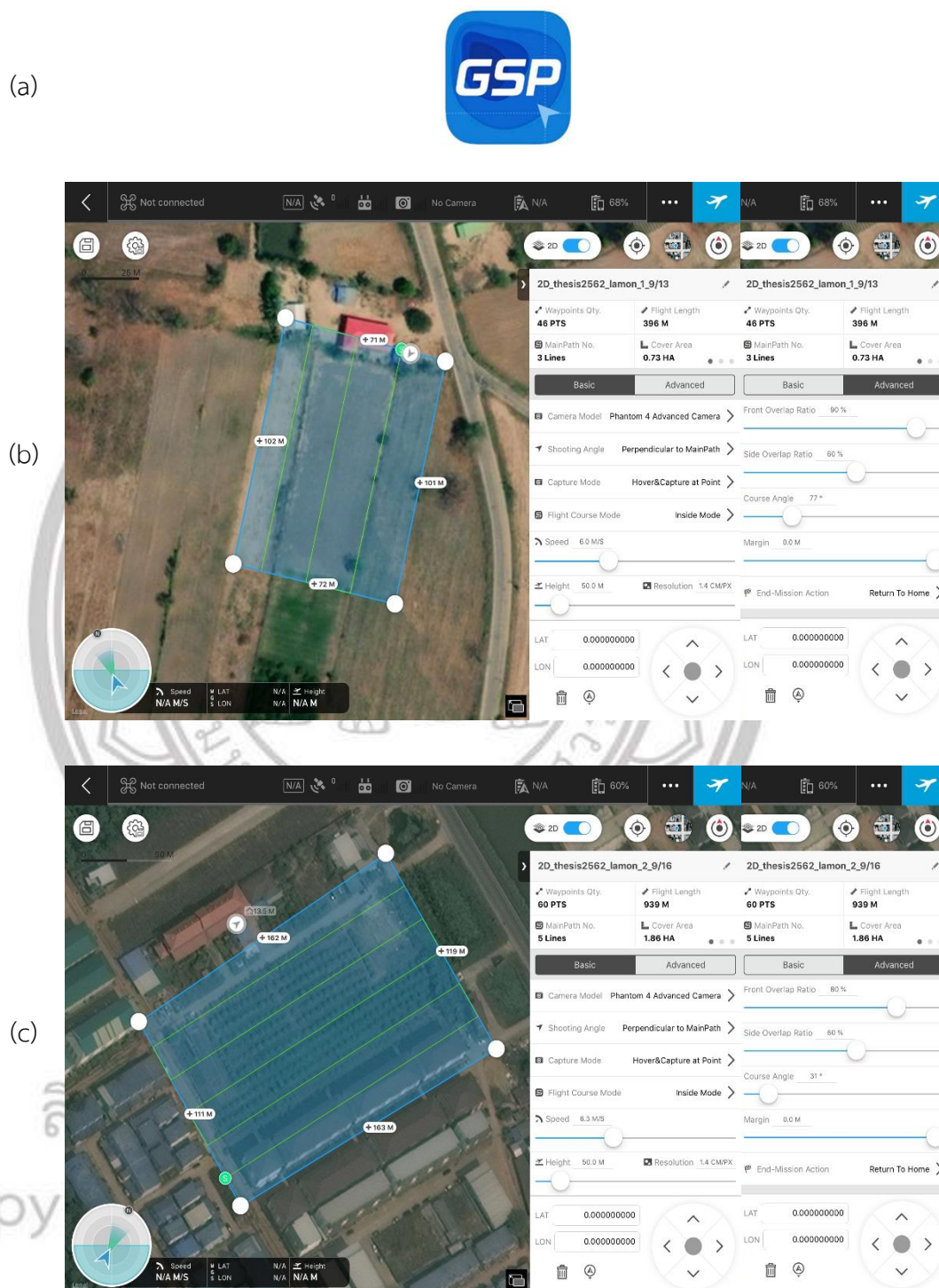
ภาพ 3.2 ข้อมูลภาพถ่ายทางอากาศจาก Google map: (a) ส้ม และ (b) ทูเรียน

### 3.2.2 ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ

ผู้วิจัยได้ทำการวางแผนแนวจินเพื่อถ่ายภาพแปลงปลูกต้นมะนาวที่ใช้ในการศึกษาจำนวน 2 แห่ง คือ บริเวณตำบลสามพวง อำเภอกีรีมาศ จังหวัดสุโขทัย และตำบลท่าโพธิ์ อำเภอมืองพิษณุโลก จังหวัดพิษณุโลก โดยใช้แอปพลิเคชัน DJI GS PRO ดังภาพ 3.3(a) เพื่อทำการวางแผนแนวจินบนพื้นที่สวนมะนาวของจังหวัดสุโขทัย ดังภาพ 3.3(b) โดยกำหนดระดับความสูงบินที่ 50 เมตร กำหนดให้มีส่วนซ้อนทับในแนวจินเดียวกันและระหว่างแนวจินคิดเป็นร้อยละ 90 และ 60 ตามลำดับ จำนวนทั้งสิ้น 3 แนวจิน ครอบคลุมพื้นที่ประมาณ 4.5 ไร่ ความยาวเส้นทางการบินรวมทั้งหมด 396 เมตร ซึ่งได้ภาพทั้งหมดจำนวน 46 ภาพ โดยในซอฟต์แวร์ DJI GS PRO พบว่าพื้นที่บินถ่ายไม่มีพื้นที่สวนหรือพื้นที่พืชใดๆ เนื่องจากข้อมูลภาพที่นำมาเป็นภาพเก่า และผู้วิจัยได้ทำการบินถ่ายภาพบนพื้นที่สวนมะนาวของจังหวัดพิษณุโลก ดังภาพ 3.3(c) โดยบินในระดับความสูงประมาณ 50 เมตร กำหนดให้มีส่วนซ้อนทับในแนวจินและระหว่างแนวจินที่ร้อยละ 80 และ 60 ตามลำดับ จำนวน 5 แนวจิน ครอบคลุมพื้นที่ประมาณ 11.6 ไร่ ความยาวเส้นทางการบิน 939 เมตร ซึ่งได้ภาพทั้งหมดจำนวน 60 ภาพ

Copyright by Naresuan University

All rights reserved



ภาพ 3.3 ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ: (a) แอปพลิเคชัน DJI GS PRO, (b) การวางแผนแนวจังหวัดสุโขทัย และ (c) การวางแผนแนวจังหวัดพิษณุโลก



### 3.2.3 ข้อมูลภาคสนาม

ในการลงภาคสนามผู้วิจัยได้วางขอบเขตสวนมะนาวทั้ง 2 สวน โดยทำการทดลองและตรวจนับจำนวน ซึ่งสวนมะนาวของจังหวัดสุโขทัย มีพื้นที่ประมาณ 1 ไร่ พบว่ามีต้นมะนาวจำนวน 94 ต้น ดังภาพ 3.4(a) และสวนมะนาวของจังหวัดพิษณุโลก มีพื้นที่ประมาณ 1 ไร่ พบว่ามีต้นมะนาวจำนวน 102 ต้น ดังภาพ 3.4(b)



ภาพ 3.4 ขอบเขตสวนมะนาวและจำนวนต้นมะนาว: (a) สวนมะนาวของจังหวัดสุโขทัย และ (b) สวนมะนาวของจังหวัดพิษณุโลก

### 3.3 การประมวลผลข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

ผู้วิจัยได้ใช้โปรแกรม Pix4Dmapper ในการประมวลผลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ โดยมีขั้นตอนการประมวลผล ดังนี้

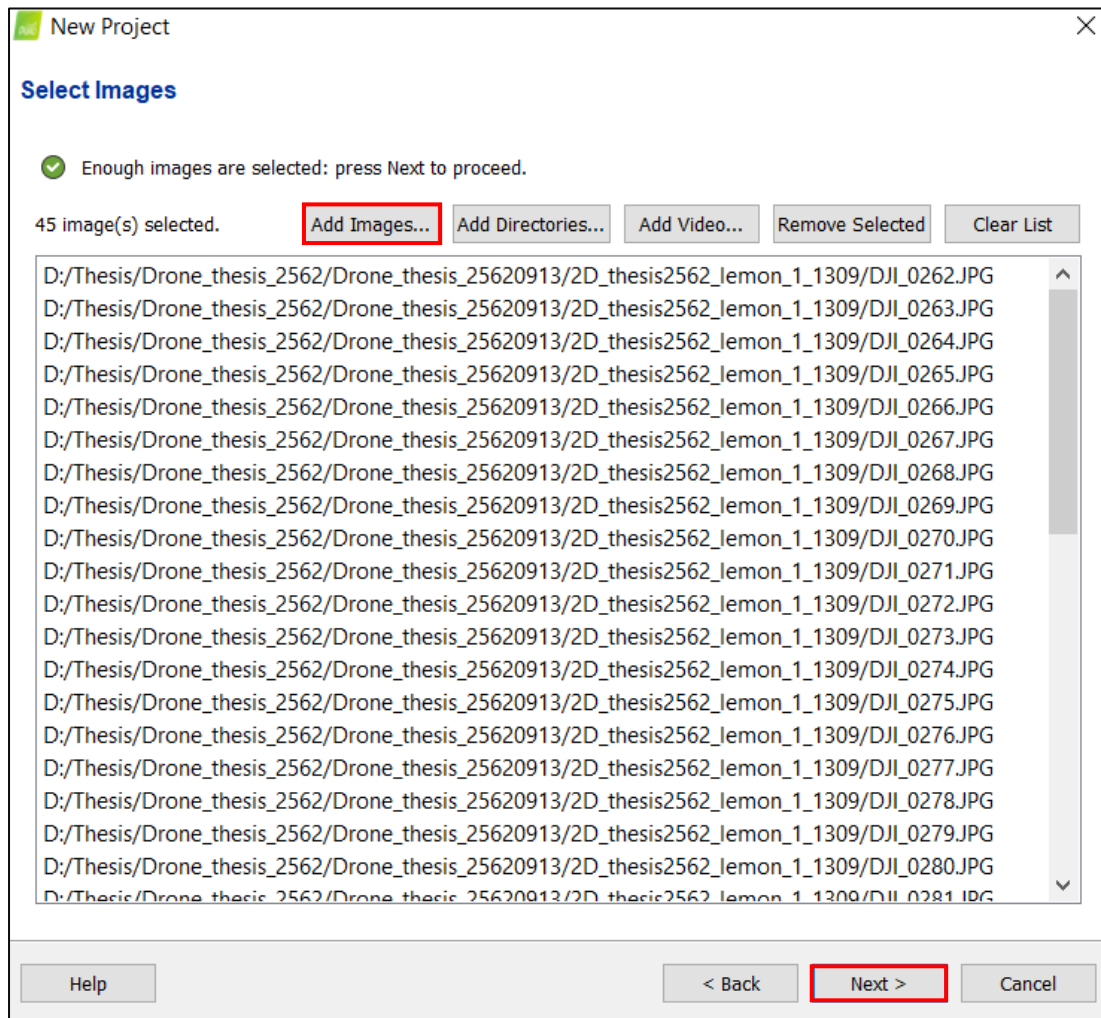
#### 3.3.1 การประมวลผลภาพถ่ายทางอากาศด้วยโปรแกรม Pix4Dmapper

เริ่มต้นทำงานโดยการเปิดโปรแกรม Pix4Dmapper เมื่อโปรแกรมเริ่มทำงานให้ไปที่เมนู Project เลือก New Project จะแสดงหน้าต่างสำหรับกำหนดค่า Project ดังภาพ 3.5 โดยทำการกรอกชื่อ Project ในช่อง Name จากนั้นกด Browse ด้านขวามือเพื่อเลือกที่สำหรับเก็บข้อมูลที่กำลังจะสร้างขึ้น แล้วกด Next

ภาพ 3.5 หน้าต่างสำหรับกำหนดค่า Project ในโปรแกรม Pix4Dmapper

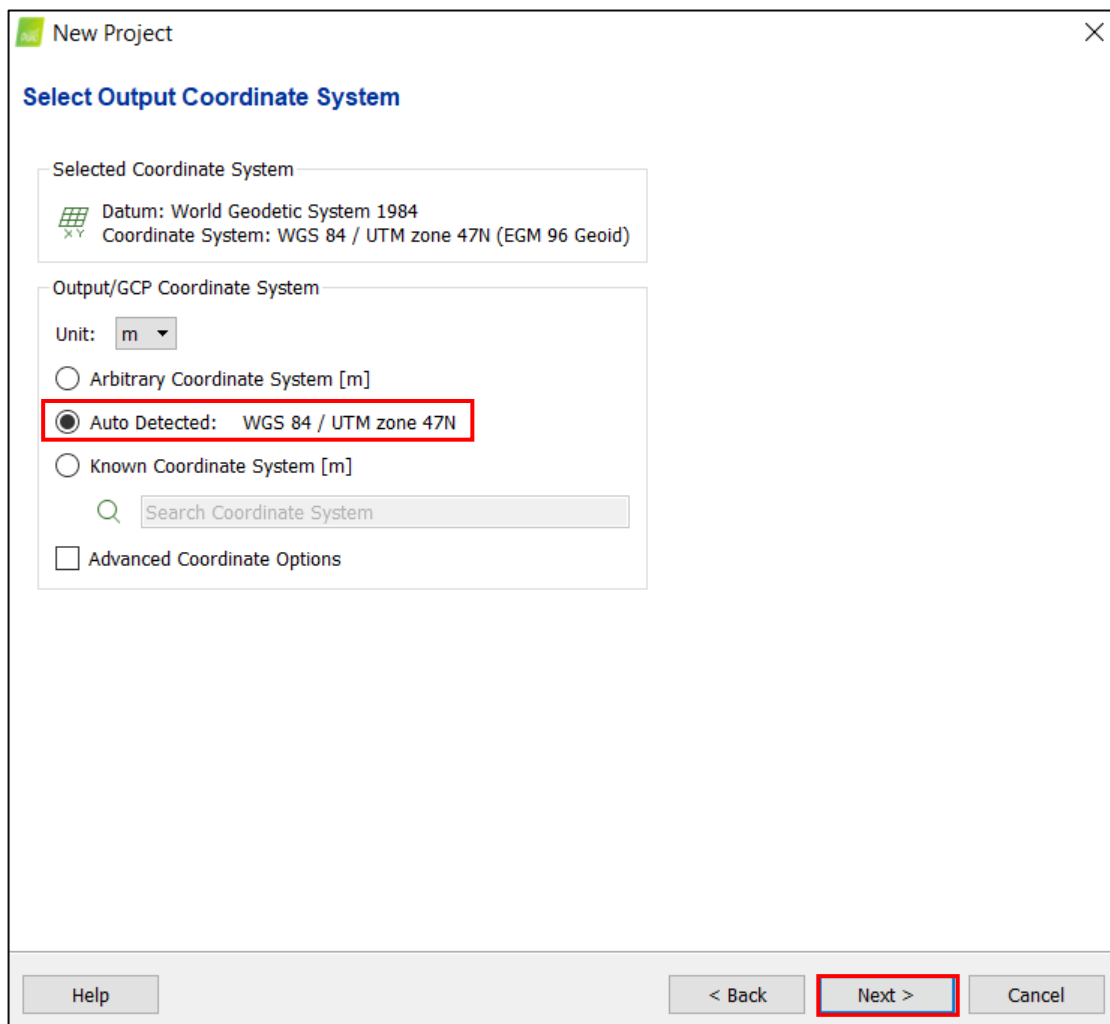


การนำเข้รูปที่ต้องการโดยกดปุ่ม Add Images แล้วทำการเลือกรูปที่ต้องการประมวลผล เมื่อเลือกภาพตามที่ต้องการทั้งหมดแล้วกด Open เพื่อนำเข้รูปภาพ จากนั้นตรวจสอบข้อมูลที่น่าเข้ว่าถูกต้องหรือไม่ แล้วกด Next ดังภาพ 3.6



ภาพ 3.6 การนำเข้ภาพถ่ายทางอากาศในโปรแกรม Pix4Dmapper

การตรวจสอบหรือเลือกระบบพิกัด WGS84/UTM Zone 47N โดยเมื่อตรวจสอบหรือเลือกระบบพิกัดที่ต้องการเสร็จสิ้นแล้วให้ทำการกด Next ซึ่งงานวิจัยนี้กำหนดค่าพิกัดเป็น Auto Detected: WGS84/UTM Zone 47N ดังภาพ 3.7



The screenshot shows a dialog box titled "New Project" with a close button (X) in the top right corner. The main heading is "Select Output Coordinate System".

Under "Selected Coordinate System", there is a grid icon and the text: "Datum: World Geodetic System 1984" and "Coordinate System: WGS 84 / UTM zone 47N (EGM 96 Geoid)".

Under "Output/GCP Coordinate System", there is a "Unit:" dropdown menu set to "m". Below it are three radio button options:

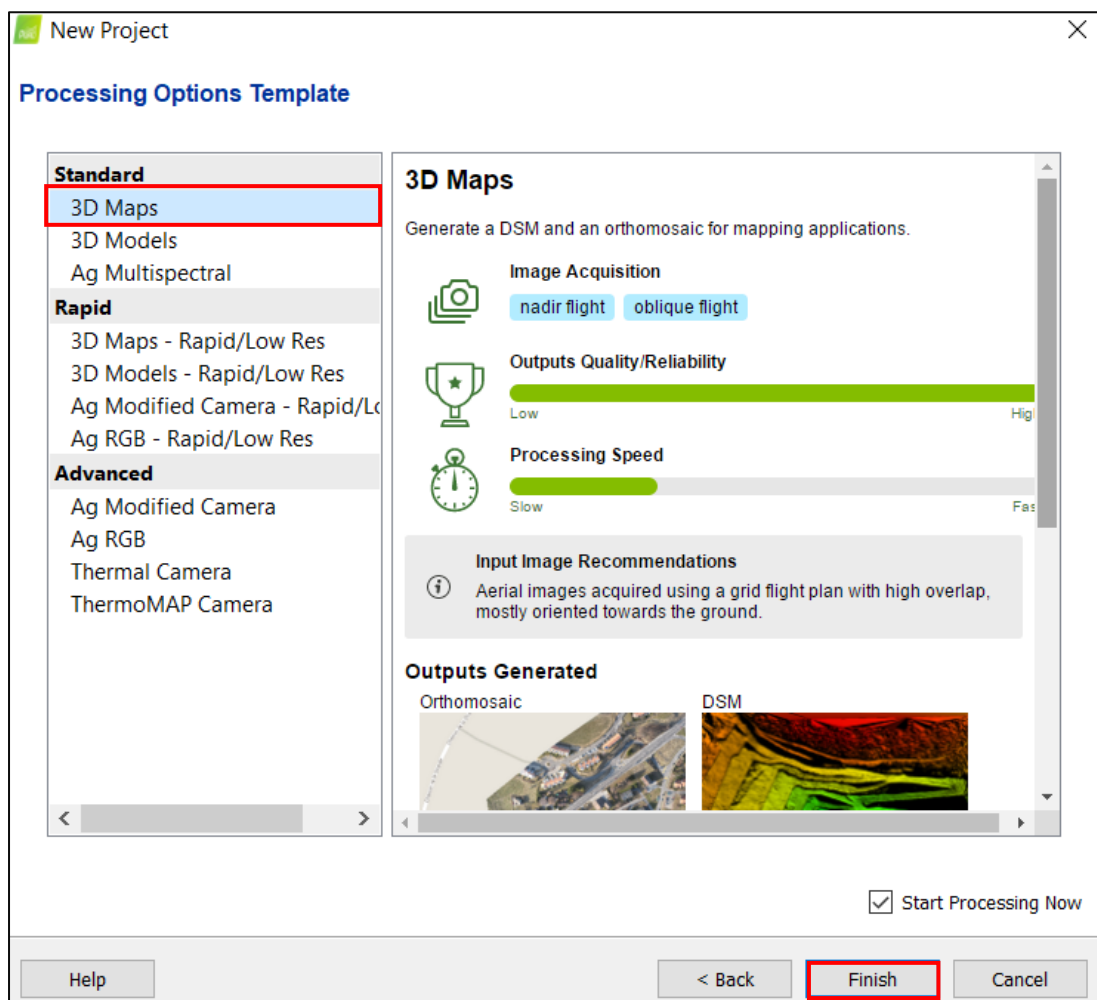
- Arbitrary Coordinate System [m]
- Auto Detected: WGS 84 / UTM zone 47N (This option is highlighted with a red box)
- Known Coordinate System [m]

Below the radio buttons is a search field with a magnifying glass icon and the text "Search Coordinate System". At the bottom left of this section is a checkbox labeled "Advanced Coordinate Options".

At the bottom of the dialog box, there are three buttons: "Help", "< Back", and "Next >" (The "Next >" button is highlighted with a red box), and "Cancel".

ภาพ 3.7 การกำหนดค่าพิกัดเป็น Auto Detected: WGS84/UTM Zone 47N

การเลือก Product Output ที่ต้องการสำหรับข้อมูล โดยเลือกที่ 3D Maps ดังภาพ 3.8 แล้วทำการกด Finish อันเป็นการสิ้นสุดการกำหนดค่า Project ใหม่ในโปรแกรม



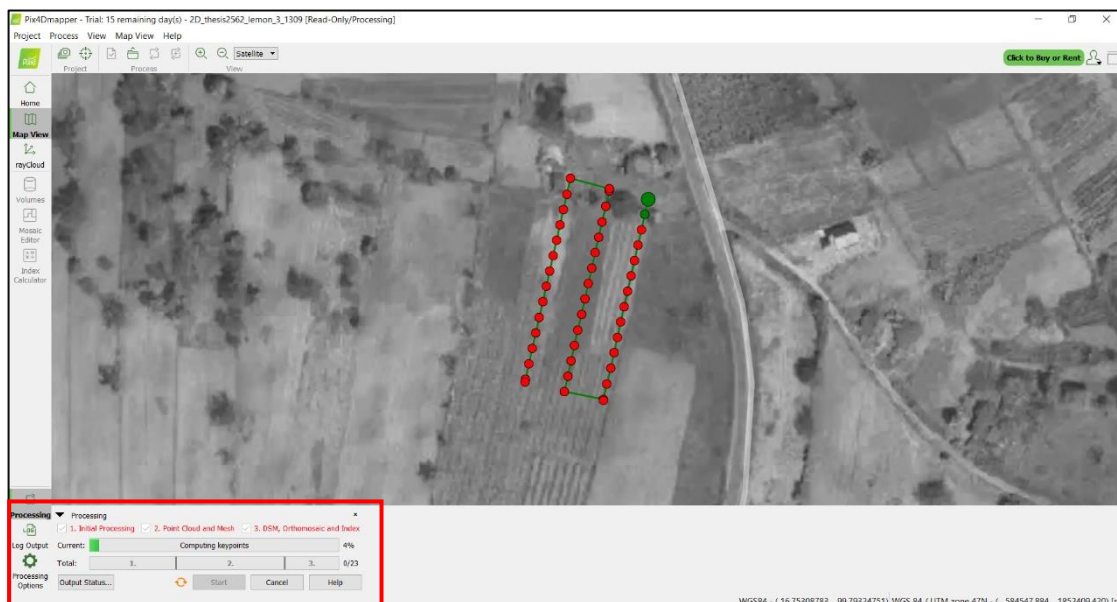
ภาพ 3.8 การกำหนดการประมวลผลเป็นแบบ 3D Maps

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

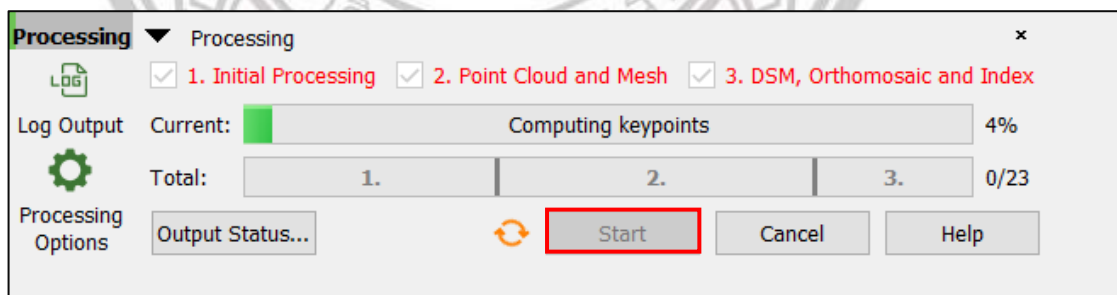
Copyright by Naresuan University

All rights reserved

การเริ่มต้นประมวลผลข้อมูล ซึ่งมีการประมวลผลทั้งหมด 3 ขั้นตอน คือ (1) Initial Processing (2) Point Cloud and Mesh และ (3) DSM, Orthomosaic and Index ให้กด Start ตรงด้านซ้ายล่างของหน้าจอ ดังภาพ 3.9 และภาพ 3.10



ภาพ 3.9 การประมวลผลข้อมูลในโปรแกรม Pix4Dmapper



ภาพ 3.10 ขั้นตอนการประมวลผลข้อมูลในโปรแกรม Pix4Dmapper

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

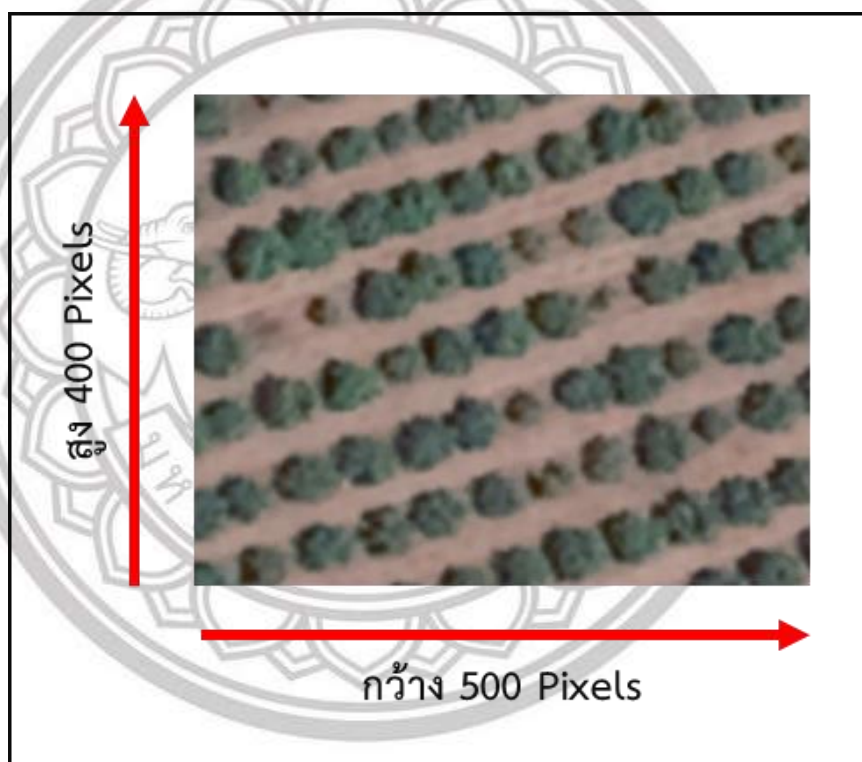
All rights reserved



### 3.4 การวิเคราะห์ข้อมูล

#### 3.4.1 การศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูล Google map

1) การซูมพื้นที่ตัวอย่างจากภาพถ่ายดาวเทียม Google map  
 ผู้วิจัยซูมพื้นที่เพื่อบันทึกภาพหน้าจอพื้นที่ตัวอย่างจากภาพถ่ายดาวเทียม Google map ทั้งสี่มุมและทุเรียนชนิดละ 30 สวน โดยใช้โปรแกรม oCam ในการบันทึกภาพหน้าจอ และกำหนดขนาดรูปภาพเท่ากับ 500x400 คือ กว้าง 500 Pixels สูง 400 Pixels ดังภาพ 3.11



ภาพ 3.11 ซูมพื้นที่ตัวอย่างจากภาพถ่ายดาวเทียม Google map

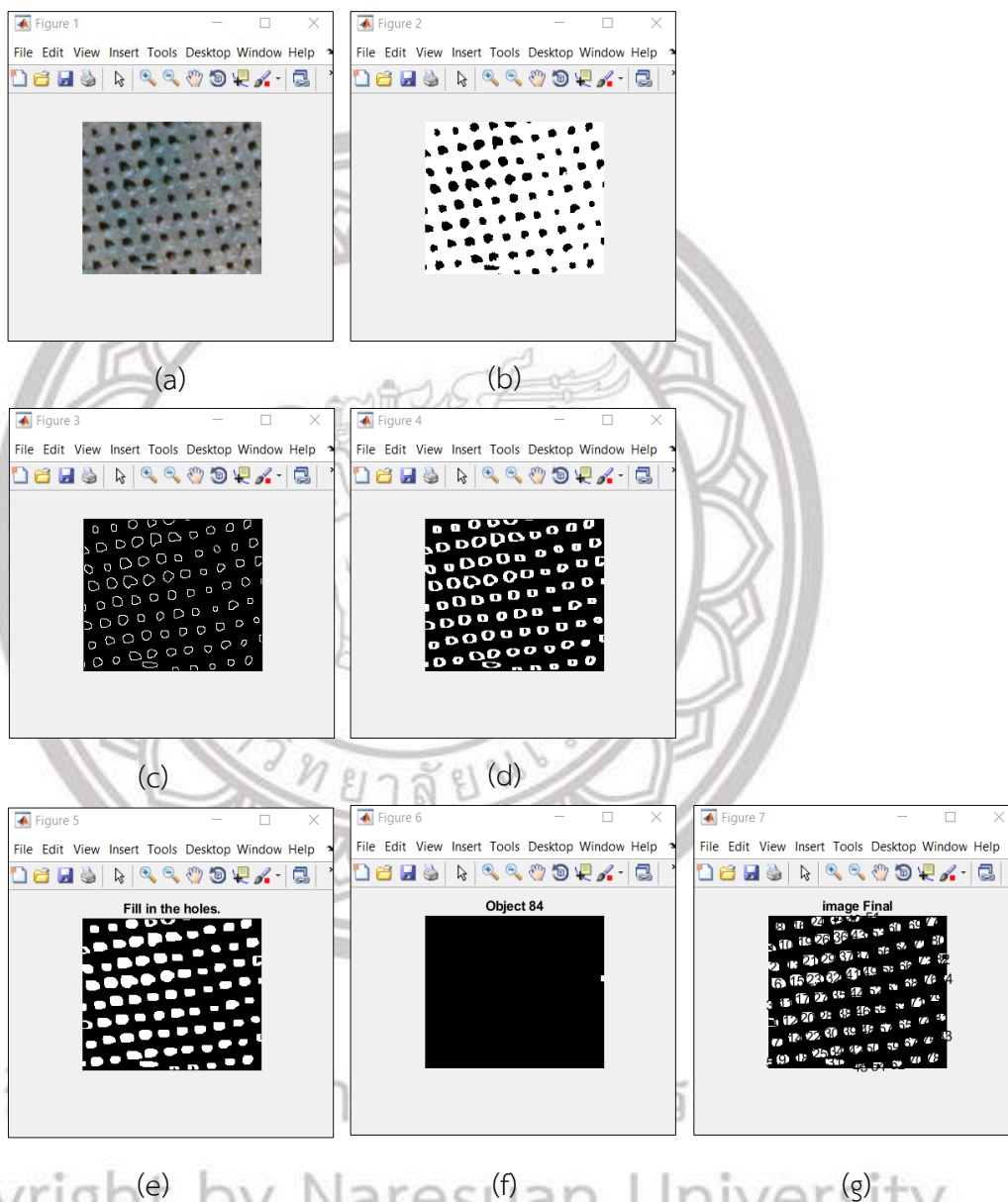
#### 2) การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติ

การพัฒนาเทคนิคหรืออัลกอริทึมที่ตรวจนับจำนวนไม้ผลอัตโนมัติในขั้นตอนนี้เป็นการทำงานเบื้องต้นด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map โดยทดลองใช้กับอัลกอริทึมที่พัฒนาในโปรแกรม MATLAB ซึ่งผู้วิจัยได้ทดลองพัฒนาเทคนิคทั้งหมด 3 เทคนิค ดังต่อไปนี้

##### (1) การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ

ขั้นตอนแรกนำภาพสีเข้าสู่โปรแกรม MATLAB โดยใช้ฟังก์ชัน `imread` ในการเรียกภาพดังภาพ 3.12(a) จากนั้นทำการแปลงภาพจากภาพสีเป็นภาพขาวดำด้วยฟังก์ชัน `im2bw` ดังภาพ 3.12(b) ขั้นตอนต่อไปทำการหาขอบของวัตถุโดยใช้ฟังก์ชัน `canny` ดังภาพ 3.12(c) เมื่อได้ขอบของวัตถุแล้วจะต้องทำการเพิ่มความหนาของเส้นขอบวัตถุที่ได้เนื่องจากขอบวัตถุบางวัตถุมีระยะห่างเกินไปทำให้ไม่สามารถตรวจนับได้โดยใช้ฟังก์ชัน `imdilate` ดังภาพ 3.12(d) ขั้นตอนต่อไปทำการเติมช่องว่างในวัตถุโดยใช้ฟังก์ชัน `holes` ดังภาพ 3.12(e) และต่อมาทำการทดสอบการนับวัตถุเพื่อตรวจสอบวัตถุว่า

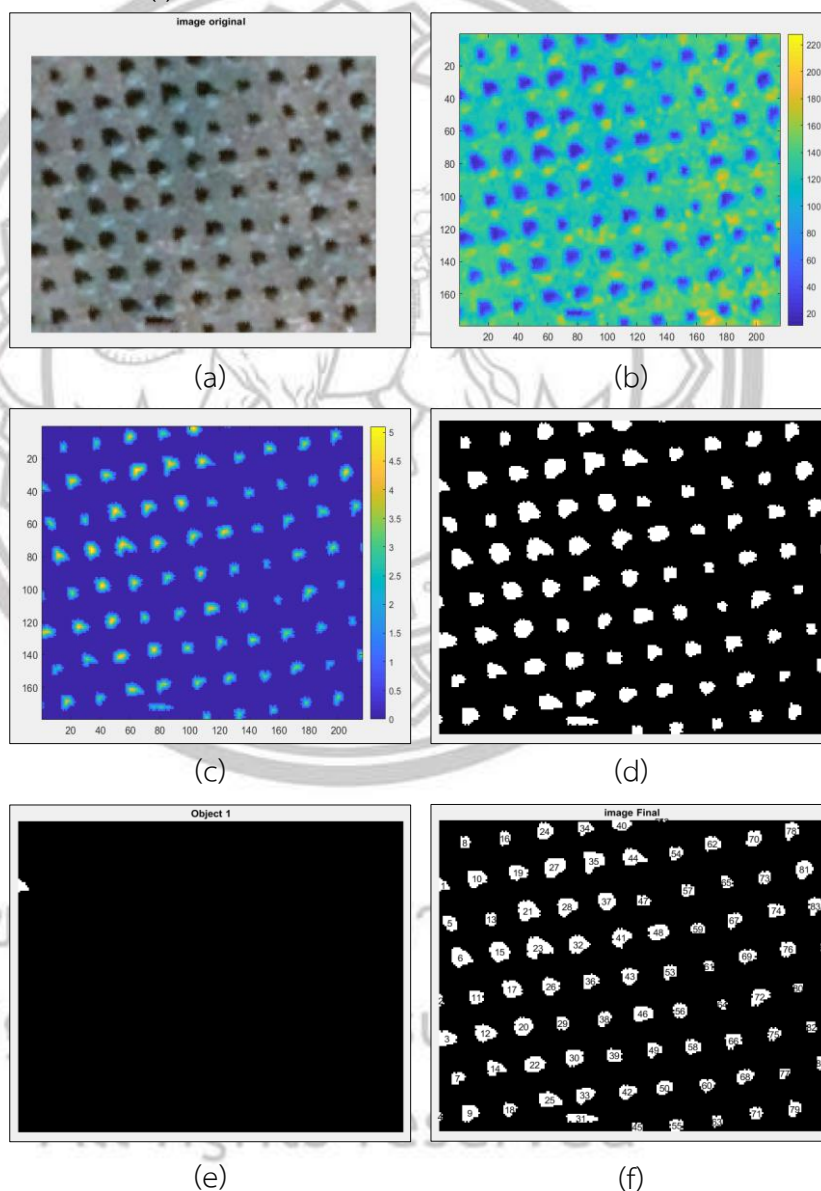
สามารถนับได้หรือไม่ดังภาพ 3.12(f) และในขั้นตอนสุดท้ายทำการวนลูปเพื่อนับวัตถุว่ามีจำนวนเท่าไรดังภาพ 3.12(g)



ภาพ 3.12 การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุด้วยโปรแกรม MATLAB: (a) การเรียกภาพด้วยฟังก์ชัน `imread`, (b) การแปลงภาพสีเป็นภาพขาวดำด้วยฟังก์ชัน `im2bw`, (c) การหาขอบของวัตถุด้วยฟังก์ชัน `canny`, (d) การเพิ่มความหนาของเส้นขอบวัตถุด้วยฟังก์ชัน `imdilate`, (e) การเติมช่องว่างในวัตถุด้วยฟังก์ชัน `holes`, (f) การทดสอบการนับวัตถุ และ (g) การวนลูปเพื่อนับวัตถุ

## (2) การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ

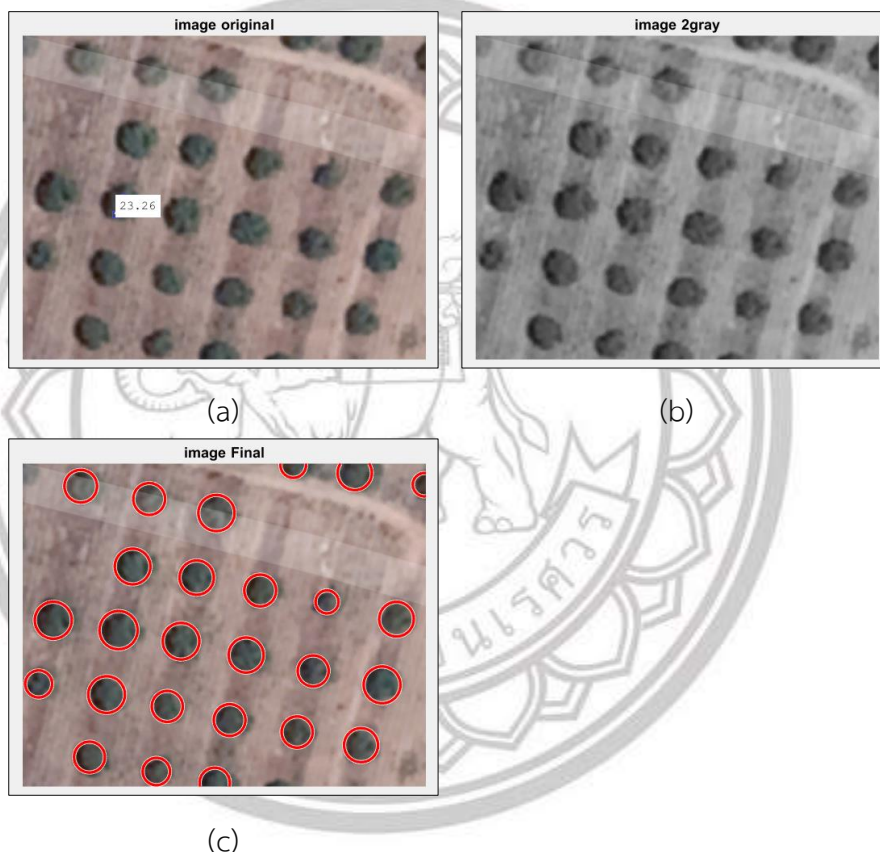
ขั้นตอนแรกนำภาพสีเข้าสู่โปรแกรม MATLAB โดยใช้ฟังก์ชัน `imread` ในการเรียกภาพดังภาพ 3.13(a) ต่อมาแสดงค่าสีของภาพโดยใช้ฟังก์ชัน `colorbar` ดังภาพ 3.13(b) จากนั้นกำหนดค่าสีตามวัตถุที่ต้องการนับดังภาพ 3.13(c) และเมื่อได้ค่าสีของวัตถุแล้วแปลงภาพสีเป็นภาพขาวดำโดยใช้ฟังก์ชัน `im2bw` ดังภาพ 3.13(d) และขั้นตอนต่อมาทำการทดสอบการนับวัตถุเพื่อตรวจสอบวัตถุว่าสามารถนับได้หรือไม่ดังภาพ 3.13(e) และขั้นตอนสุดท้ายทำการวนลูปนับวัตถุเพื่อตรวจนับวัตถุว่ามีจำนวนเท่าไรดังภาพ 3.13(f)



ภาพ 3.13 การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุด้วยโปรแกรม MATLAB: (a) การเรียกภาพด้วยฟังก์ชัน `imread`, (b) การแสดงค่าสีของภาพด้วยฟังก์ชัน `colorbar`, (c) การกำหนดค่าสีตามวัตถุที่ต้องการนับ, (d) การแปลงภาพสีเป็นภาพขาวดำด้วยฟังก์ชัน `im2bw`, (e) การทดสอบการนับวัตถุ และ (f) การวนลูปเพื่อับวัตถุ

(3) การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ

ขั้นตอนแรกนำเข้าภาพสีโดยใช้ฟังก์ชัน `imread` ดังภาพ 3.14(a) ต่อมาแปลงภาพสีเป็นภาพสีเทาโดยใช้ `rgb2gray` เนื่องจากภาพสีเทาจะทำให้สามารถระบุขอบเขตของวัตถุได้ชัดเจนดังภาพ 3.14(b) และในขั้นตอนสุดท้ายกำหนดรัศมีต่ำสุด-สูงสุดของวัตถุเพื่อทำการนับวัตถุจากรัศมีดังภาพ 3.14(c)



ภาพ 3.14 การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุด้วยโปรแกรม MATLAB: (a) การเรียกภาพด้วยฟังก์ชัน `imread`, (b) การแปลงภาพสีเป็นภาพสีเทาด้วย `rgb2gray` และ (c) การกำหนดรัศมีต่ำสุด-สูงสุดของวัตถุ

3.4.2 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

1) การแปลงนามสกุลไฟล์ภาพจาก TFF เป็น PNG

ผลลัพธ์ของกระบวนการโฟโตแกรมเมตรีจะได้ภาพออร์โธ และแบบจำลองพื้นผิวดิจิทัล โดยมีนามสกุลไฟล์ภาพเป็น TFF และไฟล์ภาพทั้งสองมีขนาดใหญ่มาก ซึ่งทำให้ไม่สามารถนำภาพเข้าสู่โปรแกรม MATLAB ได้ จึงได้ทำการแปลงนามสกุลไฟล์ภาพจาก TFF ไปเป็น PNG และลดขนาดภาพด้วยโปรแกรม ArcMap ดังภาพ 3.15



## TIFF

Export Raster Data - ขอบเขตสามเหลี่ยม2\_RGB.tif

Extent  
 Data Frame (Current)  
 Raster Dataset (Original)  
 Selected Graphics (Clipping)  Clip Inside

Spatial Reference  
 Data Frame (Current)  
 Raster Dataset (Original)

Output Raster  
 Use Renderer  Square: Cell Size (cx, cy):  0.027980000C 0.027979999  
 Force RGB Raster Size (columns, rows):  1471 2088  
 Use Colormap NoData as: 256

Name	Property
Bands	4
Pixel Depth	16 Bit
Uncompressed Size	23.43 MB
Extent (left, top, right, bottom)	( 584547.6289, 1852551.6395, 584588.7875, 1852493.2173 )
Spatial Reference	WGS_1984_UTM_Zone_47N

Location: D:\Thesis\Drone\_thesis\_2562\Drone\_thesis\_25620916\Image\_PN

Name: tif1.tif Format: TIFF

Compression Type: NONE Compression Quality (1-100): 75

[About export raster data](#) Save Cancel



## PNG

Export Raster Data - ขอบเขตสามเหลี่ยม2\_RGB.tif

Extent  
 Data Frame (Current)  
 Raster Dataset (Original)  
 Selected Graphics (Clipping)  Clip Inside

Spatial Reference  
 Data Frame (Current)  
 Raster Dataset (Original)

Output Raster  
 Use Renderer  Square: Cell Size (cx, cy):  0.09 0.09  
 Force RGB Raster Size (columns, rows):  460 649  
 Use Colormap NoData as: 256

Name	Property
Bands	3
Pixel Depth	8 Bit
Uncompressed Size	874.63 KB
Extent (left, top, right, bottom)	( 584547.9630, 1852549.8107, 584589.3271, 1852491.3774 )
Spatial Reference	WGS_1984_UTM_Zone_47N

Location: D:\Thesis\Drone\_thesis\_2562\Drone\_thesis\_25620727\Image\_PN

Name: ขอบเขตสามเหลี่ยม2 Format: PNG

Compression Type: LZ77 Compression Quality (1-100): 75

[About export raster data](#) Save Cancel

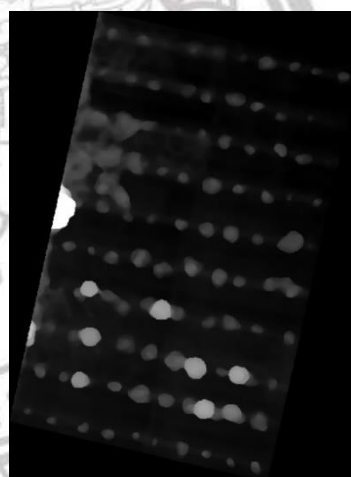
ภาพ 3.15 การแปลงนามสกุลไฟล์ภาพจาก TFF เป็น PNG

## 2) การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติ

จากภาพ 3.16 เป็นผลของการพัฒนาอัลกอริทึมในการตรวจนับจำนวนไม้ผลแบบอัตโนมัติ ขั้นตอนนี้เป็นการทดลองกับข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ โดยมีการตัดแปลงพัฒนาอัลกอริทึมเพิ่มเติมเพื่อให้ได้ผลลัพธ์ที่ดีขึ้น ซึ่งการพัฒนาอัลกอริทึมนั้นพบว่าการใช้งานตัวกรองหรือฟิลเตอร์กับภาพด้วยฟังก์ชัน imfilter นั้น ส่งผลทำให้การตรวจวัดจำนวนต้นไม้ดีขึ้น โดยทดลองกำหนดขนาดหน้าต่างของตัวกรองหรือ windowSize ซึ่งการทดลองได้แบ่งข้อมูลเพื่อนำไปวิเคราะห์จำนวน 4 ชั้นข้อมูล เพื่อให้ทราบถึงความถูกต้องของชั้นข้อมูลทั้งหมด ดังภาพ 3.16(a) คือภาพออร์โธของสวนมะนาวในจังหวัดสุโขทัย ภาพ 3.16(b) คือแบบจำลองพื้นผิวดิจิทัลของสวนมะนาวในจังหวัดสุโขทัย ภาพ 3.16(c) คือภาพออร์โธของสวนมะนาวในจังหวัดพิษณุโลก และภาพ 3.16(d) คือแบบจำลองพื้นผิวเชิงเลขของสวนมะนาวในจังหวัดพิษณุโลก



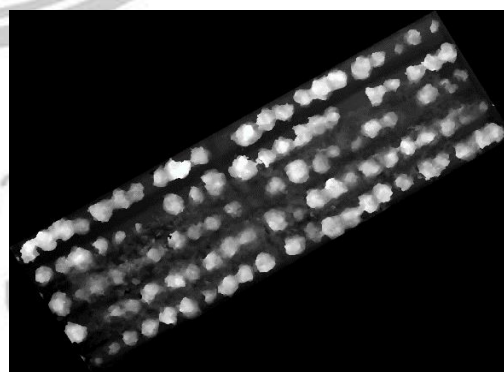
(a)



(b)



(c)



(d)

ภาพ 3.16 ภาพถ่ายทางอากาศจากอากาศยานไร้คนขับทั้ง 4 ชั้นข้อมูล: (a) ภาพออร์โธสวนมะนาวในจังหวัดสุโขทัย, (b) แบบจำลองพื้นผิวดิจิทัลสวนมะนาวในจังหวัดสุโขทัย, (c) ภาพออร์โธสวนมะนาวในจังหวัดพิษณุโลก และ (d) แบบจำลองพื้นผิวเชิงเลขสวนมะนาวในจังหวัดพิษณุโลก

### 3) พัฒนาหน้าต่างแสดงผล GUI MATLAB

อัลกอริทึมที่ได้พัฒนาอัลกอริทึมในโปรแกรม MATLAB นั้นมีความซับซ้อน ซึ่งไม่เหมาะต่อการใช้งานในกลุ่มของบุคคลทั่วไป ทำให้ผู้วิจัยได้พัฒนา GUI (Graphical User Interface) ขึ้นมาดังภาพ 3.17 ซึ่งจะประกอบไปด้วยหน้าต่างสำหรับแสดงภาพถ่ายทางอากาศทั้งหมด 4 ส่วน ที่ได้จากการประมวลผลในแต่ละขั้นตอนที่ได้พัฒนาขึ้นมา โดยในส่วนของ Processing ผู้ใช้งานสามารถที่จะกำหนดค่าพารามิเตอร์เพื่อทดสอบค่าความอ่อนไหวหรือ Sensitivity เพื่อหาค่าพารามิเตอร์ที่เหมาะสมในการสกัดและตรวจนับไม้ผลในแต่ละภาพที่ต้องการได้



Copyright by Mahasarakham University  
ภาพ 3.17 หน้าต่างแสดงผล GUI ในโปรแกรม MATLAB

All rights reserved

### 3.4.3 การตรวจสอบความถูกต้องของเทคนิคกับข้อมูลที่ได้นับจริงจากการแปลภาพถ่ายด้วยสายตา

การตรวจสอบความถูกต้องของเทคนิคกับข้อมูลที่ได้นับจริงจากการแปลภาพถ่ายด้วยสายตา จะใช้วิธีทางสถิติด้วยการหาค่าความคลาดเคลื่อนเฉลี่ยสัมบูรณ์ (Mean Absolute Error: MAE) ดังสมการ 1 และค่ารากที่สองของความคลาดเคลื่อนยกกำลังสองเฉลี่ย (Root Mean Square Error: RMSE) ดังสมการ 2 ถ้าค่าผลลัพธ์ทั้งสองสมการต่ำจะมีค่าความถูกต้องสูง แต่ถ้าค่าผลลัพธ์ทั้งสองสมการสูงก็จะมีค่าความถูกต้องต่ำ (วิภารัตน์ อัมพะวัน, 2561)

$$MAE = \frac{1}{n} \sum_{i=1}^n |Estimated\ value - Observed\ value| \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [Estimated\ value - Observed\ value]^2} \quad (2)$$

โดยที่ Estimated value คือ จำนวนไม้ผลจากการตรวจนับแบบอัตโนมัติ  
Observed value คือ จำนวนไม้ผลจากการตรวจนับจริง  
MAE คือ ความคลาดเคลื่อนเฉลี่ยสัมบูรณ์  
RMSE คือ ค่ารากที่สองของความคลาดเคลื่อนยกกำลังสองเฉลี่ย  
n คือ จำนวนข้อมูลทั้งหมด

## 3.5 การเตรียมความพร้อม

### 3.5.1 การศึกษาอากาศยานไร้คนขับ

การเตรียมความพร้อมก่อนจะทำการศึกษาในพื้นที่ศึกษาจริงนั้น ผู้วิจัยได้มีการเตรียมความพร้อมในการศึกษาข้อมูลเกี่ยวกับอากาศยานไร้คนขับ โดยทำการศึกษาข้อมูล คุณสมบัติ ระบบการทำงาน ของตัวเครื่อง และทำความเข้าใจเกี่ยวกับการบังคับอากาศยานไร้คนขับ เพื่อลดข้อผิดพลาดและอุบัติเหตุที่อาจจะเกิดขึ้นได้ นอกจากนี้ยังได้มีการทดลองบังคับอากาศยานไร้คนขับจริงบริเวณลานข้างคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ดังภาพ 3.18 อากาศยานไร้คนขับที่ทำการศึกษานั้นมี 3 ตัว คือ อากาศยานไร้คนขับ Syma X5UW 720P WIFI อากาศยานไร้คนขับ SYMA X8PRO GPS และอากาศยานไร้คนขับ DJI Phantom 4 Advance โดยอากาศยานไร้คนขับมีคุณสมบัติดังแสดงในตาราง 3.2

Copyright by Naresuan University

All rights reserved



ตาราง 3.2 คุณสมบัติของอากาศยานไร้คนขับที่ทำการศึกษา

ลำดับ	อากาศยานไร้คนขับ	คุณสมบัติ
1	Syma X5UW 720P WIFI	<ul style="list-style-type: none"> <li>- กล้องส่งสัญญาณสด 720P WIFI Micro SD</li> <li>- หน้าจอเชื่อมต่อด้วยโทรศัพท์ระบบ iOS และ android</li> <li>- วิทยุควบคุม 2.4 GHz</li> <li>- โหมดปรับความเร็ว 2 สปีด</li> <li>- โหมดล๊อคความสูง</li> <li>- โหมด one key return</li> <li>- ระยะบินไกล 200 เมตร</li> <li>- เวลาการบิน 10 นาที</li> <li>- เวลาการชาร์จพลังงาน 45 นาที</li> <li>- แบต Li-Po 3.7 V850 mAh สายชาร์จ USB</li> <li>- ใบพัดสำรอง 1 ชุด การ์ดกันใบพัด</li> <li>- ถ่านใช้กับวิทยุ ขนาด AA 4 ก้อน</li> </ul>
2	SYMA X8PRO GPS	<ul style="list-style-type: none"> <li>- ขนาดตัวเครื่อง 40x19 เซนติเมตร</li> <li>- ควบคุมการบินด้วยโหมด GPs (ดาวเทียม)</li> <li>- กล้องขนาด 720P วิทยุ</li> <li>- ปรับมุมกล้องถ่ายภาพได้ 90 องศา (จากกรีโมท)</li> <li>- พิสัยการบิน 200 เมตร</li> <li>- บินลงอัตโนมัติเมื่อแบตอ่อน</li> <li>- ระยะการบิน 10 นาที</li> <li>- แบตชาร์จ 7.4V 2000 mAh</li> </ul>
3	DJI Phantom 4 Advance	<ul style="list-style-type: none"> <li>- บินได้นานสูงสุด 30 นาที</li> <li>- ต้านทานแรงลมได้ 10 เมตร/วินาที</li> <li>- ระบบควบคุมด้วยสัญญาณดาวเทียม</li> <li>- ตัวเครื่องหนัก 1,368 กรัม</li> <li>- ระบบบินไกลถึง 7 กิโลเมตร</li> <li>- ระบบบินกลับจุดขึ้นอัตโนมัติ Return to Home</li> <li>- ความเร็วสูงสุดในโหมดสปอร์ต 72 กิโลเมตร/ชั่วโมง</li> <li>- มีระบบกันกล้องสั่น 3 แกน</li> <li>- ถ่ายภาพนิ่งด้วยความละเอียด 20 ล้านพิกเซล</li> <li>- ถ่ายวิดีโอความละเอียดระดับ 4K (3840p 60fps)</li> </ul>



ภาพ 3.18 ทดลองบังคับอากาศยานไร้คนขับ

### 3.5.2 การศึกษาการประมวลผลข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับด้วยโปรแกรม Agisoft Photoscan และ Pix4Dmapper

หลังจากเตรียมความพร้อมในเรื่องอากาศยานไร้คนขับแล้ว ผู้วิจัยก็ได้ทำการเตรียมความพร้อมในเรื่องการประมวลผลด้วยโปรแกรม Agisoft Photoscan ดังภาพ 3.19 และโปรแกรม Pix4Dmapper ดังภาพ 3.20 ในการประมวลผลภาพถ่ายทางอากาศที่ได้จากการถ่ายบริเวณลานข้างคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ทั้ง 2 โปรแกรม โดยจากการทดลองประมวลผลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับพบว่าการประมวลผลด้วยโปรแกรม Agisoft Photoscan ทั้ง 3 แบบ คือ แบบความละเอียด Low Medium และ High แสดงให้เห็นว่าการประมวลผลแบบ Low นั้นมีการสร้างจุดความสูงหนาแน่นที่ไม่ละเอียดเพียงพอต่อการนำมาวิเคราะห์เพื่อตรวจนับทรงพุ่มของต้นไม้ และจากการเปรียบเทียบระหว่าง 2 โปรแกรม พบว่าผลของการ Classification point cloud ของโปรแกรม Pix4Dmapper มีความแม่นยำมากกว่าผลจากโปรแกรม Agisoft Photoscan



ภาพ 3.19 การประมวลผลข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับด้วยโปรแกรม Agisoft Photoscan

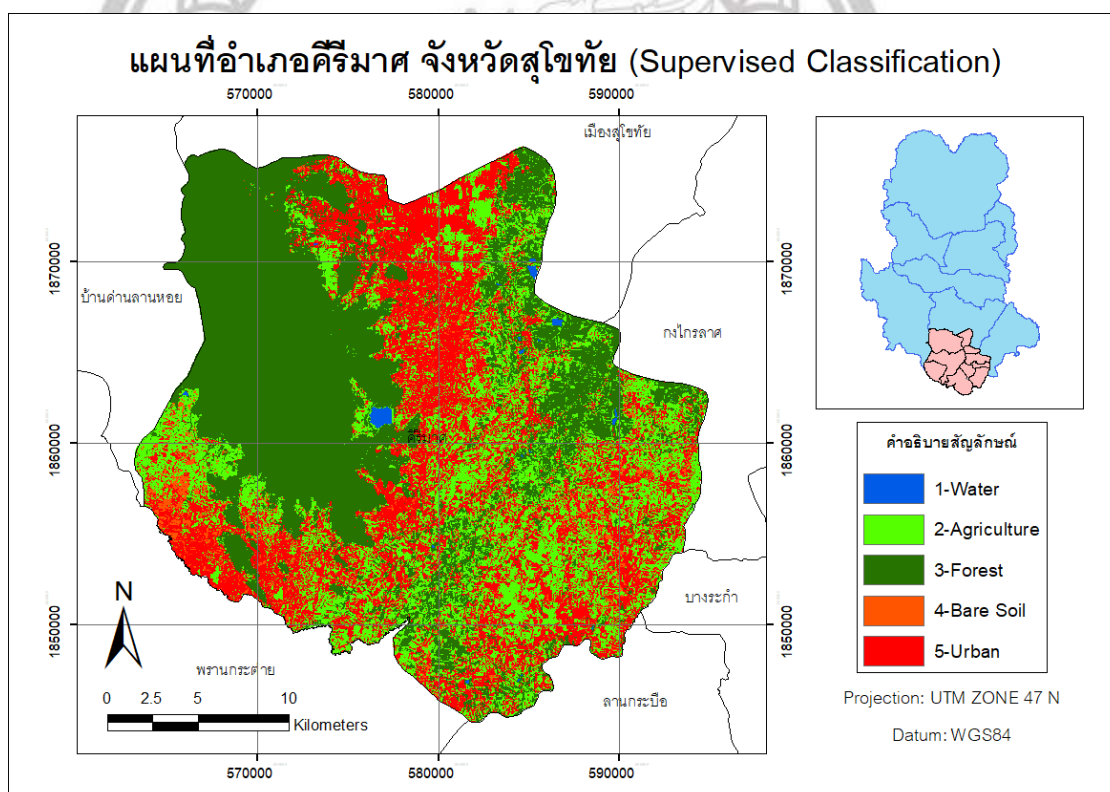


ภาพ 3.20 การประมวลผลข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับด้วยโปรแกรม Pix4Dmapper



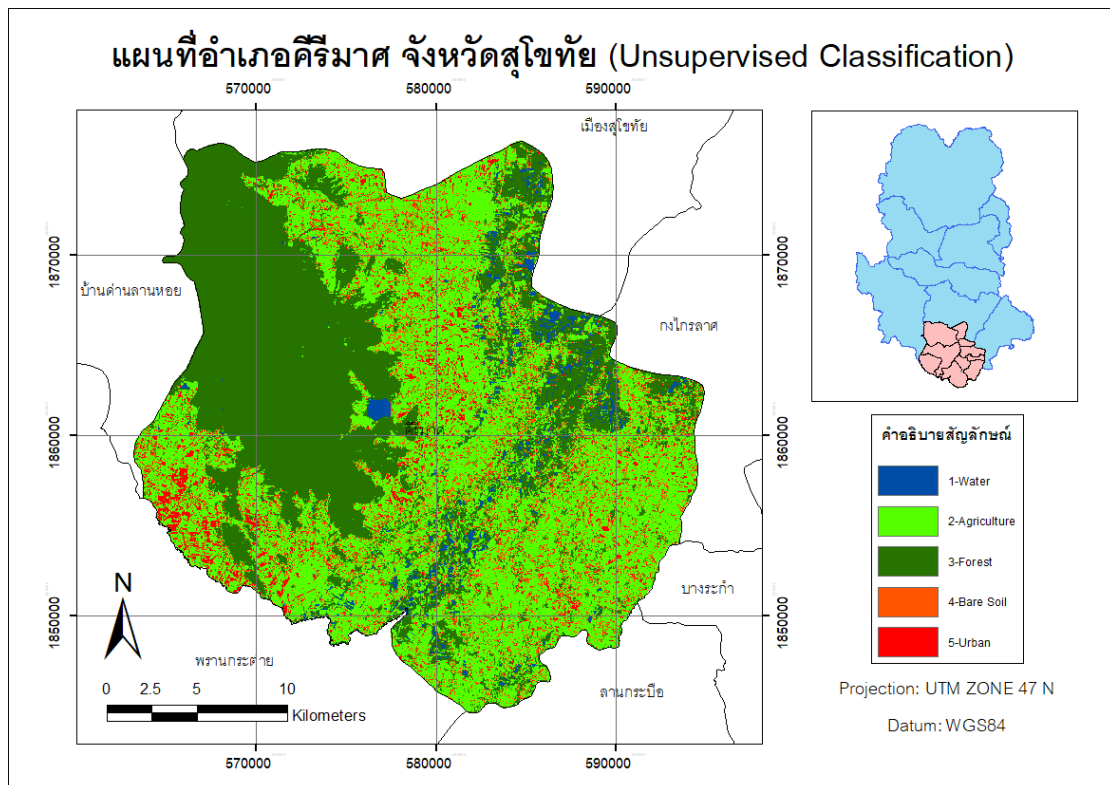
### 3.5.3 การศึกษาเรื่อง Remote sensing ในโปรแกรม ArcMap

ผู้วิจัยได้ทำการเตรียมความพร้อมในเรื่อง Remote sensing โดยศึกษาเรียนรู้ตั้งแต่พื้นฐานเบื้องต้นใหม่เกี่ยวกับทฤษฎี หลักการ และทำการทดลองในเรื่องการจำแนกสิ่งปรากฏบนภาพข้อมูลดาวเทียม ซึ่งจะประกอบด้วยการจำแนกข้อมูลแบบควบคุม (Supervised Classification) ดังภาพ 3.21 การจำแนกประเภทข้อมูลแบบไม่ควบคุม (Unsupervised Classification) ดังภาพ 3.22 และการตรวจสอบความถูกต้อง (Accuracy Assessment) โดยทำการทดลองในพื้นที่อำเภอคีรีมาศ จังหวัดสุโขทัย ผลลัพธ์ที่ได้พบว่า ความถูกต้องของการจำแนกข้อมูลแบบควบคุมมีค่ามากกว่าการจำแนกประเภทข้อมูลแบบไม่ควบคุม นอกจากนี้การศึกษาเรื่อง Remote sensing ยังเป็นพื้นฐานในการทำความเข้าใจเกี่ยวกับข้อมูลภาพ การประมวลผลภาพ ซึ่งสามารถนำไปประยุกต์ใช้กับการวิจัยในครั้งนี้ได้



ภาพ 3.21 การจำแนกประเภทข้อมูลแบบควบคุม อำเภอคีรีมาศ จังหวัดสุโขทัย





ภาพ 3.22 การจำแนกประเภทข้อมูลแบบไม่ควบคุม อำเภอคีรีมาศ จังหวัดสุโขทัย

### 3.5.4 การศึกษาการใช้โปรแกรม MATLAB

ผู้วิจัยได้เตรียมความพร้อมโดยเข้าร่วมการอบรมเรื่อง การประมวลผลภาพดิจิทัล (Digital Image Processing) ซึ่งเป็นการประมวลผลในโปรแกรม MATLAB โดย ดร.ธราวิเชษฐ์ ธิติจรูญโรจน์ จากคณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ และเทคโนโลยีสารสนเทศ มหาวิทยาลัยนเรศวร เป็นเวลา 4 วัน โดยได้เรียนรู้ถึงหลักการขั้นพื้นฐานจนถึงเทคนิควิธีการต่างๆ ที่ใช้ในการประมวลผลภาพดิจิทัล การทดลองการรู้จำตัวอักษรภาษาไทย การรู้จำตัวเลข และการรู้จำผลไม้ และผู้วิจัยยังได้ศึกษาเรียนรู้ด้วยตนเองผ่านอินเทอร์เน็ตในเรื่องการพัฒนาหน้าต่างแสดงผล GUI ด้วยโปรแกรม MATLAB ซึ่งจะเป็นผลลัพธ์ในงานวิจัยนี้ ซึ่งการศึกษาการใช้โปรแกรม MATLAB ยังเป็นพื้นฐานที่สำคัญในการวิจัยครั้งนี้เนื่องจากเป็นเครื่องมือหลักในการวิเคราะห์ข้อมูล

Copyright by Naresuan University

All rights reserved

## บทที่ 4

### ผลการวิจัย

การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ โดยบูรณาการกระบวนการโฟโตแกรมเมตรีและเทคโนโลยีการประมวลผลภาพเชิงเลข รวมทั้งได้พัฒนา Graphic User Interface (GUI) ที่ง่ายขึ้นสำหรับผู้ใช้งาน ผู้วิจัยได้กำหนดผลการวิจัยเป็น 4 ส่วน ดังนี้

4.1 การสร้างข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

4.2 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

4.3 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

4.4 การพัฒนา Graphic User Interface (GUI)

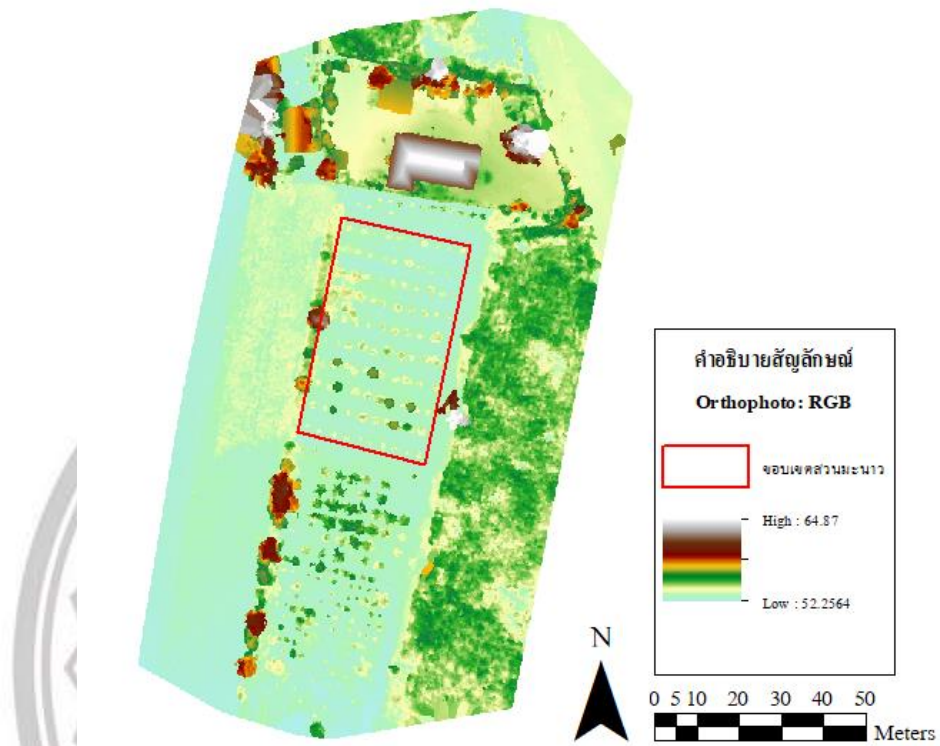
#### 4.1 การสร้างข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

##### 4.1.1 ผลการประมวลผลภาพถ่ายจากโปรแกรม Pix4Dmapper

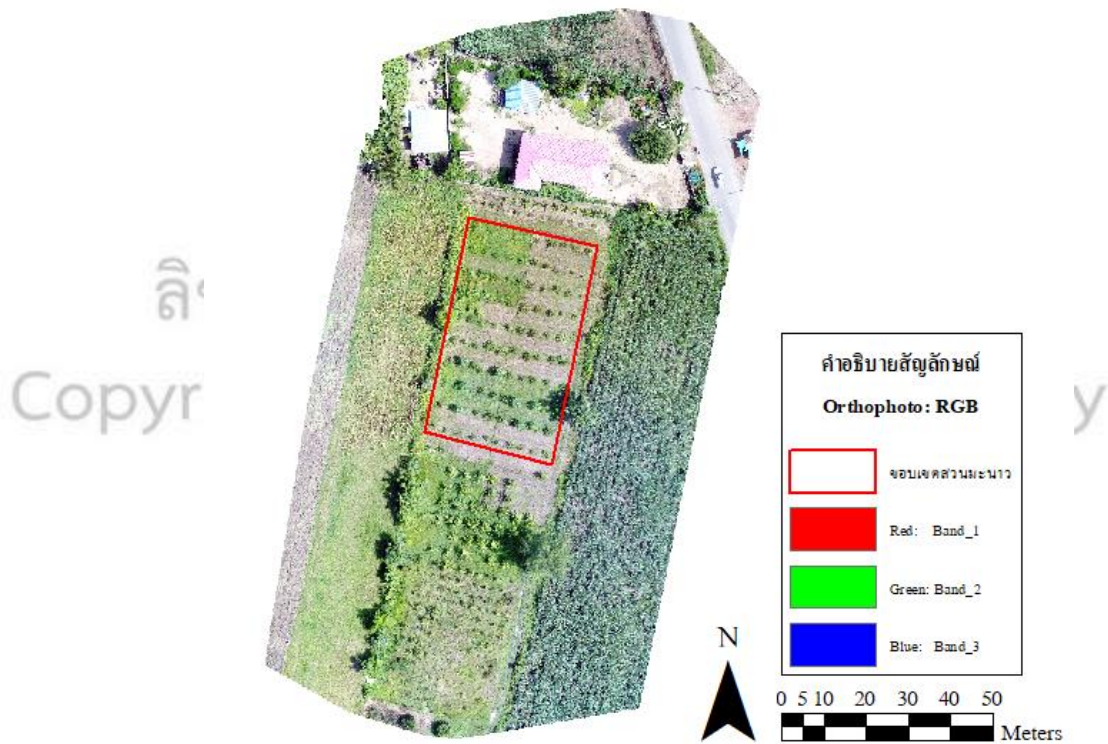
ผู้ศึกษาได้ทำการถ่ายภาพด้วยอากาศยานไร้คนขับบนสวนมะนาวในตำบลสามพวง อำเภอดำรงวิทยารัชมาศ จังหวัดสุโขทัย (ภาพ 4.1) โดยได้ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับจำนวนทั้งหมด 46 ภาพ ครอบคลุมพื้นที่ 4.5 ไร่ เมื่อนำไปประมวลผลด้วยกระบวนการโฟโตแกรมเมตรีโดยใช้โปรแกรม Pix4Dmapper พบว่าได้แบบจำลองความสูงพื้นผิวเชิงเลข (Digital Surface Model: DSM) ดังภาพ 4.2 และแผนที่ภาพถ่ายทางอากาศ (Orthophoto) ดังภาพ 4.3



ภาพ 4.1 ภาพถ่ายจากอากาศยานไร้คนขับบนสวนมะนาว ตำบลสามพวง อำเภอดำรงวิทยารัชมาศ จังหวัดสุโขทัย



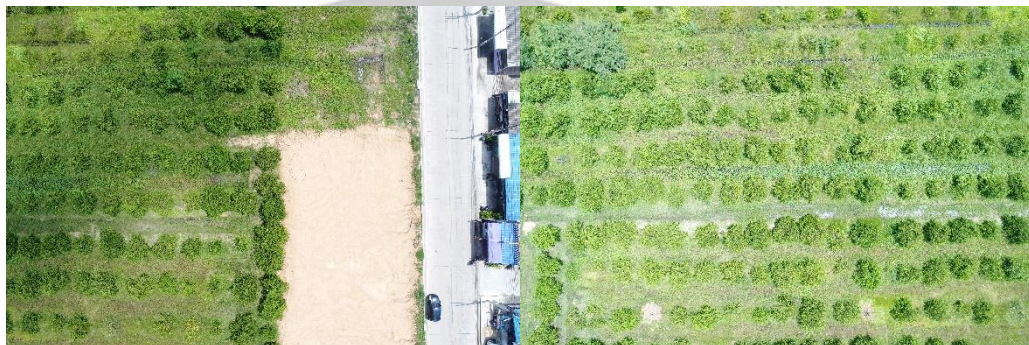
ภาพ 4.2 แบบจำลองความสูงพื้นผิวเชิงเลขของสวนมะนาว ตำบลสามพวง อำเภอกีรีมาศ จังหวัดสุโขทัย



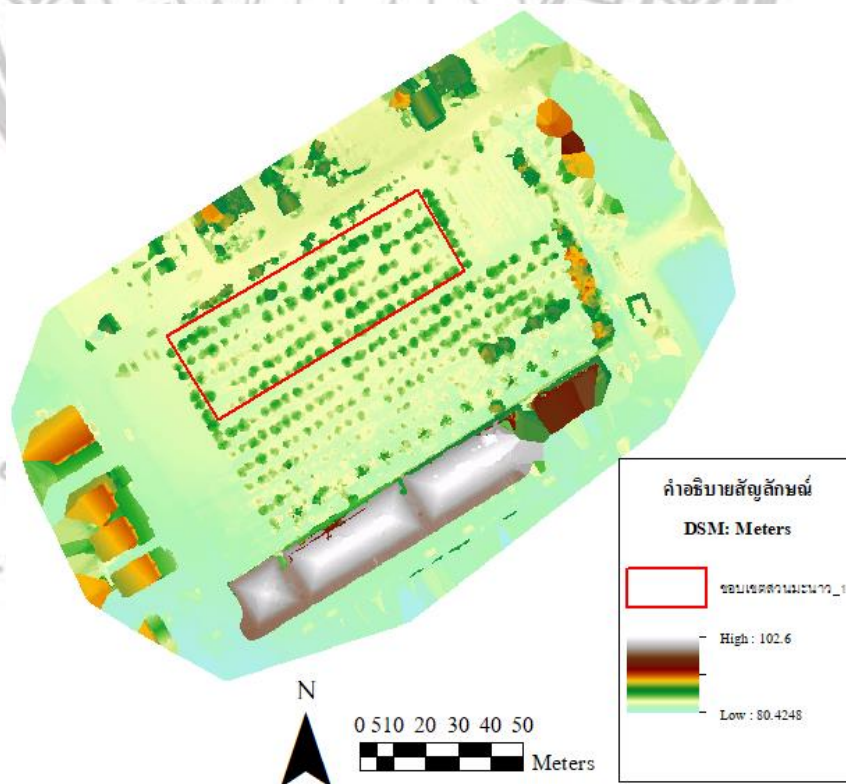
ภาพ 4.3 แผนที่ภาพถ่ายทางอากาศของสวนมะนาว ตำบลสามพวง อำเภอกีรีมาศ จังหวัดสุโขทัย



การถ่ายภาพด้วยอากาศยานไร้คนขับบนสวนมะนาวในตำบลท่าโพธิ์ อำเภอพิษณุโลก จังหวัดพิษณุโลก (ภาพ 4.4) ผู้ศึกษาได้ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับจำนวน 60 ภาพครอบคลุมพื้นที่ 11.625 ไร่ เมื่อนำไปประมวลผลด้วยกระบวนการโฟโตแกรมเมตรีโดยใช้โปรแกรม Pix4Dmapper ดังภาพ 4.5 จะได้แบบจำลองความสูงพื้นผิวเชิงเลข (Digital Surface Model: DSM) และแผนที่ภาพถ่ายทางอากาศ (Orthophoto) ดังภาพ 4.6

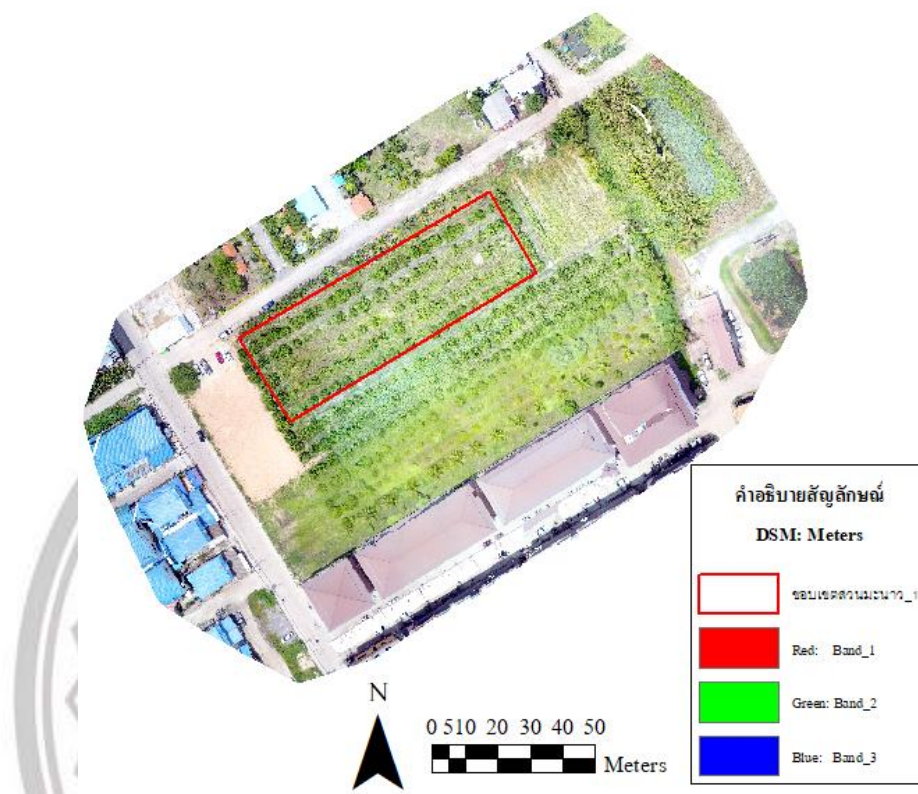


ภาพ 4.4 ภาพถ่ายจากอากาศยานไร้คนขับบนสวนมะนาว ตำบลท่าโพธิ์ อำเภอพิษณุโลก จังหวัดพิษณุโลก



ภาพ 4.5 แบบจำลองความสูงพื้นผิวเชิงเลขของสวนมะนาว ตำบลท่าโพธิ์ อำเภอพิษณุโลก จังหวัดพิษณุโลก





ภาพ 4.6 แผนที่ภาพถ่ายทางอากาศของสวนมะนาว ตำบลท่าโพธิ์ อำเภอพิชัยโลก จังหวัดพิชัยโลก

#### 4.2 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

ผลการศึกษาและพัฒนาอัลกอริทึมการตรวจนับจำนวนไม้ผลแบบอัตโนมัติจากข้อมูล Google map บนสวนส้มและสวนทุเรียน โดยมีทั้งหมด 3 วิธี ได้แก่ (1) การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (2) การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ และ (3) การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ พบว่าเปอร์เซ็นต์ความถูกต้องของวิธีที่ 1, 2 และ 3 คือ 75%, 76% และ 88% ตามลำดับ (ตาราง 4.1) และจะเห็นได้ว่าผลลัพธ์ของวิธีที่ 3 มีความถูกต้องมากที่สุด รองลงมาคือ วิธีที่ 2 และวิธีที่ 1 มีเปอร์เซ็นต์ความถูกต้องน้อยที่สุด ซึ่งจะเห็นได้ว่าผลลัพธ์ที่ได้แสดงให้เห็นถึงความสำเร็จของวิธีการที่พัฒนาขึ้นพร้อมเปอร์เซ็นต์ความถูกต้อง

#### ตาราง 4.1 เปอร์เซ็นต์ความถูกต้องของการศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผลัดโนมติ จากข้อมูล Google map

เปอร์เซ็นต์ความถูกต้อง (%)				
	1. การตรวจนับ จำนวนวัตถุ จากขอบเขตของวัตถุ	2. การตรวจนับ จำนวนวัตถุ จากค่าสีของวัตถุ	3. การตรวจนับ จำนวนวัตถุ จากรัศมีของวัตถุ	เปอร์เซ็นต์ เฉลี่ย
ส้ม	72%	78%	88%	79%
ทุเรียน	77%	73%	87%	79%
เปอร์เซ็นต์เฉลี่ย	75%	76%	88%	

#### 4.2.1 วิธีการตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map พบว่าสวนส้มให้ค่าร้อยละความถูกต้องมากที่สุดเท่ากับ 100.00 และร้อยละความถูกต้องน้อยที่สุดเท่ากับ -63.33 โดยร้อยละความถูกต้องเฉลี่ยรวมมีค่าเท่ากับ 72.13 (ตาราง 4.2) ส่วนสวนทุเรียนให้ค่าร้อยละความถูกต้องมากที่สุดเท่ากับ 100.00 และร้อยละความถูกต้องน้อยที่สุดเท่ากับ -33.33 โดยมีร้อยละความถูกต้องเฉลี่ยรวมเท่ากับ 76.55 ดังตาราง 4.3

#### 4.2.2 วิธีการตรวจนับจำนวนวัตถุจากค่าสีของวัตถุด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map พบว่าสวนส้มให้ค่าร้อยละความถูกต้องมากที่สุดเท่ากับ 100.00 และร้อยละความถูกต้องน้อยที่สุดเท่ากับ 23.33 โดยร้อยละความถูกต้องเฉลี่ยรวมมีค่าเท่ากับ 77.67 (ตาราง 4.4) ส่วนสวนทุเรียนให้ค่าร้อยละความถูกต้องมากที่สุดเท่ากับ 98.00 และร้อยละความถูกต้องน้อยที่สุดเท่ากับ 21.88 โดยร้อยละความถูกต้องเฉลี่ยรวมเท่ากับ 73.05 ดังตาราง 4.5

#### 4.2.3 วิธีการตรวจนับจำนวนวัตถุจากรัศมีของวัตถุด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map พบว่าสวนส้มให้ค่าร้อยละความถูกต้องมากที่สุดเท่ากับ 98.39 และร้อยละความถูกต้องน้อยที่สุดเท่ากับ 60.00 โดยร้อยละความถูกต้องเฉลี่ยรวมมีค่าเท่ากับ 88.16 (ตาราง 4.6) ส่วนสวนทุเรียนให้ค่าร้อยละความถูกต้องมากที่สุดเท่ากับ 100.00 และร้อยละความถูกต้องน้อยที่สุดเท่ากับ 28.57 โดยร้อยละความถูกต้องเฉลี่ยรวมมีค่าเท่ากับ 87.42 ดังตาราง 4.7

ตาราง 2.2 ผลการตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (ส้ม) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (ส้ม)			
ภาพที่	สำรวจภาคสนาม	รันโค้ด	ร้อยละความถูกต้อง
1	71	72	98.59
2	82	42	51.22
3	86	48	55.81
4	105	44	41.90
5	160	46	28.75
6	70	53	75.71
7	30	79	-63.33
8	110	99	90.00
9	70	86	77.14
10	360	177	49.17
11	72	74	97.22
12	59	42	71.19
13	116	77	66.38
14	56	51	91.07
15	106	70	66.04
16	50	54	92.00
17	50	50	100.00
18	80	101	73.75
19	120	124	96.67
20	62	56	90.32
21	85	78	91.76
22	58	61	94.83
23	69	66	95.65
24	72	84	83.33
25	55	27	49.09
26	75	46	61.33
27	76	74	97.37
28	110	69	62.73
29	50	55	90.00
30	76	85	88.16
		รวม	72.13

ตาราง 4.3 ผลการตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (ทุเรียน) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (ทุเรียน)			
ภาพที่	สำรวจภาคสนาม	รันโค้ด	ร้อยละความถูกต้อง
1	32	46	56.25
2	28	27	96.43
3	28	39	60.71
4	37	43	83.78
5	40	44	90.00
6	60	130	-16.67
7	57	99	26.32
8	40	33	82.50
9	22	35	40.91
10	30	36	80.00
11	23	25	91.30
12	30	33	90.00
13	23	31	65.22
14	75	74	98.67
15	60	56	93.33
16	56	55	98.21
17	55	53	96.36
18	54	58	92.59
19	42	43	97.62
20	50	47	94.00
21	34	36	94.12
22	58	62	93.10
23	63	63	100.00
24	32	37	84.38
25	45	53	82.22
26	50	47	94.00
27	31	39	74.19
28	40	43	92.50
29	12	28	-33.33
30	45	44	97.78
		รวม	76.55



ตาราง 4.4 ผลการตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (ส้ม) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (ส้ม)			
ภาพที่	สำรวจภาคสนาม	รันโค้ด	ร้อยละความถูกต้อง
1	71	90	73.24
2	82	138	31.71
3	86	121	59.30
4	105	134	72.38
5	160	247	45.63
6	70	108	45.71
7	30	53	23.33
8	110	170	45.45
9	70	79	87.14
10	360	477	67.50
11	72	75	95.83
12	59	75	72.88
13	116	108	93.10
14	56	63	87.50
15	106	119	87.74
16	50	60	80.00
17	50	51	98.00
18	80	84	95.00
19	120	115	95.83
20	62	75	79.03
21	85	91	92.94
22	58	59	98.28
23	69	83	79.71
24	72	74	97.22
25	55	53	96.36
26	75	101	65.33
27	76	87	85.53
28	110	89	80.91
29	50	50	100.00
30	76	78	97.37
		รวม	77.67

ตาราง 4.5 ผลการตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (ทุเรียน) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (ทุเรียน)			
ภาพที่	สำรวจภาคสนาม	รันโค้ด	ร้อยละความถูกต้อง
1	32	57	21.88
2	28	40	57.14
3	28	46	35.71
4	37	46	75.68
5	40	42	95.00
6	60	68	86.67
7	57	101	22.81
8	40	33	82.50
9	22	26	81.82
10	30	42	60.00
11	23	30	69.57
12	30	24	80.00
13	23	32	60.87
14	75	100	66.67
15	60	90	50.00
16	56	79	58.93
17	55	59	92.73
18	54	59	90.74
19	42	43	97.62
20	50	52	96.00
21	34	37	91.18
22	58	64	89.66
23	63	64	98.41
24	32	44	62.50
25	45	70	44.44
26	50	49	98.00
27	31	30	96.77
28	40	46	85.00
29	12	15	75.00
30	45	59	68.89
		รวม	73.07

ตาราง 4.6 ผลการตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (ส้ม) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (ส้ม)			
ภาพที่	สำรวจภาคสนาม	รันโค้ด	ร้อยละความถูกต้อง
1	71	64	90.14
2	82	69	84.15
3	86	68	79.07
4	105	98	93.33
5	160	132	82.50
6	70	63	90.00
7	30	28	93.33
8	110	66	60.00
9	70	61	87.14
10	360	336	93.33
11	72	79	90.28
12	59	63	93.22
13	116	138	81.03
14	56	59	94.64
15	106	119	87.74
16	50	52	96.00
17	50	56	88.00
18	80	66	82.50
19	120	129	92.50
20	62	63	98.39
21	85	76	89.41
22	58	52	89.66
23	69	67	97.10
24	72	67	93.06
25	55	61	89.09
26	75	60	80.00
27	76	59	77.63
28	110	99	90.00
29	50	54	92.00
30	76	68	89.47
		รวม	88.16

ตาราง 4.7 ผลการตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (ทุเรียน) ด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

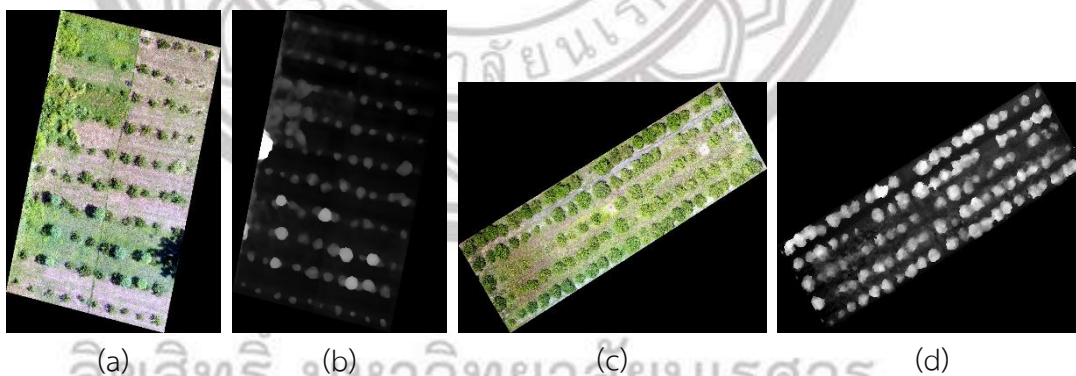
การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (ทุเรียน)			
ภาพที่	สำรวจภาคสนาม	รันโค้ด	ร้อยละความถูกต้อง
1	32	32	100.00
2	28	38	64.29
3	28	48	28.57
4	37	35	94.59
5	40	35	87.50
6	60	61	98.33
7	57	48	84.21
8	40	42	95.00
9	22	25	86.36
10	30	37	76.67
11	23	23	100.00
12	30	34	86.67
13	23	18	78.26
14	75	68	90.67
15	60	57	95.00
16	56	59	94.64
17	55	46	83.64
18	54	62	85.19
19	42	37	88.10
20	50	53	94.00
21	34	40	82.35
22	58	52	89.66
23	63	65	96.83
24	32	37	84.38
25	45	52	84.44
26	50	49	98.00
27	31	30	96.77
28	40	41	97.50
29	12	14	83.33
30	45	44	97.78
		รวม	87.42



#### 4.3 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

ผลการพัฒนาอัลกอริทึมในการตรวจนับจำนวนไม้ผลแบบอัตโนมัติดังภาพ 4.7 ในขั้นตอนนี้เป็นการทดลองกับข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ โดยได้มีการตัดแปลงพัฒนาโค้ดเพิ่มเติมเพื่อให้ได้ผลลัพธ์ที่ดีขึ้น ซึ่งจากการพัฒนาโค้ดพบว่าการใช้งานตัวกรองหรือฟิลเตอร์กับภาพด้วยฟังก์ชัน imfilter ส่งผลทำให้การตรวจนับจำนวนต้นไม้มีประสิทธิภาพที่ดีมากยิ่งขึ้น ผู้วิจัยได้ทดลองกำหนดขนาดหน้าต่างของตัวกรองหรือ windowSize โดยแบ่งข้อมูลเพื่อนำไปวิเคราะห์ทั้งหมด 4 ชั้นข้อมูล ดังภาพ 4.7(a) คือ ภาพออร์โธของสวนมะนาวในจังหวัดสุโขทัย ภาพ 4.7(b) คือ แบบจำลองพื้นผิวดิจิทัลของสวนมะนาวในจังหวัดสุโขทัย ภาพ 4.11(c) คือ ภาพออร์โธของสวนมะนาวในจังหวัดพิษณุโลก และภาพ 4.7(d) คือ แบบจำลองพื้นผิวเชิงเลขของสวนมะนาวในจังหวัดพิษณุโลก

ผลการศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูลภาพถ่ายทางอากาศด้วยอากาศยานไร้คนขับ พบว่าเปอร์เซ็นต์ความถูกต้องของวิธีที่ 1, 2 และ 3 คือ 74%, 79% และ 92% ตามลำดับ โดยวิธีที่ 3 ให้ค่าความถูกต้องในแต่ละการทดลองมากที่สุดดังตาราง 4.8 ซึ่งการนำ DSM มาใช้จำแนกจะทำให้สามารถแยกไม้ผลออกจากพื้นหลังได้ง่ายขึ้นเนื่องจากมีความแตกต่างของความสูงอย่างเห็นได้ชัด ส่วนการใช้ภาพออร์โธอย่างเดียวยังคงเป็นปัญหาเนื่องจากถ้าสวนนั้นมีค่าสีของพื้นหลังใกล้เคียงกับค่าสีของทรงพุ่มจะทำให้อัลกอริทึมไม่สามารถแยกพื้นหลังออกได้ ทำให้การสร้างเส้นขอบของวัตถุที่ใช้ canny มีความผิดพลาด



ภาพ 4.7 ภาพถ่ายทางอากาศจากอากาศยานไร้คนขับทั้ง 4 ชั้นข้อมูล: (a) ภาพออร์โธของสวนมะนาวในจังหวัดสุโขทัย, (b) แบบจำลองพื้นผิวดิจิทัลของสวนมะนาวในจังหวัดสุโขทัย, (c) ภาพออร์โธของสวนมะนาวในจังหวัดพิษณุโลก และ (d) แบบจำลองพื้นผิวเชิงเลขของสวนมะนาวในจังหวัดพิษณุโลก

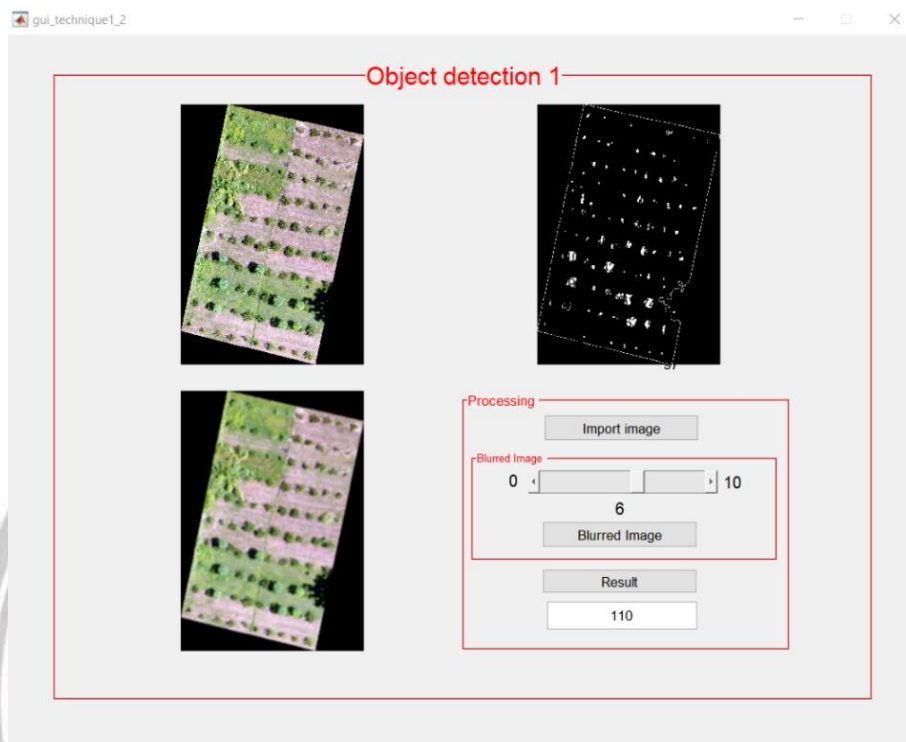
ตาราง 4.8 เปอร์เซ็นต์ความถูกต้องของการศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูลภาพถ่ายทางอากาศด้วยอากาศยานไร้คนขับ

Area	Data	เปอร์เซ็นต์ความถูกต้อง (%)			เปอร์เซ็นต์เฉลี่ย (%)
		1. การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ	2. การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ	3. การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ	
สวนมะนาว	lemon	77%	69%	90%	79%
สุโขทัย	(RGB)				
สวนมะนาว	lemon	86%	98%	99%	94%
สุโขทัย	(DSM)				
สวนมะนาว	lemon	51%	72%	78%	67%
พิษณุโลก	(RGB)				
สวนมะนาว	lemon	80%	78%	99%	86%
พิษณุโลก	(DSM)				
เปอร์เซ็นต์เฉลี่ย (%)		74%	79%	92%	

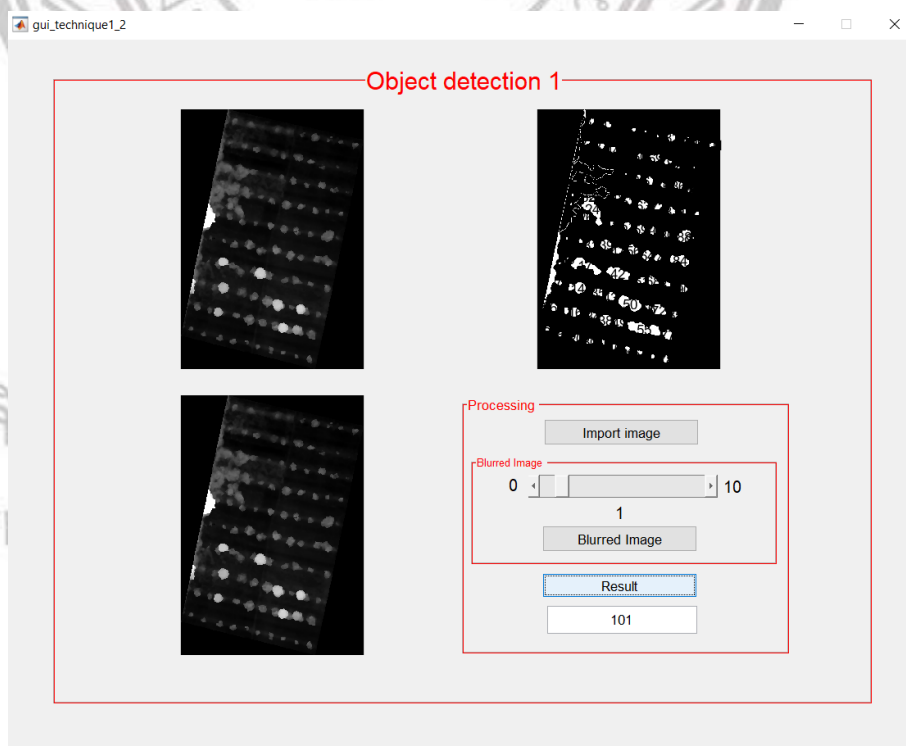
#### 4.4 การพัฒนา Graphic User Interface (GUI)

โค้ดที่ได้พัฒนาอัลกอริทึมในโปรแกรม MATLAB นั้นมีความซับซ้อนซึ่งไม่เหมาะต่อการใช้งานในกลุ่มบุคคลทั่วไป ผู้วิจัยจึงได้พัฒนา Graphical User Interface หรือ GUI ขึ้นมา ซึ่งประกอบด้วยหน้าต่างสำหรับแสดงการทำงานของอัลกอริทึมที่ได้จากการประมวลผลในแต่ละขั้นตอนที่พัฒนาขึ้นมาจำนวน 3 ส่วน ได้แก่ ส่วนแสดงภาพ ส่วนประมวลผล (Processing) และส่วนของผลลัพธ์ โดยส่วนของ Processing ผู้ใช้งานสามารถกำหนดค่าพารามิเตอร์เพื่อทดสอบค่าความอ่อนไหว (Sensitivity) ได้โดยหาค่าพารามิเตอร์ที่เหมาะสมในการสกัดและตรวจนับต้นไม้ผลในแต่ละภาพที่ต้องการด้วย GUI การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุของวิธีการนี้มีพารามิเตอร์ 1 ตัว คือ ค่า windowSize ในฟังก์ชัน imfilter ดังภาพ 4.8 และภาพ 4.9 สำหรับ GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุในวิธีการนี้มีพารามิเตอร์ 2 ตัว คือ ค่า windowSize ในฟังก์ชัน imfilter และค่าสีของวัตถุดังภาพ 4.10 และภาพ 4.11 และ GUI การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุในวิธีการนี้มีพารามิเตอร์ 3 ตัว คือ ค่า windowSize ในฟังก์ชัน imfilter รัศมีสูงสุด-ต่ำสุดของวัตถุดังภาพ 4.12 และภาพ 4.13

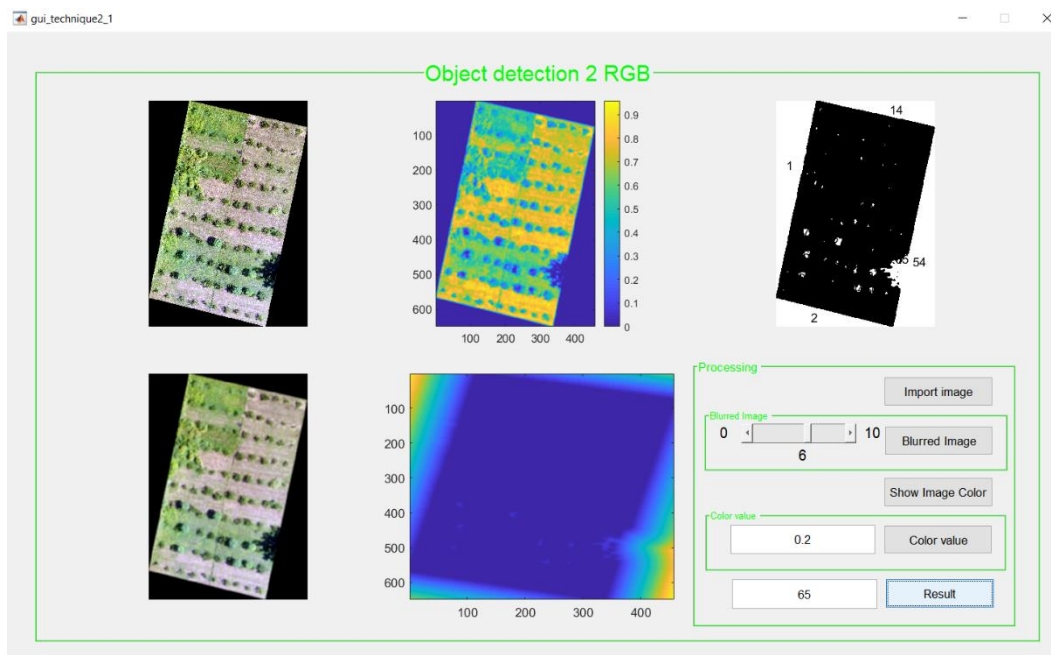
All rights reserved



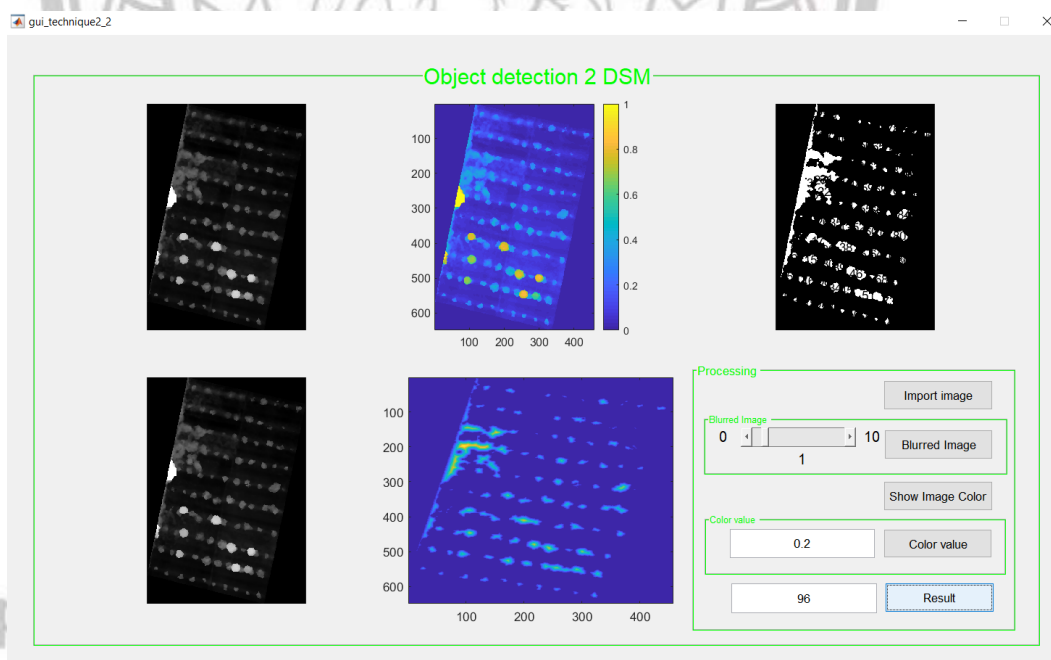
ภาพ 4.8 GUI การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (RGB) สวนมะนาว จังหวัดสุโขทัย



ภาพ 4.9 GUI การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ (DSM) สวนมะนาว จังหวัดสุโขทัย

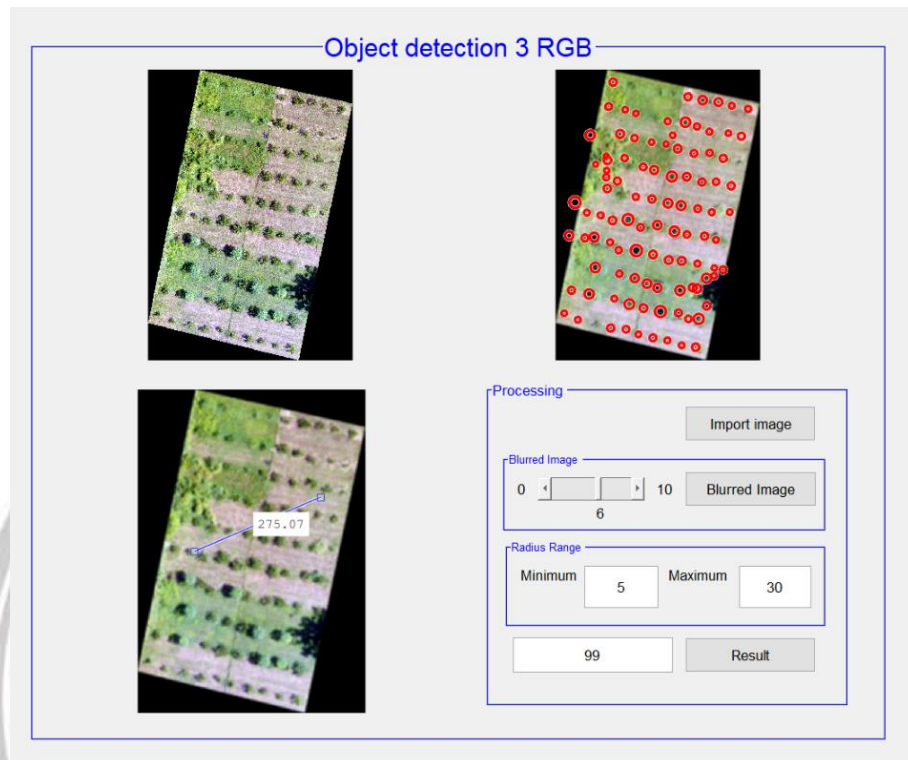


ภาพ 4.10 GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (RGB) สวนมะนาว จังหวัดสุโขทัย



ภาพ 4.11 GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (DSM) สวนมะนาว จังหวัดสุโขทัย





ภาพ 4.12 GUI การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (RGB) สวนมะนาว จังหวัดสุโขทัย



ภาพ 4.13 GUI การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (DSM) สวนมะนาว จังหวัดสุโขทัย

## บทที่ 5

### สรุปและอภิปรายผลการศึกษา

การศึกษาครั้งนี้มุ่งเน้นที่การศึกษาและพัฒนาอัลกอริทึมในการตรวจนับจำนวนไม้ผลแบบอัตโนมัติโดยการประยุกต์ใช้ข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ โดยบูรณาการระหว่างกระบวนการโฟโตแกรมเมตรีและเทคโนโลยีการประมวลผลภาพเชิงเลข นอกจากนี้ยังได้พัฒนา Graphic User Interface (GUI) เพื่อให้ง่ายต่อผู้ใช้งาน ซึ่งสามารถสรุปผลการศึกษาได้ดังนี้

#### 5.1 อภิปรายผล

##### 5.1.1 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายดาวเทียมจาก Google map

###### 1) ลักษณะข้อมูลภาพถ่ายดาวเทียมจาก Google map

ข้อมูลภาพถ่ายดาวเทียมจาก Google map ที่ได้ทำการสุ่มพื้นที่โดยส่วนใหญ่มีลักษณะของแปลงปลูกที่เป็นระเบียบจัดเรียงเป็นแนว มีระยะห่างระหว่างต้นพอสมควร มีวัชพืชปกคลุมดินน้อย มีขนาดของทรงพุ่มที่ใหญ่สามารถมองเห็นได้ชัดเจน และข้อสำคัญคือภาพถ่ายดาวเทียมจาก Google map มีรายละเอียดของภาพที่น้อย ทำให้การตรวจนับจำนวนไม้ผลแบบอัตโนมัติมีประสิทธิภาพในระดับดี

###### 2) การตรวจสอบความถูกต้องของวิธีการกับข้อมูลที่น่าเชื่อถือจริงจากการแปลภาพถ่ายด้วยสายตา

ผลการศึกษาและพัฒนาอัลกอริทึมการตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูล Google map พบว่ามีเปอร์เซ็นต์ความถูกต้องของวิธีที่ 1, 2 และ 3 คือ 75%, 76% และ 88% ตามลำดับ และผลลัพธ์ของวิธีที่ 3 มีเปอร์เซ็นต์ความถูกต้องมากที่สุด รองลงมาคือ วิธีที่ 2 และวิธีที่ 1 มีเปอร์เซ็นต์ความถูกต้องน้อยที่สุด ซึ่งจะเห็นได้ว่าผลลัพธ์ที่ได้แสดงให้เห็นถึงความสำเร็จของอัลกอริทึมการพัฒนาขึ้น อีกทั้งมีเปอร์เซ็นต์ความถูกต้องของการวิเคราะห์เกิน 70% แม้ว่าวิธีที่ 1 และวิธีที่ 2 จะมีเปอร์เซ็นต์ความถูกต้องน้อย แต่ความสามารถในการแยกสัญญาณยังคงอยู่ในระดับดี อย่างไรก็ตามผลลัพธ์ที่ได้ยังมีความผิดพลาดอยู่เนื่องจากใช้ข้อมูลภาพอย่างเดียวในการจำแนกทรงพุ่มออกจากพื้นหลัง และอัลกอริทึมที่ได้พัฒนาขึ้นยังไม่สมบูรณ์ จึงส่งผลให้ผลลัพธ์ยังมีความผิดพลาด

ผลลัพธ์ที่ได้แตกต่างจาก Gistda Academy (2015) ที่ได้ร้อยละความถูกต้องของการตรวจนับจำนวนต้นปาล์มมากกว่า 90 เนื่องจากมีการใช้ภาพถ่ายดาวเทียม Quickbird และ WorldView-2 ที่มีความละเอียดสูง และยังมีอัลกอริทึมในการตรวจนับจำนวนไม้ผลที่ดี จึงทำให้ร้อยละความถูกต้องมีค่าสูง

### 5.1.2 การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

#### 1) ลักษณะข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

ภาพถ่ายจากอากาศยานไร้คนขับของสวนมะนาวจังหวัดสุโขทัยและจังหวัดพิษณุโลก มีความสว่างที่ไม่สม่ำเสมอทั้งทั้งสวน บางภาพมีความสว่างมาก บางภาพมีความสว่างน้อย เพราะมีเมฆเคลื่อนผ่านขณะที่ถ่ายภาพ แต่ความสว่างถือว่าเพียงพอต่อการนำมาประมวลผล

ลักษณะแปลงปลูกของสวนมะนาวในจังหวัดสุโขทัย มีการปลูกที่เป็นระเบียบจัดเรียงเป็นแนว มีระยะห่างระหว่างต้นพอสมควร แต่มีวัชพืชปกคลุมดินอยู่ และยังมีทรงพุ่มที่ขนาดเล็กเนื่องจากอายุการปลูกยังน้อย ส่วนสวนมะนาวในจังหวัดพิษณุโลก มีวัชพืชปกคลุมดินอยู่บ้าง แต่ทรงพุ่มมีขนาดใหญ่กว่าเนื่องจากอายุการปลูกที่มากกว่า ซึ่งทรงพุ่มที่มีขนาดใหญ่ก็เป็นอีกปัญหาหนึ่งเนื่องจากเมื่อมองในมุมสูงจะทำให้เห็นว่าบางต้นมีทรงพุ่มที่ติดกัน และจะส่งผลโดยตรงทำให้ความถูกต้องมีค่าน้อยลง ด้วยเหตุนี้ผู้วิจัยจึงได้ทำการเลือกพื้นที่บางส่วนจากสวนทั้งหมดมาประมวลผลเพื่อให้ผลลัพธ์มีความถูกต้องที่มากขึ้น

#### 2) การตรวจสอบความถูกต้องของวิธีการกับข้อมูลที่นับได้จริงจากการแปลภาพถ่ายด้วยสายตา

ผลการพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ พบว่าร้อยละความถูกต้องของวิธีที่ 1, 2 และ 3 คือ 74%, 79% และ 92% ตามลำดับ พบว่าวิธีที่ 3 ให้ค่าความถูกต้องในแต่ละการทดลองมากที่สุด โดยที่การนำ DSM มาใช้จำแนกทำให้สามารถแยกต้นไม้ผลออกจากพื้นหลังได้ง่ายขึ้น เนื่องจากมีความแตกต่างของความสูงอย่างชัดเจน ส่วนการใช้ภาพออร์โธอย่างเดียวยังคงเป็นปัญหาเนื่องจากพื้นที่สวนมีค่าสีของพื้นหลังใกล้เคียงกับค่าสีของทรงพุ่มจะทำให้อัลกอริทึมไม่สามารถแยกพื้นหลังออกได้ทำให้การสร้างเส้นขอบของวัตถุที่ใช้ canny มีความผิดพลาด

ผลลัพธ์ที่ได้สอดคล้องกับงานของ Dilek (2018) ที่ได้พัฒนาอัลกอริทึมในการตรวจวัดจำนวนต้นมะนาวจากข้อมูลภาพถ่ายจากอากาศยานไร้คนขับซึ่งพบว่าความถูกต้องของการตรวจนับจำนวนต้นมะนาวที่ใช้ข้อมูลจาก DSM มีค่ามากกว่าการใช้ภาพออร์โธ

### 5.2 สรุป

การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับของสวนมะนาวในจังหวัดสุโขทัย และสวนมะนาวในจังหวัดพิษณุโลก โดยการประมวลผลโฟโตแกรมเมตรีด้วยโปรแกรม Pix4Dmapper เพื่อให้ได้ภาพออร์โธ และภาพแบบจำลองพื้นผิวดิจิตอล DSM แล้วทำการประมวลผลภาพดิจิทัลด้วยโปรแกรม MATLAB เพื่อตรวจนับจำนวนไม้ผลอัตโนมัติ สามารถสรุปผลได้ ดังนี้

1. การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูลภาพจาก Google map เพื่อเป็นแนวทางในการพัฒนาอัลกอริทึมสำหรับภาพถ่ายจากอากาศยานไร้คนขับ พบว่าค่าความถูกต้องในการตรวจนับยังมีค่าต่ำ เนื่องจากเป็นการใช้ข้อมูลภาพอย่างเดียวในการจำแนกพื้นหลังกับทรงพุ่มของต้นไม้

2. อัลกอริทึมที่ถูกพัฒนาเพื่อการตรวจนับจำนวนไม้ผลัดโนมิตีจากข้อมูลภาพถ่ายทางอากาศด้วยอากาศยานไร้คนขับมีร้อยละความถูกต้องเฉลี่ยสูงเมื่อเทียบกับภาพจาก Google map เนื่องจากมีการพัฒนาโค้ดเพิ่มเติม และยังสามารถนำข้อมูล DSM เข้ามาประกอบในการช่วยจำแนกพื้นหลังภาพกับทรงพุ่ม ทำให้มีการตรวจนับจำนวนไม้ผลัดโนมิตีที่ดีขึ้น

อย่างไรก็ตามยังมีข้อจำกัดบางประการ โดยพบว่าหากภาพที่ใช้ในการจำแนกมีลักษณะทรงพุ่มที่ไม่ชัดเจน หรือพื้นดินมีวัชพืชปกคลุม จะส่งผลให้การตรวจนับจำนวนไม้ผลัดโนมิตีมีความผิดพลาด หากมีการใช้เซนเซอร์ช่วงคลื่นอินฟราเรดใกล้ร่วมกับช่วงคลื่นอินฟราเรดความร้อนจะทำให้การจำแนกทรงพุ่มของพืชสวนกับพื้นดินนั้นมีความถูกต้องมากยิ่งขึ้น

### 5.3 ปัญหาและอุปสรรค

1. ข้อมูลภาพถ่ายดาวเทียมจาก Google map มีความละเอียดที่ต่ำ ซึ่งอาจเป็นสาเหตุหนึ่งที่ทำให้เปอร์เซ็นต์ความถูกต้องมีค่าน้อย
2. สภาพภูมิอากาศในขณะที่ถ่ายภาพสวนมะนาวในจังหวัดสุโขทัย มีความแปรปรวน ลมแรง และมีเมฆมาก ซึ่งเป็นอุปสรรคในการถ่ายภาพเป็นอย่างมาก
3. พื้นที่ตัวอย่างทั้ง 2 สวน มีลักษณะแปลงปลูกที่ยังไม่ดีพอ เนื่องจากแปลงปลูกมีวัชพืชปกคลุมดิน ซึ่งเป็นปัญหาอย่างมากในการประมวลผล

### 5.4 ข้อเสนอแนะ

1. ควรเลือกพื้นที่ตัวอย่างที่มีลักษณะแปลงปลูกที่ดี ไม่มีวัชพืชปกคลุมดิน ปลูกเป็นระเบียบจัดเรียงเป็นแนว มีระยะห่างระหว่างต้น และมีทรงพุ่มขนาดใหญ่แยกออกจากกันชัดเจน
2. ในการลงพื้นที่ถ่ายภาพ ควรเลือกถ่ายในช่วงที่ไม่มีลมแรง และท้องฟ้าแจ่มใส ไม่มีเมฆ
3. ควรนำปัจจัยนอกจากข้อมูล DSM และภาพออร์โธมาประมวลผลร่วมด้วย เพื่อผลลัพธ์ที่หลากหลาย และความถูกต้องที่มากยิ่งขึ้น

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved





บรรณานุกรม

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## บรรณานุกรม

- กิตติวัฒน์ นิมเกิดผล. (2549). การตรวจเอกสารแบบตัวเลือกด้วยการประมวลผลภาพ. คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี. สืบค้นเมื่อ 13 มีนาคม 2562, จาก <http://www.repository.rmutt.ac.th/bitstream/handle/123456789/880/053614.pdf?sequence=1>
- กิตติศักดิ์ ศรีกลาง. (2559). การวางแผนโครงการทำแผนที่จากภาพถ่ายทางอากาศ. สืบค้นเมื่อ 10 มีนาคม 2562, จาก [www.rtsd.mi.th/Arial\\_Project\\_Planing.pdf](http://www.rtsd.mi.th/Arial_Project_Planing.pdf)
- คริส เอื้อไพบูลย์ชัย. (2558). การตัดขนาดผลล้าโดยใช้การแปลงฮัฟ Longan Grading Using Hough Transform. สืบค้นเมื่อ 10 มีนาคม 2562, จาก [http://webstaff.kmutt.ac.th/~werapon.chi/Documents/2016\\_ICSEC\\_2\\_Abstract.pdf](http://webstaff.kmutt.ac.th/~werapon.chi/Documents/2016_ICSEC_2_Abstract.pdf)
- ชุตินันต์ เกิดวิบูลย์เวช. (2554). จากตามมนุษย์สู่สมองคอมพิวเตอร์. สืบค้นเมื่อ 10 มีนาคม 2562, จาก <http://info.gotomanager.com/news/printnews.aspx?id=93754>
- ธราวุฒิ บุญเหลือ. (2556). การประยุกต์ใช้อากาศยานไร้คนขับ เพื่อจัดทำภาพถ่ายทางอากาศสำหรับการออกแบบชุมชนเมือง : กรณีศึกษาพื้นที่ มหาวิทยาลัยมหาสารคาม วิทยาเขตขามเรียง . วารสารมนุษยศาสตร์ และ สังคมศาสตร์มหาวิทยาลัยมหาสารคาม, ปีที่ 34, ฉบับที่ 5 (ก.ย.-ต.ค. 2558). หน้า 55-68.
- นฤชาติ ณ ตะกั่วทุ่ง. (2557). เช็ชชื่อจากภาพถ่าย. วิทยานิพนธ์ วศ.บ., มหาวิทยาลัยสงขลานครินทร์, ภูเก็ต
- ผศ.ดร.สุชาติ คุ้มมะณี. (2558). เชี่ยวชาญการเขียนโปรแกรมด้วยไพธอน. สืบค้นเมื่อ 30 มีนาคม 2562. จาก <https://isan.msu.ac.th/suchart/Python/ProgrammingExpertwithPython.pdf>
- มหศักดิ์ เกตุฉ่ำ. (2561) การหาขอบภาพ (Edge Detection). สืบค้นเมื่อ 10 มีนาคม 2562, จาก [www2.it.kmutnb.ac.th/teacher/FileDL/DrMahasak142256116161.pptx](http://www2.it.kmutnb.ac.th/teacher/FileDL/DrMahasak142256116161.pptx)
- วิทย์ เทียงบูรณธรรม. (2536). พจนานุกรมสมุนไพรรไทย. กรุงเทพฯ : พิมพ์ครั้งที่ 2, สำนักพิมพ์สุริยบรรณ.
- วิภารัตน์ อัมพะวัน. (2561). การศึกษาเทคนิคการประมาณสภาพกายภาพต้นยางพาราจากข้อมูลอากาศยานไร้คนขับ. วิทยานิพนธ์ วท.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก.
- วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์. (2561). มาตรฐานการสำรวจด้วยอากาศยานไร้คนขับเพื่องานวิศวกรรม.
- วิศัลย์ ประสงค์สุข. (2556). Virtual Machine หลาย OS ในเครื่องเดียว. สืบค้นเมื่อ 10 มีนาคม 2562, จาก [https://www.etda.or.th/app/webroot/files/1/files/20130118\\_Virtual\\_Machine.pdf](https://www.etda.or.th/app/webroot/files/1/files/20130118_Virtual_Machine.pdf)

- สุนิสา เรืองศรี และสุวารี ศรีอาพรรณ. (2552). **ตรวจจับวัตถุต้องสงสัยด้วยการประมวลผลภาพดิจิทัล**. สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี. สืบค้นเมื่อ 10 มีนาคม 2562, จาก <http://sutir.sut.ac.th:8080/sutir/bitstream/123456789/7320/2/Fulltext.pdf>
- หทัยทิพย์ เงินอิน. (2561). **การหาปริมาณการกักเก็บคาร์บอนของสวนสักด้วยอากาศยานไร้คนขับในพื้นที่ตำบลวังนกแอ่น อำเภอวังทอง จังหวัดพิษณุโลก**. วิทยานิพนธ์ วท.บ., มหาวิทยาลัยนเรศวร, พิษณุโลก.
- Acer. (2018). **NITRO 5**. Retrieved March 15, 2019, from <https://www.acer.com/ac/th/TH/content/series/nitro5>
- Agisoft. (2017). **Agisoft PhotoScan User Manual Standard Edition, Version 1.3**. Retrieved March 19, 2019, from [www.agisoft.com/pdf/photoscan\\_1\\_3\\_en.pdf](http://www.agisoft.com/pdf/photoscan_1_3_en.pdf)
- Dilek, k. (2018). **Automatic citrus tree extraction from UAV images and digital surface models using circular Hough transform**. Computers and Electronics in Agriculture, 150: pp.289-301
- DJI. (2018). **PHANTOM 4 SPECS**. Retrieved March 15, 2019, from [www.dji.com](http://www.dji.com)
- Emad Ali Al-helaly, (2018). **A Count of Palm Trees from Satellite Image**. Contemporary Engineering Sciences, Vol. 11, 2018, no. 74, 3663 - 3672
- Gistda Academy. (2015). **ระบบการแจกจ่ายต้นไม้ในแปลงปลูกการเกษตร**. สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน). สืบค้นเมื่อ 15 มีนาคม 2562, จาก <https://gistda.or.th/main/th/node/808>
- Greenwood, F. (2015). **How to make maps with drones**. Retrieved March 15, 2019, from [www.drones.newamerica.org](http://www.drones.newamerica.org)
- Juthawut Chantharamalee. (2016). **Digital image Processing**. Retrieved March 25, 2019, from [http://dusithost.dusit.ac.th/~juthawut\\_cha/download/Chapter01 Introduction- to-Digital-Image-Processing%20\(JC\).pdf](http://dusithost.dusit.ac.th/~juthawut_cha/download/Chapter01%20Introduction-to-Digital-Image-Processing%20(JC).pdf)
- Kotyza, J. (2018). **Detection of Directions in an Image as a Method for Circle Detection**. IFAC-PapersOnLine, 51(6): pp.496-501
- Maciel, Z. (2018). **Automatic Citrus Tree Detection from UAV Images based on Convolutional Neural Networks**. IBM Research, 146: pp. 22290-240
- NEXTSOFTWAREHOUSE. (2014). **การดึงวัตถุพื้นหน้าออกจากพื้นหลังโดยใช้ Threshold**. สืบค้นเมื่อ 15 มีนาคม 2562, จาก <https://nextsoftwares.wordpress.com/2014/05/22/การดึงวัตถุพื้นหน้าออก/>
- Tanzeel, U. Rehman (2019). **Current and future applications of statistical machine learning algorithms for agricultural machine vision systems**. Computers and Electronics in Agriculture, 156: pp.585-60



ภาคผนวก

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved





ภาคผนวก ก

รูปภาพการลงพื้นที่ภาคสนาม

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved



ลงพื้นที่ ครั้งที่ 1 (14/07/2019) สวนมะนาว จังหวัดสุโขทัย



ลงพื้นที่ ครั้งที่ 1 (14/07/2019) สวนมะนาว จังหวัดสุโขทัย





ลงพื้นที่ ครั้งที่ 2 (31/07/2019) สวนมะนาว จังหวัดสุโขทัย



ลงพื้นที่ ครั้งที่ 2 (31/07/2019) สวนมะนาว จังหวัดสุโขทัย



ลงพื้นที่ ครั้งที่ 3 (13/09/2019) สวนมะนาว จังหวัดสุโขทัย



ลงพื้นที่ ครั้งที่ 3 (13/09/2019) สวนมะนาว จังหวัดสุโขทัย





ลงพื้นที่ ครั้งที่ 4 (16/09/2019) สวนมะนาว จังหวัดพิษณุโลก



ลงพื้นที่ ครั้งที่ 4 (16/09/2019) สวนมะนาว จังหวัดพิษณุโลก

Copy



ภาคผนวก ข

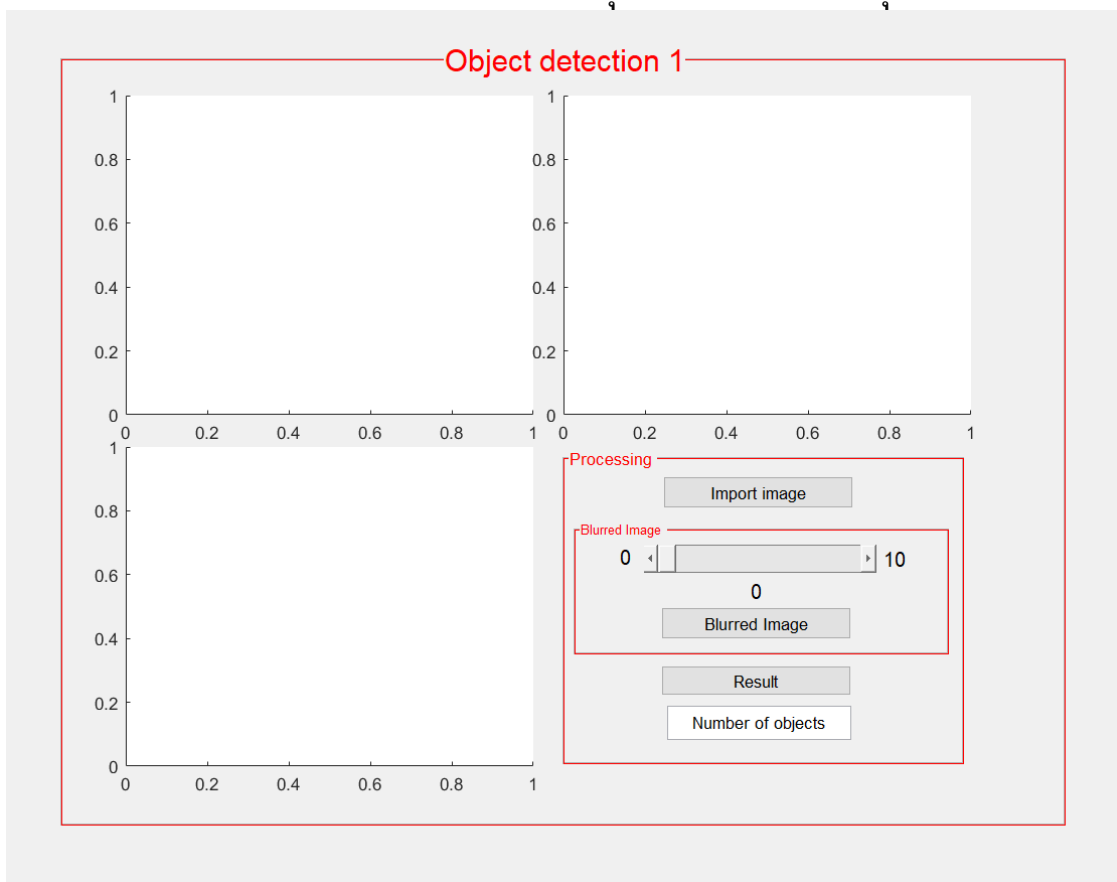
ที่พัฒนาอัลกอริทึมขึ้น

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## GUI การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ



```
function varargout = gui_technique1_2(varargin)
% GUI_TECHNIQUE1_2 MATLAB code for gui_technique1_2.fig
% GUI_TECHNIQUE1_2, by itself, creates a new GUI_TECHNIQUE1_2 or raises the
existing
% singleton*.
% ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
% H = GUI_TECHNIQUE1_2 returns the handle to a new GUI_TECHNIQUE1_2 or the
handle to
% the existing singleton*.
% All rights reserved
% GUI_TECHNIQUE1_2('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_TECHNIQUE1_2.M with the given input
arguments.
%
% GUI_TECHNIQUE1_2('Property','Value',...) creates a new GUI_TECHNIQUE1_2 or
raises the
```

```

% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before gui_technique1_2_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to gui_technique1_2_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui_technique1_2

% Last Modified by GUIDE v2.5 25-Sep-2019 01:37:35

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @gui_technique1_2_OpeningFcn, ...
    'gui_OutputFcn', @gui_technique1_2_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gui_technique1_2 is made visible.
function gui_technique1_2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.

```



```

% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui_technique1_2 (see VARARGIN)

% Choose default command line output for gui_technique1_2
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_technique1_2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_technique1_2_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Img
[baseFileName, folder] = uigetfile({'*.png'});
fullFileName = fullfile(folder, baseFileName);
Img = imread(fullFileName);
axes(handles.axes1);
imshow(Img);

% --- Executes on button press in pushbutton3.

```

```

function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Img

x = str2double(get(handles.text5,'string'));
assignin('base','x',x);

rgbImage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');

BW = im2bw(blurredImage, graythresh(blurredImage));

e=edge(BW, 'canny');

bw2 = imfill(e, 'holes');

axes(handles.axes2);
imshow (blurredImage);

L = bwlabel(bw2);
axes(handles.axes3);
imshow (bw2);

s = regionprops(L, 'Centroid');
hold on
for k = 1:numel(s)
    c = s(k).Centroid;
    text(c(1), c(2), sprintf('%d', k), ...
        'HorizontalAlignment', 'center', ...
        'VerticalAlignment', 'middle');

```

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร  
Copyright by Naresuan University

All rights reserved

```

end
hold off
set(handles.edit1,'string',k);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Img

x = str2double(get(handles.text5,'string'));
assignin('base','x',x);

rgbImage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');

BW = im2bw(blurredImage, graythresh(blurredImage));

e=edge(BW, 'canny');

bw2 = imfill(e, 'holes');
axes(handles.axes2);
imshow (blurredImage);

% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

```

```

%     get(hObject,'Min') and get(hObject,'Max') to determine range of slider

valor=get(hObject,'value');
set(handles.text5,'string',num2str(valor));

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over pushbutton1.
function pushbutton1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes during object creation, after setting all properties.
function text5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```



```

% --- Executes during object creation, after setting all properties.
function text2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

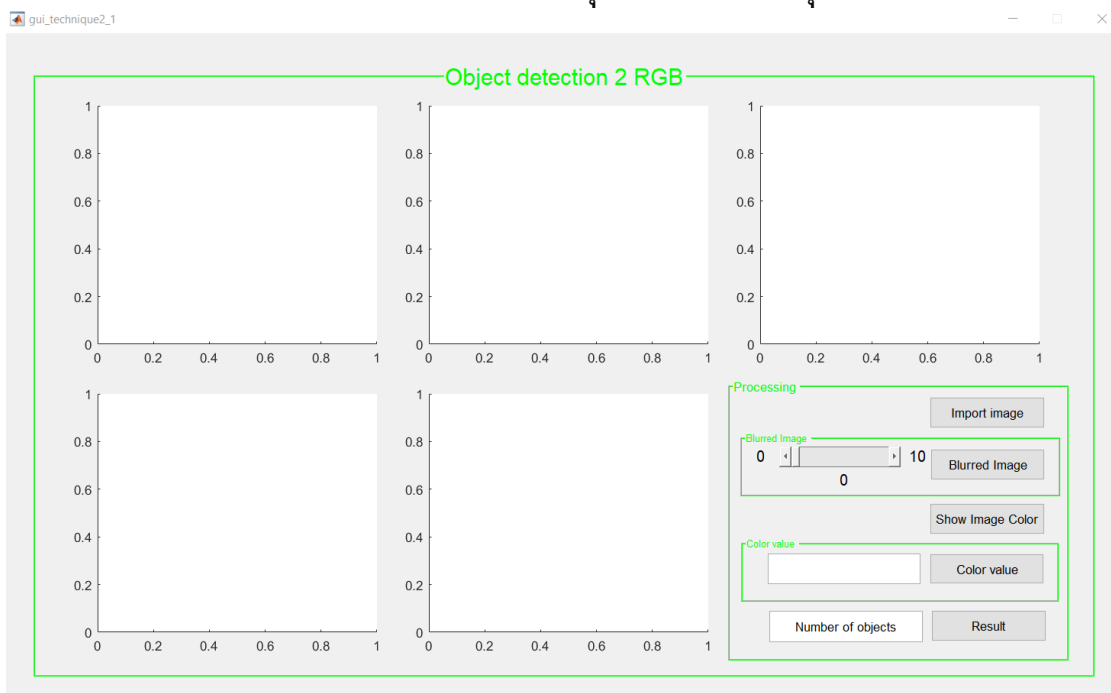
```

ลิขสิทธิ์ © มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (RGB)



```
function varargout = gui_technique2_1(varargin)
% GUI_TECHNIQUE2_1 MATLAB code for gui_technique2_1.fig
%   GUI_TECHNIQUE2_1, by itself, creates a new GUI_TECHNIQUE2_1 or raises the
existing
%   singleton*.
%
%   H = GUI_TECHNIQUE2_1 returns the handle to a new GUI_TECHNIQUE2_1 or the
handle to
%   the existing singleton*.
%   ลิขสิทธิ์ มหาวิทยาลัยนเรศวร
%   Copyright by Naresuan University
%   GUI_TECHNIQUE2_1('CALLBACK',hObject,eventData,handles,...) calls the local
function named CALLBACK in GUI_TECHNIQUE2_1.M with the given input
arguments.
%   All rights reserved
%   GUI_TECHNIQUE2_1('Property','Value',...) creates a new GUI_TECHNIQUE2_1 or
raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before gui_technique2_1_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to gui_technique2_1_OpeningFcn via varargin.
```

```
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help gui_technique2_1
```

```
% Last Modified by GUIDE v2.5 25-Sep-2019 02:45:04
```

```
% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @gui_technique2_1_OpeningFcn, ...
    'gui_OutputFcn', @gui_technique2_1_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if narginout
```

```
    [varargout{1:narginout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
```

```
    gui_mainfcn(gui_State, varargin{:});
```

```
end
```

```
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before gui_technique2_1 is made visible.
```

```
function gui_technique2_1_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% This function has no output args, see OutputFcn.
```

```
% hObject handle to figure
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% varargin command line arguments to gui_technique2_1 (see VARARGIN)
```



```

% Choose default command line output for gui_technique2_1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_technique2_1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_technique2_1_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

global Img
[baseFileName, folder] = uigetfile({'*.png'});
fullFileName = fullfile(folder, baseFileName);
Img = imread(fullFileName);
axes(handles.axes1);
imshow(Img);

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)

```



```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
global Img
```

```
x = str2double(get(handles.text4,'string'));
```

```
assignin('base','x',x);
```

```
rgbImage = im2double(Img);
```

```
windowSize = x;
```

```
avg3 = ones(windowSize) / windowSize^2;
```

```
blurredImage = imfilter(rgbImage, avg3, 'conv');
```

```
axes(handles.axes2);
```

```
imshow (blurredImage);
```

```
img=double(blurredImage);
```

```
r=blurredImage(:,:,1);
```

```
g=blurredImage(:,:,1);
```

```
b=blurredImage(:,:,1);
```

```
axes(handles.axes3);
```

```
imagesc(g),axis image;colorbar
```

```
n = str2double(get(handles.edit1,'string'));
```

```
assignin('base','n',n);
```

```
im_dist=bwdist(g > n);
```

```
imagesc(im_dist,'Parent',handles.axes4),
```

```
im_dist=bwdist(g>n);
```

```
BW = im2bw(im_dist, graythresh(im_dist));
```

```
L = bwlabel(BW);
```

```
s = regionprops(L, 'Centroid');
```



```

axes(handles.axes5);
imshow (BW);

hold on
for k = 1:numel(s)
    c = s(k).Centroid;
    text(c(1), c(2), sprintf('%d', k), ...
        'HorizontalAlignment', 'center', ...
        'VerticalAlignment', 'middle');
end
hold off

set(handles.edit3,'string',k);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);

rgblmage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgblmage, avg3, 'conv');

axes(handles.axes2);
imshow (blurredImage);

img=double(blurredImage);
r=blurredImage(:,:,1);
g=blurredImage(:,:,1);

```

```
b=blurredImage(:,:,1);
```

```
axes(handles.axes3);
```

```
imagesc(g),axis image;colorbar
```

```
function edit3_Callback(hObject, eventdata, handles)
```

```
% hObject handle to edit3 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit3 as text
```

```
% str2double(get(hObject,'String')) returns contents of edit3 as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit3_CreateFcn(hObject, eventdata, handles)
```

```
% hObject handle to edit3 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
```

```
end
```

```
% --- Executes on button press in pushbutton6.
```

```
function pushbutton6_Callback(hObject, eventdata, handles)
```

```
% hObject handle to pushbutton6 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
global lmg
```

```
x = str2double(get(handles.text4,'string'));
```

```
assignin('base','x',x);
```

```

rgbImage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');

axes(handles.axes2);
imshow (blurredImage);

img=double(blurredImage);
r=blurredImage(:,:,1);
g=blurredImage(:,:,1);
b=blurredImage(:,:,1);

axes(handles.axes3);
imagesc(g),axis image;colorbar

n = str2double(get(handles.edit1,'string'));
assignin('base','n',n);

% rgbim_dist=bwdist(g > n);
% dseim_dist=bwdist(g < n);

im_dist=bwdist(g > n);

imagesc(im_dist,'Parent',handles.axes4),
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

```



```
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
%    See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton2.
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pushbutton2 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     structure with handles and user data (see GUIDATA)
```

```
global Img
```

```
x = str2double(get(handles.text4,'string'));
```

```
assignin('base','x',x);
```

```
rgblmage = im2double(Img);
```

```
windowSize = x;
```

```
avg3 = ones(windowSize) / windowSize^2;
```

```
blurredImage = imfilter(rgblmage, avg3, 'conv');
```

```
axes(handles.axes2);
```

```
imshow (blurredImage);
```

```
% --- Executes on slider movement.
```

```
function slider1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to slider1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles     structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
```

```
%    get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

```

valor=get(hObject,'value');
set(handles.text4,'string',num2str(valor));

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2

% --- Executes during object creation, after setting all properties.
function text4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

```

% --- Executes when Tag is resized.
function Tag_SizeChangedFcn(hObject, eventdata, handles)
% hObject    handle to Tag (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.
function Tag_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Tag (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% switch(get(eventdata.NewValue,'Tag'));
%   case 'radiobutton1'
%     get(handles.radiobutton1,'string');
%   case 'radiobutton2'
%     get(handles.radiobutton2,'string');
% end

```

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## GUI การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ (DSM)



```
function varargout = gui_technique2_2(varargin)
% GUI_TECHNIQUE2_2 MATLAB code for gui_technique2_2.fig
%   GUI_TECHNIQUE2_2, by itself, creates a new GUI_TECHNIQUE2_2 or raises the
existing
%   singleton*.
%
%   H = GUI_TECHNIQUE2_2 returns the handle to a new GUI_TECHNIQUE2_2 or the
handle to
%   the existing singleton*.
%   लिखित्री महाविद्यालयनरेस्वर
%   GUI_TECHNIQUE2_2('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI_TECHNIQUE2_2.M with the given input
arguments.
%
%   All rights reserved
%   GUI_TECHNIQUE2_2('Property','Value',...) creates a new GUI_TECHNIQUE2_2 or
raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before gui_technique2_2_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to gui_technique2_2_OpeningFcn via varargin.
```



```

%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui_technique2_2

% Last Modified by GUIDE v2.5 25-Sep-2019 03:02:40

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @gui_technique2_2_OpeningFcn, ...
    'gui_OutputFcn', @gui_technique2_2_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gui_technique2_2 is made visible.
function gui_technique2_2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to gui_technique2_2 (see VARARGIN)

```

```

% Choose default command line output for gui_technique2_2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_technique2_2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_technique2_2_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

global Img
[baseFileName, folder] = uigetfile({'*.png'});
fullFileName = fullfile(folder, baseFileName);
Img = imread(fullFileName);
axes(handles.axes1);
imshow(Img);

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
global Img
```

```
x = str2double(get(handles.text4,'string'));
```

```
assignin('base','x',x);
```

```
rgbImage = im2double(Img);
```

```
windowSize = x;
```

```
avg3 = ones(windowSize) / windowSize^2;
```

```
blurredImage = imfilter(rgbImage, avg3, 'conv');
```

```
axes(handles.axes2);
```

```
imshow(blurredImage);
```

```
img=double(blurredImage);
```

```
r=blurredImage(:,:,1);
```

```
g=blurredImage(:,:,1);
```

```
b=blurredImage(:,:,1);
```

```
axes(handles.axes3);
```

```
imagesc(g),axis image;colorbar
```

```
n = str2double(get(handles.edit1,'string'));
```

```
assignin('base','n',n);
```

```
im_dist=bwdist(g < n);
```

```
imagesc(im_dist,'Parent',handles.axes4),
```

```
im_dist=bwdist(g<n);
```

```
BW = im2bw(im_dist, graythresh(im_dist));
```

```
L = bwlabel(BW);
```

```
s = regionprops(L, 'Centroid');
```



```

axes(handles.axes5);
imshow (BW);

hold on
for k = 1:numel(s)
    c = s(k).Centroid;
    text(c(1), c(2), sprintf('%d', k), ...
        'HorizontalAlignment', 'center', ...
        'VerticalAlignment', 'middle');
end
hold off

set(handles.edit3,'string',k);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);

rgblmage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgblmage, avg3, 'conv');
axes(handles.axes2);
imshow (blurredImage);

img=double(blurredImage);
r=blurredImage(:,:,1);
g=blurredImage(:,:,1);
b=blurredImage(:,:,1);

```



```
axes(handles.axes3);
imagesc(g),axis image;colorbar
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);
```

```
rgbImage = im2double(Img);
```

```

windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');

axes(handles.axes2);
imshow (blurredImage);

img=double(blurredImage);
r=blurredImage(:,:,1);
g=blurredImage(:,:,1);
b=blurredImage(:,:,1);

axes(handles.axes3);
imagesc(g,axis image;colorbar

n = str2double(get(handles.edit1,'string'));
assignin('base','n',n);

% rgbim_dist=bwdist(g > n);
% dseim_dist=bwdist(g < n);

im_dist=bwdist(g < n);

imagesc(im_dist,'Parent',handles.axes4),

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in pushbutton2.
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% hObject handle to pushbutton2 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
global Img
```

```
x = str2double(get(handles.text4,'string'));
```

```
assignin('base','x',x);
```

```
rgblmage = im2double(Img);
```

```
windowSize = x;
```

```
avg3 = ones(windowSize) / windowSize^2;
```

```
blurredImage = imfilter(rgblmage, avg3, 'conv');
```

```
axes(handles.axes2);
```

```
imshow (blurredImage);
```

```
% --- Executes on slider movement.
```

```
function slider1_Callback(hObject, eventdata, handles)
```

```
% hObject handle to slider1 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
```

```
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

```

valor=get(hObject,'value');
set(handles.text4,'string',num2str(valor));

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton1

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2
% --- Executes during object creation, after setting all properties.
function text4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when Tag is resized.
function Tag_SizeChangedFcn(hObject, eventdata, handles)
% hObject    handle to Tag (see GCBO)

```



```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.
function Tag_CreateFcn(hObject, eventdata, handles)
% hObject handle to Tag (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% switch(get(eventdata.NewValue,'Tag'));
% case 'radiobutton1'
% get(handles.radiobutton1,'string');
% case 'radiobutton2'
% get(handles.radiobutton2,'string');
% end
```

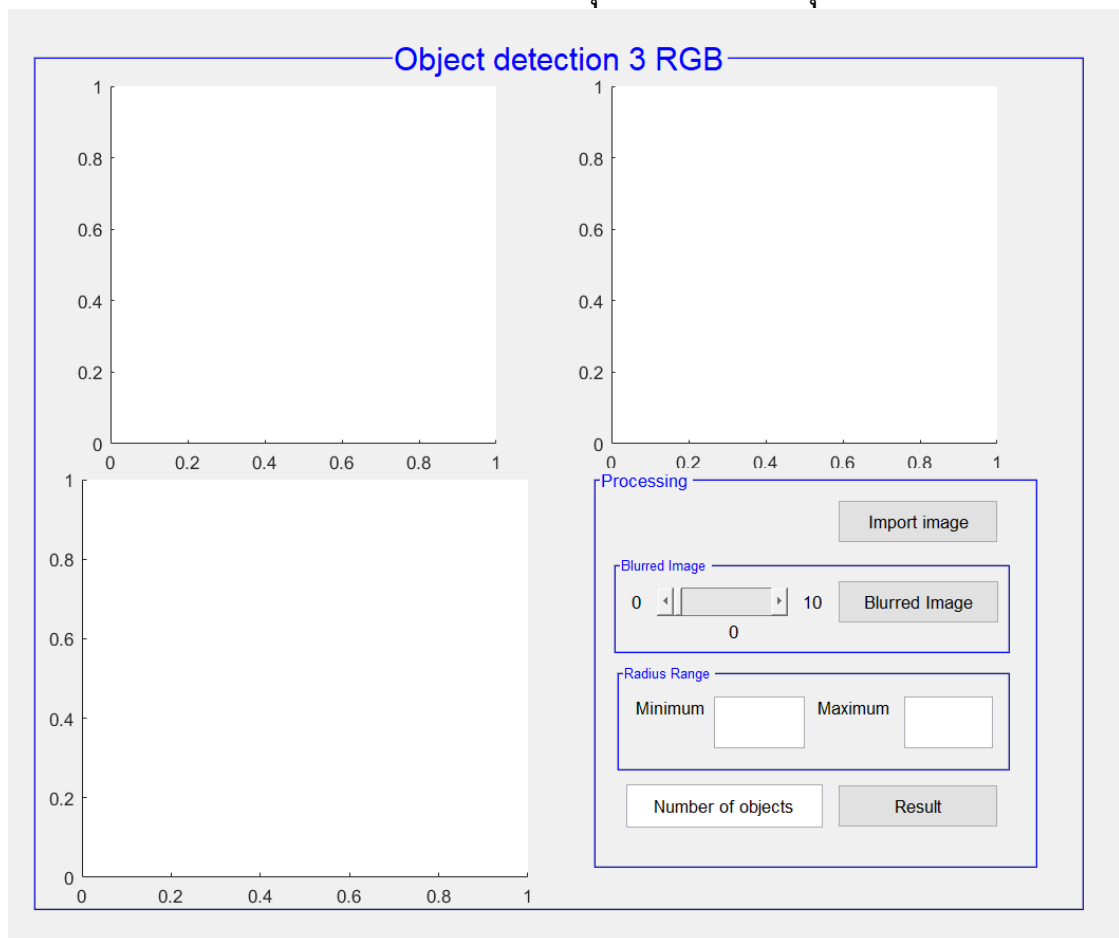


ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

### GUI การตรวจนับจำนวนวัตถุจากสีของวัตถุ (RGB)



```

function varargout = gui_technique3_1(varargin)
% GUI_TECHNIQUE3_1 MATLAB code for gui_technique3_1.fig
%   GUI_TECHNIQUE3_1, by itself, creates a new GUI_TECHNIQUE3_1 or raises the
existing
%   singleton*.
%
%   H = GUI_TECHNIQUE3_1 returns the handle to a new GUI_TECHNIQUE3_1 or the
handle to
%   the existing singleton*.
%
%   GUI_TECHNIQUE3_1('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI_TECHNIQUE3_1.M with the given input
arguments.
%

```

```

% GUI_TECHNIQUE3_1('Property','Value',...) creates a new GUI_TECHNIQUE3_1 or
raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before gui_technique3_1_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to gui_technique3_1_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui_technique3_1

% Last Modified by GUIDE v2.5 25-Sep-2019 03:45:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @gui_technique3_1_OpeningFcn, ...
    'gui_OutputFcn', @gui_technique3_1_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gui_technique3_1 is made visible.

```

```

function gui_technique3_1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui_technique3_1 (see VARARGIN)

% Choose default command line output for gui_technique3_1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_technique3_1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_technique3_1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Img

[baseFileName, folder] = uigetfile({'*.png'});
fullFileName = fullfile(folder, baseFileName);
Img = imread(fullFileName);

```



```

axes(handles.axes1);
imshow(Img);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);

rgbImage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');

axes(handles.axes4);
imshow (blurredImage);

x = str2double(get(handles.edit1,'string'));
assignin('base','x',x);
y = str2double(get(handles.edit2,'string'));
assignin('base','y',y);

z = 0.85;

[centers,radii] = imfindcircles(blurredImage,[x y],'ObjectPolarity','dark', ...
'Sensitivity',z)
h = viscircles(centers,radii);
ans = length(centers)
set(handles.edit3,'string',ans);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

```

```
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);
```

```
rgbImage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');
```

```
axes(handles.axes3);
imshow (blurredImage);
t = imdistline;
```

```
% --- Executes on slider movement.
```

```
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
```

```
%     get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

```
valor=get(hObject,'value');
set(handles.text4,'string',num2str(valor));
```

```
% --- Executes during object creation, after setting all properties.
```

```
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: slider controls usually have a light gray background.
```

```

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

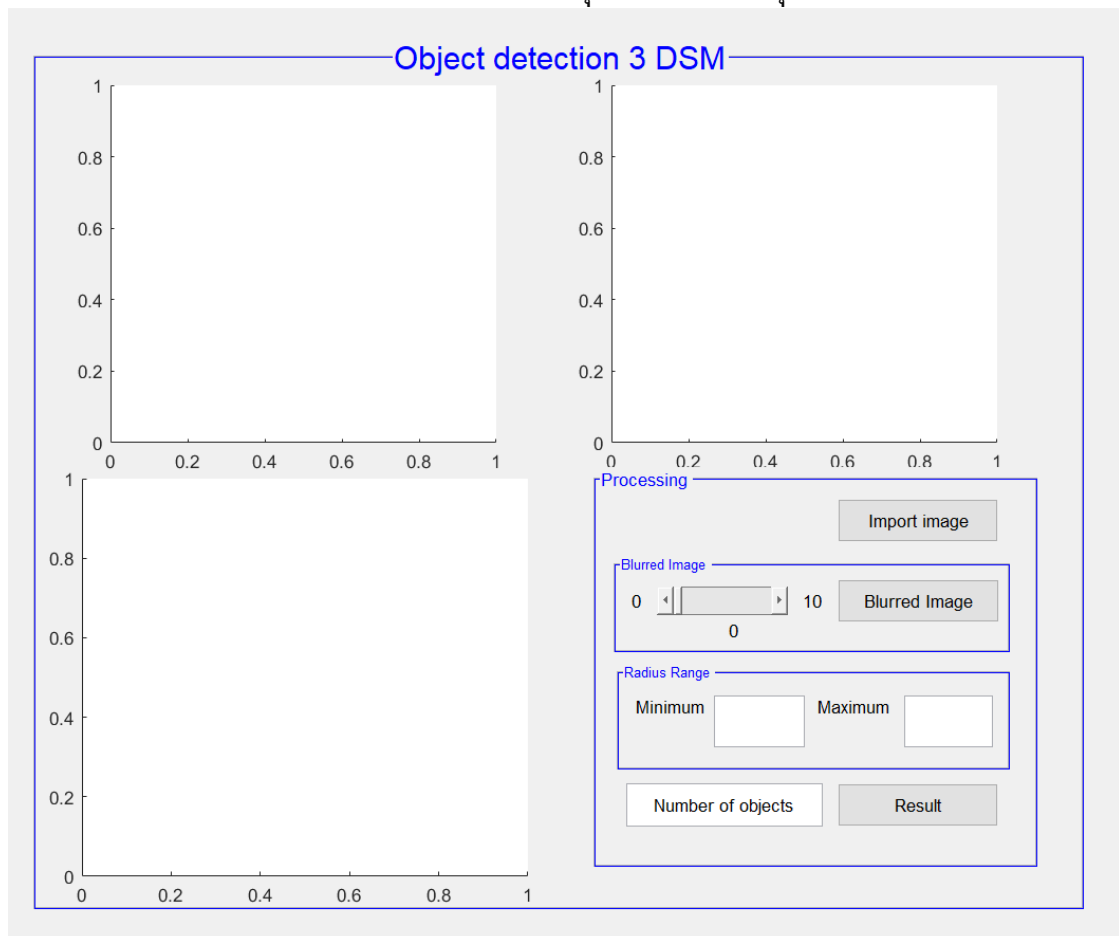
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit2 as text
% str2double(get(hObject,'String')) returns contents of edit2 as a double
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all properties.
function text4_CreateFcn(hObject, eventdata, handles)
% hObject handle to text4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```



## GUI การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ (DSM)



```

function varargout = gui_technique3_2(varargin)
% GUI_TECHNIQUE3_2 MATLAB code for gui_technique3_2.fig
%   GUI_TECHNIQUE3_2, by itself, creates a new GUI_TECHNIQUE3_2 or raises the
existing
%   singleton*.
%
%   H = GUI_TECHNIQUE3_2 returns the handle to a new GUI_TECHNIQUE3_2 or the
handle to
%   the existing singleton*.
%
%   GUI_TECHNIQUE3_2('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI_TECHNIQUE3_2.M with the given input
arguments.
%

```

```

% GUI_TECHNIQUE3_2('Property','Value',...) creates a new GUI_TECHNIQUE3_2 or
raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before gui_technique3_2_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to gui_technique3_2_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui_technique3_2

% Last Modified by GUIDE v2.5 25-Sep-2019 04:06:47

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @gui_technique3_2_OpeningFcn, ...
    'gui_OutputFcn', @gui_technique3_2_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gui_technique3_2 is made visible.

```

```

function gui_technique3_2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui_technique3_2 (see VARARGIN)

% Choose default command line output for gui_technique3_2
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_technique3_2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_technique3_2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global Img
[baseFileName, folder] = uigetfile({'*.png'});
fullFileName = fullfile(folder, baseFileName);
Img = imread(fullFileName);
axes(handles.axes1);

```

```

imshow(Img);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);

rgblmage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgblmage, avg3, 'conv');

it=graythresh(blurredImage);
bw=imbinarize(blurredImage,it);

x = str2double(get(handles.edit1,'string'));
assignin('base','x',x);
y = str2double(get(handles.edit2,'string'));
assignin('base','y',y);

z = 0.85;

axes(handles.axes4);
imshow (bw);

[centers,radii] = imfindcircles(bw,[x y],'ObjectPolarity','bright', ...
'Sensitivity',z)

h = viscircles(centers,radii);
ans = length(centers)
set(handles.edit3,'string',ans);

```



```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
global Img
x = str2double(get(handles.text4,'string'));
assignin('base','x',x);
```

```
rgbImage = im2double(Img);
windowSize = x;
avg3 = ones(windowSize) / windowSize^2;
blurredImage = imfilter(rgbImage, avg3, 'conv');
```

```
axes(handles.axes3);
imshow (blurredImage);
```

```
it=graythresh(blurredImage);
bw=imbinarize(blurredImage,it);
```

```
t = imdistline;
```

```
% --- Executes on slider movement.
```

```
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%       get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

```
valor=get(hObject,'value');
set(handles.text4,'string',num2str(valor));
```

```
% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```
set(hObject,'BackgroundColor','white');  
end  
  
% --- Executes during object creation, after setting all properties.  
function text4_CreateFcn(hObject, eventdata, handles)  
% hObject    handle to text4 (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    empty - handles not created until after all CreateFcns called
```



ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved





ภาคผนวก ค

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่าย  
จากอากาศยานไร้คนขับ

Development Algorithm for Automatic Extraction of fruit trees  
from UAV imagery

อุไรรัตน์ อุ้นเมือง<sup>1\*</sup> และนัฐพล มหาวิค<sup>1</sup>

Urairat Unmueang<sup>1\*</sup> and Nattapon Mahavik<sup>1</sup>

<sup>1</sup>ภาควิชาทรัพยากรธรรมชาติและสิ่งแวดล้อม คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและ  
สิ่งแวดล้อม มหาวิทยาลัยนเรศวร

\* Corresponding author. Email: urairatu59@email.nu.ac.th

บทคัดย่อ

จำนวนของต้นไม้ที่อยู่ในแปลงเกษตรนั้นเป็นข้อมูลที่สำคัญในการจัดการด้านเกษตรกรรม เนื่องจากจำนวนต้นไม้สามารถนำมาประมาณผลผลิตต่อพื้นที่เพาะปลูกได้ เพื่อเป็นข้อมูลช่วยตัดสินใจในการบริหารจัดการพื้นที่การเกษตรให้มีประสิทธิภาพมากยิ่งขึ้น ในการศึกษาครั้งนี้มุ่งที่จะพัฒนาอัลกอริทึมในโปรแกรม MATLAB เพื่อการตรวจนับจำนวนต้นไม้ผลแบบอัตโนมัติจากข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ คณะผู้วิจัยได้พัฒนา 3 เทคนิค คือ 1. การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ 2. การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ และ 3. การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ จากนั้นทำการตรวจสอบความถูกต้องของเทคนิค ในแต่ละเทคนิคกับข้อมูลที่ได้นับจริงจากการแปลภาพถ่ายด้วยสายตา ซึ่งได้ศึกษาพื้นที่แปลงเพาะปลูกสวนมะนาวใน อำเภอคีรีมาศ จังหวัดสุโขทัย โดยการศึกษาและพัฒนาอัลกอริทึมเบื้องต้นนั้นได้ใช้ข้อมูลภาพ Google map ของสวนไม้ผล 2 ชนิดคือ ส้มและทุเรียน จำนวนสวนละ 30 ภาพ จากการตรวจสอบความถูกต้องพบว่าร้อยละความถูกต้องของเทคนิคที่ 1, 2 และ 3 มีค่าเท่ากับ 75%, 76% และ 88% ตามลำดับ และยังได้มีการศึกษาและพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ ซึ่งได้นำข้อมูลระดับความสูงเชิงเลข (Digital Surface Model หรือ DSM) เข้ามาใช้ในการพัฒนาอัลกอริทึมพบว่าร้อยละความถูกต้องของเทคนิคที่ 1, 2 และ 3 มีค่าเท่ากับ 74%, 79% และ 92% ตามลำดับ และยังพบว่าเมื่อนำข้อมูล DSM มาใช้ในการศึกษาส่งผลให้มีค่าความถูกต้องของการนับจำนวนต้นไม้ที่ได้จากอัลกอริทึมจะมีค่าสูงขึ้น

คำสำคัญ: อากาศยานไร้คนขับ, การสกัดขอบเขตต้นไม้, การประมวลผลภาพดิจิทัล

Copyright by Naresuan University

All rights reserved

### Abstract

Tree counts and sizes are important information to apply for crop yield estimation and agricultural planning. Therefore, implementation for automatic extraction of trees numbers from remotely sensed data is a challenging task. In this study, a novel approach and algorithm developed in MATLAB is proposed for the automatic extraction of citrus trees using unmanned aerial vehicle (UAV). Initially, preliminary testing by using Google image data have been done using of 2 fruit orchards, orange and durian with 30 images per orchard the research team 3 methods developed: 1. To count objects from its boundary 2. To examine the number of objects using the color its 3. To count the number of objects using its radius. To validate the performance of the developed extraction techniques, the ground truth images were interpreted. The accuracy of techniques 1, 2 and 3 has shown the absolute mean values of 75%, 76% and 88%, respectively. Improved accuracy results were found when the improved algorithm was included DSM (Digital Surface Model) generated from UAV data. It was found that the accuracy of techniques 1, 2 and 3 are 74%, 79% and 92%, respectively. It was found that the accuracy of counting the fruit tree obtained from the developed algorithm would be usually high incorporated with DSM data.

Keywords: Unmanned Aerial Vehicle, Tree extraction, digital image processing

ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## ที่มาและความสำคัญ

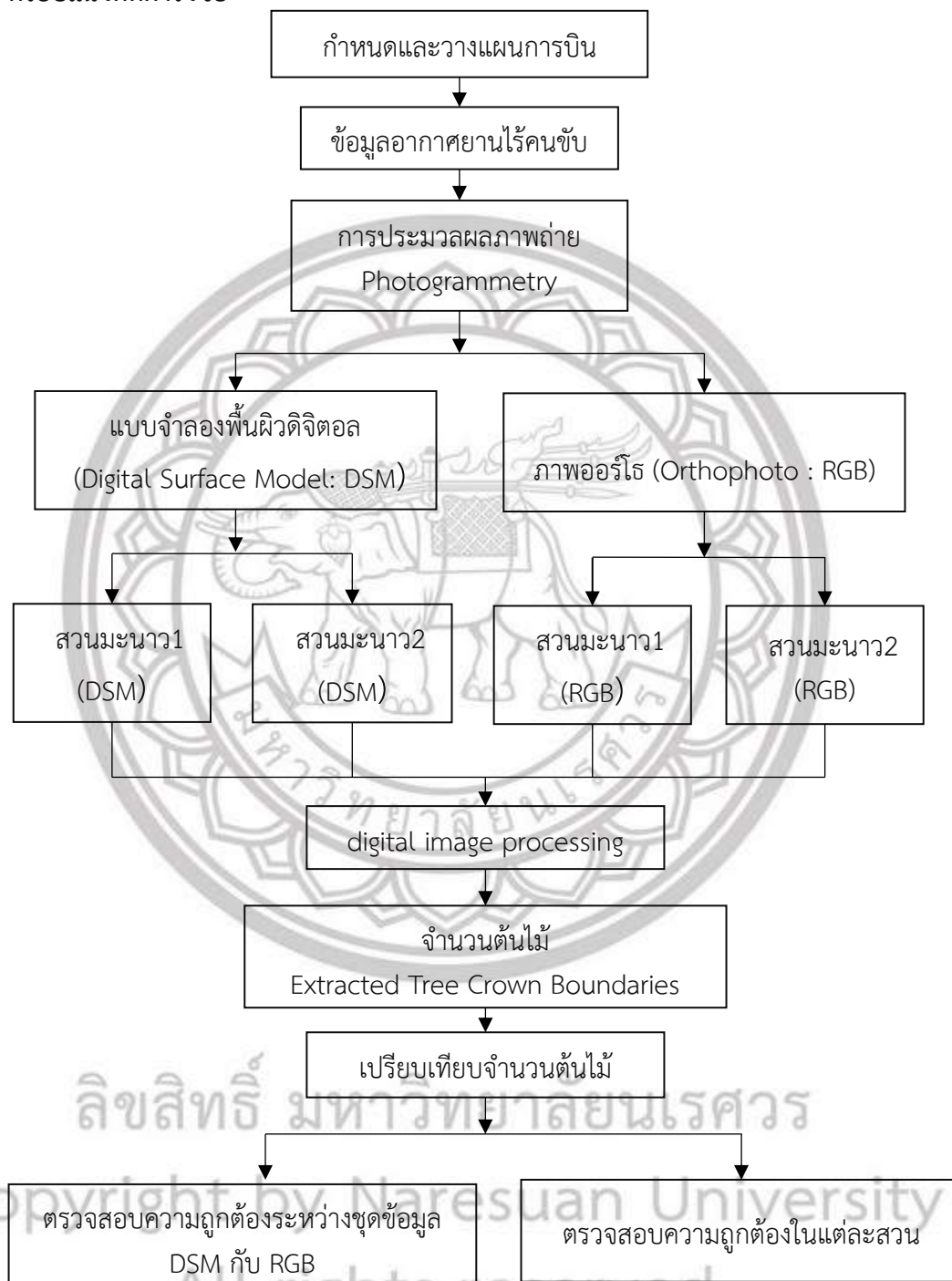
อากาศยานไร้คนขับ (Unmanned Aerial Vehicle; UAV) เป็นเทคโนโลยีสมัยใหม่ที่กำลังได้รับความนิยมนำมาใช้งานด้านสำรวจ โดยสามารถบังคับตัวอากาศยานด้วยสัญญาณวิทยุ ซึ่งมีระบบควบคุมและการสั่งการบินทั้งแบบอัตโนมัติและแบบกึ่งอัตโนมัติ สามารถควบคุมระยะไกลแทนนักบินจริง ในอดีตการใช้ประโยชน์อยู่ในแวดวงที่จำกัด มีการใช้ประโยชน์ในทางการทหารเท่านั้นเพราะมีค่าใช้จ่ายสูง แต่ในปัจจุบันเทคโนโลยีอากาศยานไร้คนขับได้มีการพัฒนาให้มีประสิทธิภาพมากขึ้นและมีราคาที่ถูกลงด้วย ทำให้มีการประยุกต์ใช้อากาศยานไร้คนขับในต่างงานอย่างหลากหลาย (ธราวุฒิ บุญเหลือ, 2556)[1] เช่น การวางแผนเพื่อพัฒนาเมืองและสภาพแวดล้อม การสำรวจทรัพยากรธรรมชาติ การสำรวจเพื่อวางแผนแก้ไขปัญหาอุตสาหกรรมและการเกษตรกรรม เป็นต้น นอกจากนี้ภาพถ่ายจากอากาศยานไร้คนขับนั้นสามารถสร้างผลิตภัณฑ์แบบจำลองความสูงโดยหลักการโฟโตแกรมเมตรี ซึ่งสามารถนำไปประยุกต์ใช้ในการประมาณความค่าสูงของวัตถุ รวมถึงการนำมาเป็นปัจจัยที่ประกอบการตรวจนับจำนวนของวัตถุ ต้นไม้ อาคาร สิ่งปลูกสร้างได้อีกด้วย

จำนวนของต้นไม้ที่อยู่ในแปลงเพาะปลูกเป็นข้อมูลสำคัญที่ใช้ในการจัดการด้านเกษตรกรรมเพื่อใช้เป็นข้อมูลพื้นฐานในการทราบถึงจำนวนผลผลิตต่อพื้นที่ ในปัจจุบันมีเทคโนโลยีที่แพร่หลายในนำมาใช้ตรวจนับจำนวนต้นไม้ที่โดยการใช้อุปกรณ์ที่มาจากแหล่งอันได้แก่ ภาพดาวเทียม ภาพถ่ายทางอากาศ ภาพจากอากาศยานไร้คนขับ เนื่องจากสามารถมองเห็นภาพรวมของพื้นที่ขนาดใหญ่ได้โดยไม่ต้องเข้าไปสำรวจในพื้นที่จริง (Gistda Academy, 2558)[4] อีกทั้งยังช่วยประหยัดเวลาในการการนับด้วยแรงงาน ทำให้เสียเวลาและงบประมาณอย่างมาก โดยเฉพาะพืชสวนที่มีการปลูกพื้นจำนวนมาก เช่น สวนมะนาว สวนปาล์ม สวนทุเรียน เป็นต้น นอกจากนี้การเข้าสำรวจในบางพื้นที่มักจะพบว่ามีต้นไม้อื่นขึ้นตามแนวทางเดินทำให้ยากต่อการเข้าถึง อย่างไรก็ตามปัจจุบันยังใช้การตรวจนับด้วยแรงงานคนเพราะยังไม่มีเทคโนโลยีที่เหมาะสมและถูกต้องในการตรวจนับโดยอัตโนมัติ จึงทำให้การตรวจนับจำนวนต้นไม้ มีความยุ่งยากและใช้เวลาค่อนข้างมาก ทำให้เทคโนโลยีการสำรวจด้วยภาพจากอากาศยานไร้คนขับเป็นทางออกหนึ่ง โดยหากมีการนำเทคโนโลยีการประมวลผลภาพมาพัฒนาเพื่อสร้างอัลกอริทึมในการตรวจนับแบบอัตโนมัติบนข้อมูลภาพถ่ายทางอากาศจะทำให้ประหยัดต้นทุนในการประมาณผลผลิตได้อย่างมหาศาล

การศึกษานี้จึงมุ่งศึกษาและพัฒนาอัลกอริทึมในการตรวจนับจำนวนไม้ผลแบบอัตโนมัติโดยการประยุกต์ใช้ข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ ด้วยการบูรณาการกระบวนการโฟโตแกรมเมตรี และเทคโนโลยีการประมวลผลภาพเชิงเลข นอกจากนี้ยังได้พัฒนา Graphic User Interface (GUI) เพื่ออำนวยความสะดวกแก่ผู้ใช้งาน (Juthawut Chantharamalee, 2016)[5]



กรอบแนวคิดการวิจัย



ภาพ 1 กรอบแนวคิดการวิจัย

คณะผู้วิจัยได้ออกแบบแนวคิดการวิจัยเพื่อใช้ในการพัฒนาอัลกอริทึมดังกล่าว 1 โดยกรอบแนวคิดนี้ได้มาจากการทดลองใช้ภาพที่ได้จาก Google map ก่อน ทำให้ทราบถึงแนวทางการพัฒนาโค้ดใน MATLAB ที่มุ่งพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนต้นไม้ผลจากภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับที่ได้นำอัลกอริทึมที่พัฒนาขึ้นนี้นำมาทดสอบกับสวนมะนาวด้วยการใช้ข้อมูลทั้งจาก แบบจำลองพื้นผิวเชิงเลขและภาพออร์โธ จากนั้นทำการตรวจวัดความถูกต้องด้วยข้อมูลจากการแปลด้วยสายตาและข้อมูลจากการสำรวจจริงในภาคสนาม

## อุปกรณ์และวิธีการ

### 1. การเตรียมข้อมูล

#### 1.1 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย

#### ตาราง 3 ข้อมูลและเครื่องมือที่ใช้ในการวิจัย

ลำดับ	ข้อมูล / อุปกรณ์	ประเภท	แหล่งที่มา / คุณสมบัติ
1	ภาพถ่ายทางอากาศ	Raster	- DJI Phantom 4 Advanced
2	DJI Phantom 4 Advanced	Multicopter	- น้ำหนัก (รวมแบตเตอรี่และใบพัด) 1380 กรัม - ความละเอียดภาพนิ่ง 20 Mega-Pixel - แบตเตอรี่ Lipo ความจุ 5350 mAH - ระบุตำแหน่งดาวเทียม GPS / GLONASS
3	ขอบเขตข้อมูลจังหวัด อำเภอดำบล	Vector	- กรมการปกครอง
4	คอมพิวเตอร์โน้ตบุ๊ก Acer Nitro 5 AN515-51	Hardware	- Intel Core i7-7700HQ - NVIDIA GeForce GTX 1050Ti (4GB GDDR5) - 8 GB DDR4 - 1 TB 5400 RPM + 128 GB SSD
5	Agisoft Photoscan	Software	- โปรแกรมสร้างแผนที่จาก ข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ
6	Pix4d Mapper	Software	- โปรแกรมสร้างแผนที่จาก ข้อมูลภาพถ่ายจากอากาศยานไร้คนขับรุ่นทดลองใช้
7	ArcGIS	Software	- โปรแกรมสารสนเทศทางภูมิศาสตร์
8	MATLAB	Software	- โปรแกรมคำนวณทางคณิตศาสตร์

#### 1.2 การเก็บรวบรวมข้อมูล

คณะผู้วิจัยได้ทำการเก็บรวบรวมข้อมูลที่ใช้ในการศึกษาเพื่อเปรียบเทียบผลของการพัฒนาอัลกอริทึมกับข้อมูลจาก 2 แหล่งได้แก่ ข้อมูลภาพถ่ายทางอากาศจาก Google map กับ ข้อมูลภาพถ่ายทางอากาศที่ได้จากอากาศยานไร้คนขับ โดยใช้วิธีการแปลด้วยสายตาเพื่อจำแนกจำนวนต้นไม้เพื่อนำมาตรวจสอบความถูกต้องของอัลกอริทึมที่คณะผู้วิจัยได้พัฒนาขึ้น และรวบรวมข้อมูลที่ได้จากการลงภาคสนามตรวจสอบ

### 1.2.1 ข้อมูลภาพถ่ายทางอากาศจาก Google map

คณะผู้วิจัยได้ทำการเลือกพื้นที่ตัวอย่างจาก Google map โดยเลือกพืชสวนที่มีลักษณะทรงพุ่มคล้ายกับทรงพุ่มของต้นมะนาว เพื่อนำภาพไปทดลองกับอัลกอริทึมเบื้องต้นที่ได้พัฒนาใน MATLAB ก่อนที่จะทดลองกับข้อมูลจริง ซึ่งผู้วิจัยได้เลือกพืชมา 2 ชนิด คือ ส้ม และ ทุเรียน จำนวนชนิดละ 30 สวน เพื่อทดสอบความสามารถในการนับจำนวนต้นไม้ของอัลกอริทึมที่ได้พัฒนาขึ้นมา

### 1.2.2 ข้อมูลภาพถ่ายทางอากาศจากอากาศยานไร้คนขับ

คณะผู้วิจัยได้วางแผนแนวนบินในการถ่ายภาพแปลงปลูกต้นมะนาวที่ใช้ในการศึกษาทั้ง 2 แห่ง คือ บริเวณ ต.สามพวง อ.ศรีมหาศ จ.สุโขทัย และ ต.ท่าโพธิ์ อ.เมืองพิษณุโลก จ.พิษณุโลก โดยใช้แอปพลิเคชัน DJI GS PRO เพื่อทำการวางแผนแนวนบิน ในพื้นที่สวนมะนาว จ.สุโขทัย กำหนดระดับความสูงบินที่ 50 เมตร กำหนดให้มีส่วนซ้อนทับในแนวนบินเดียวกันและระวางแนวนบิน คิดเป็นร้อยละ 90 และ 60 ตามลำดับ มีจำนวน 3 แนวนบิน ครอบคลุมพื้นที่ประมาณ 4.5 ไร่ ความยาวเส้นทางการบินรวมทั้งหมด 396 เมตร ได้จำนวนภาพทั้งหมด 46 ภาพ และในซอฟต์แวร์ DJI GS PRO พบว่าพื้นที่ที่บินถ่ายไม่พื้นที่สวนหรือพื้นที่ที่พืชใด ๆ เนื่องจากข้อมูลภาพที่ถูกนำมาขึ้นเป็นภาพที่เก่า และคณะผู้วิจัยทำการบินถ่ายภาพในพื้นที่สวนมะนาว จ.พิษณุโลก บินในระดับความสูงประมาณ 50 เมตร กำหนดให้มีส่วนซ้อนทับในแนวนบินและระหว่างแนวนบินที่ร้อยละ 80 และ 60 ตามลำดับ จำนวน 5 แนวนบิน ครอบคลุมพื้นที่ประมาณ 11.6 ไร่ ความยาวเส้นทางการบิน 939 เมตร ได้ภาพทั้งหมดจำนวน 60 ภาพ

### 1.3 การประมวลผล

การประมวลผลภาพถ่ายที่ได้จากอากาศยานไร้คนขับโดยเวอร์ชันทดลองใช้ของโปรแกรม Pix4Dmapper เลือกค่าความละเอียดของภาพเป็น Standard กำหนดค่าพิกัดฉากด้วยเส้นโครงแผนที่แบบ UTM Zone 47N หมุดหลักฐานแบบ WGS 84 กำหนดประเภทของการประมวลผลแบบ 3D Maps ซึ่งผลิตภัณฑ์ที่ได้คือ ภาพออร์โธซีและแบบจำลองพื้นผิวเชิงเลข (Digital Surface Model หรือ DSM)

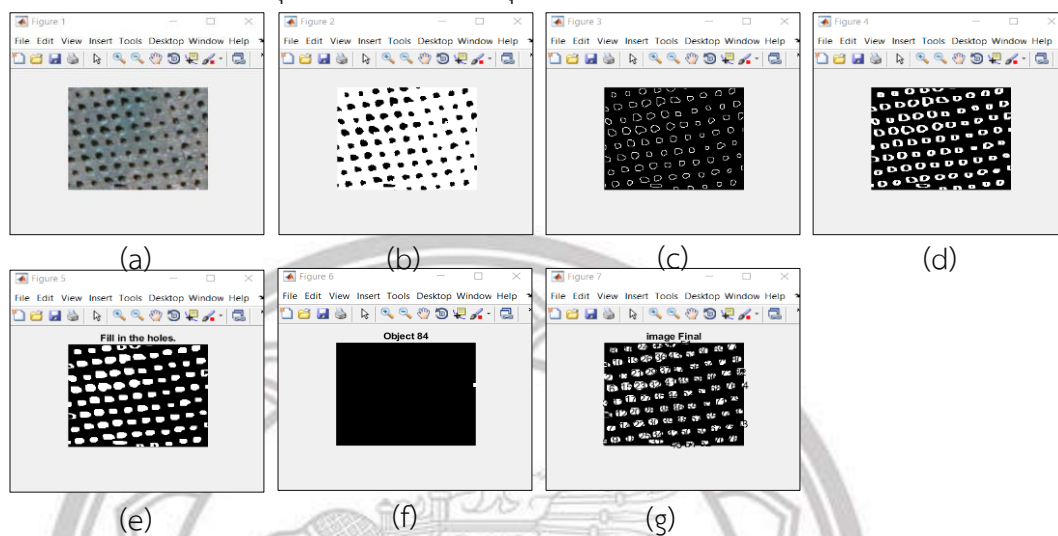
## 2. การวิเคราะห์ข้อมูล

### 2.1 การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายดาวเทียม google map

2.1.1 สุ่มพื้นที่เพื่อบันทึกภาพหน้าจอพื้นที่ตัวอย่างจากภาพถ่ายดาวเทียม google map ทั้งส้ม และ ทุเรียน จำนวนชนิดละ 30 สวน โดยใช้โปรแกรม oCam ในการบันทึกภาพหน้าจอ และกำหนดขนาดรูปภาพเท่ากับ 500x400 คือ กว้าง 500 Pixels สูง 400 Pixels

2.1.2 พัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนต้นไม้ผลแบบอัตโนมัติ การพัฒนาเทคนิคหรืออัลกอริทึมตรวจนับจำนวนไม้ผลอัตโนมัติในขั้นตอนนี้เป็นการทดลองเบื้องต้นกับข้อมูลภาพถ่ายดาวเทียม จาก Google map เพื่อใช้ทดลองกับอัลกอริทึมที่พัฒนาในโปรแกรม MATLAB ซึ่งคณะผู้วิจัยได้ทดลองพัฒนาเทคนิคมาทั้งหมด 3 เทคนิค ดังต่อไปนี้

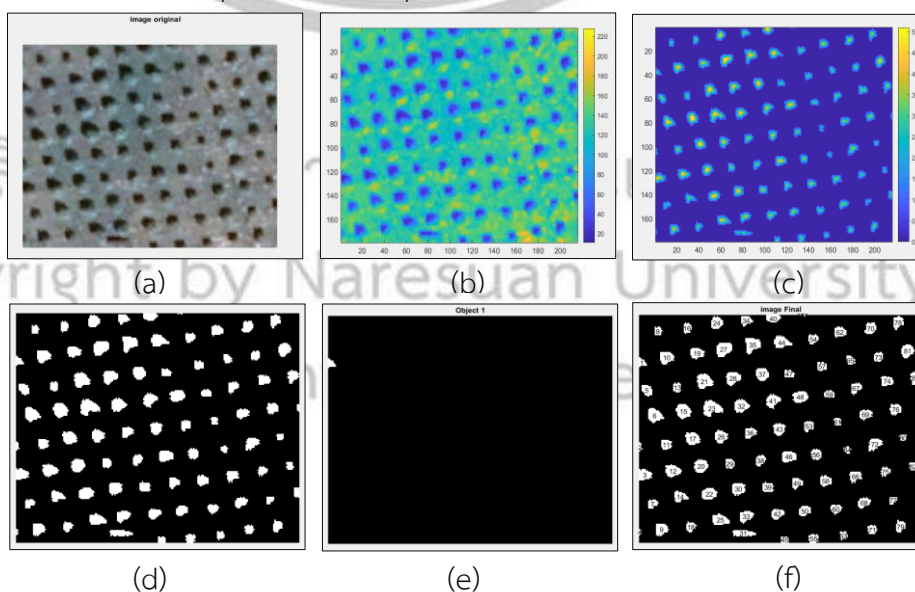
(1) การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ มีขั้นตอนการทำงานดังนี้



ภาพ 2 การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ

ขั้นตอนแรกนำภาพสีเข้าสู่ MATLAB โดยใช้ฟังก์ชัน `imread` ในการเรียกภาพ ภาพ 2(a) จากนั้นทำการแปลงภาพจากภาพสีเป็นภาพขาวดำด้วยฟังก์ชัน `im2bw` ภาพ 2(b) จากนั้นทำการหาขอบของวัตถุโดยใช้ฟังก์ชัน `canny` ภาพ 2(c) (มหศักดิ์ เกตุฉ่ำ, 2561)[2] เมื่อได้ขอบของวัตถุแล้ว จะต้องทำการเพิ่มความหนาของเส้นขอบวัตถุที่ได้เนื่องจากขอบวัตถุบางวัตถุมีระยะห่างเกินไป ทำให้ไม่สามารถตรวจนับได้ โดยใช้ฟังก์ชัน `imdilate` ภาพ 2(d) ขั้นตอนต่อไปทำการเติมช่องว่างในวัตถุโดยใช้ฟังก์ชัน `holes` ภาพ 2(e) และในขั้นตอนต่อมาทำการทดสอบการนับวัตถุเพื่อตรวจสอบวัตถุว่าสามารถนับได้หรือไม่ ภาพ 2(f) และในขั้นตอนสุดท้ายทำการวนลูปเพื่อนับวัตถุว่ามีจำนวนเท่าไร ภาพ 2(g)

(2) การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ มีขั้นตอนการทำงานดังนี้

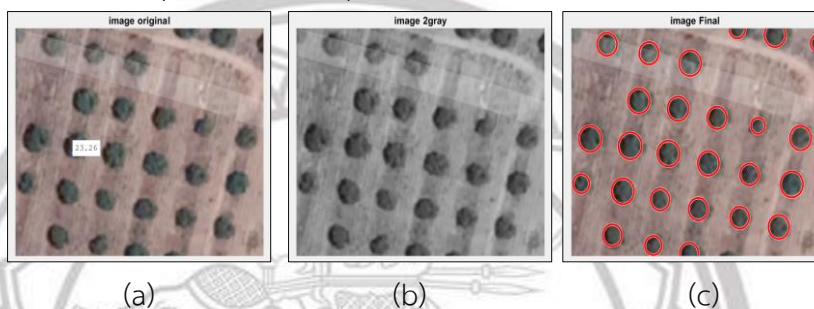


ภาพ 3 การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ



ขั้นตอนแรกนำภาพสีเข้าสู่ MATLAB โดยใช้ฟังก์ชัน `imread` ในการเรียกภาพ ภาพ 3(a) ต่อมาแสดงค่าสีของภาพ โดยใช้ฟังก์ชัน `colorbar` ภาพ 3(b) จากนั้นกำหนดค่าสีตามวัตถุที่ต้องการนับ ภาพ 3(c) และเมื่อได้ค่าสีของวัตถุแล้วแปลงภาพสีเป็นภาพขาวดำ โดยใช้ฟังก์ชัน `im2bw` ภาพ 3(d) และในขั้นตอนต่อมาทำการทดสอบการนับวัตถุเพื่อตรวจสอบวัตถุว่าสามารถนับได้หรือไม่ ภาพ 3(e) และในขั้นตอนสุดท้ายทำการวนลูปนับวัตถุเพื่อตรวจนับวัตถุว่ามีจำนวนเท่าไร ภาพ 3(f)

(3) การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ มีขั้นตอนการทำงานดังนี้



ภาพ 4 การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ

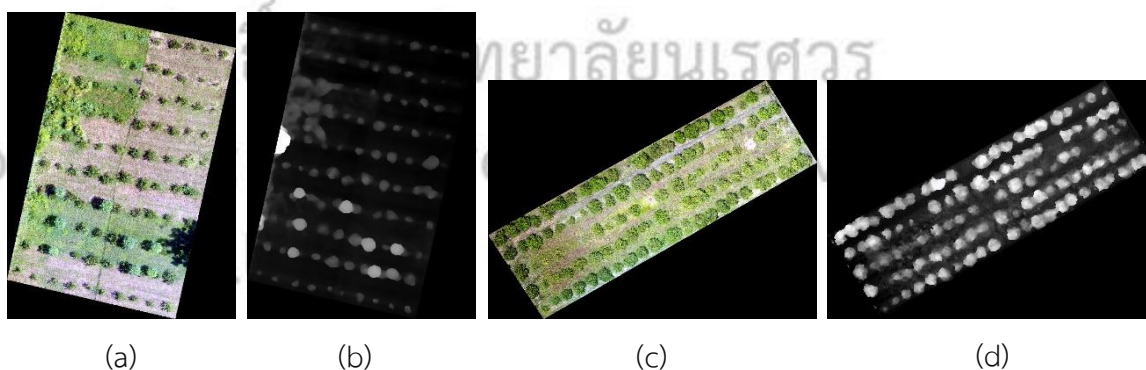
ขั้นตอนแรกนำเข้าภาพสี โดยใช้ฟังก์ชัน `imread` ภาพ 4(a) ต่อมาแปลงภาพสีเป็นภาพสีเทา โดยใช้ `rgb2gray` เนื่องจากภาพสีเทาจะทำให้สามารถระบุขอบเขตของวัตถุได้ชัดเจน ภาพ 4(b) และในขั้นตอนสุดท้ายกำหนดรัศมีต่ำสุด-สูงสุดของวัตถุเพื่อทำการนับวัตถุจากรัศมี ภาพ 4(c)

## 2.2 การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

### 2.2.1 การแปลงนามสกุลไฟล์ภาพจาก TFF เป็น PNG

เนื่องจากผลลัพธ์ของกระบวนการโฟโตแกรมเมตรีจะได้ภาพออร์โธ และ แบบจำลองพื้นผิวดิจิทัล โดยจะมีนามสกุลไฟล์ภาพเป็น TFF และไฟล์ภาพทั้งสองมีขนาดใหญ่มาก ซึ่งทำให้ไม่สามารถนำภาพเข้าสู่ โปรแกรม MATLAB ได้ จึงได้ทำการแปลงนามสกุลไฟล์ภาพจาก TFF ไปเป็น PNG และได้ลดขนาดภาพลงในโปรแกรม ArcMap

### 2.2.2 การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติ



ภาพ 5 ภาพถ่ายทางอากาศจากอากาศยานไร้คนขับทั้ง 4 ชั้นข้อมูลเพื่อเป็นปัจจัยนำเข้าในการประมวลผล

ภาพ 5 เป็นผลของการพัฒนาอัลกอริทึมในตรวจนับจำนวนไม้ผลแบบอัตโนมัติ ในขั้นตอนนี้เป็นการทดลองกับข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ โดยได้มีการดัดแปลงพัฒนาอัลกอริทึมเพิ่มเติมเพื่อให้ได้ผลลัพธ์ที่ดีขึ้น ซึ่งการพัฒนาอัลกอริทึมที่นั่นพบว่าการใช้งานตัวกรองหรือฟิลเตอร์กับภาพด้วยฟังก์ชัน imfilter นั้น ส่งผลทำให้การตรวจวัดจำนวนต้นไม้ดีขึ้น ได้ทดลองกำหนดขนาดหน้าต่างของตัวกรองหรือ window size การทดลองได้แบ่งข้อมูลเพื่อนำไปวิเคราะห์ 4 ชั้นข้อมูล คือ ภาพ 5(a) ภาพออร์โธสวันมะนาว จ.สุโขทัย ภาพ 5(b) แบบจำลองพื้นผิวดิจิตอล สวันมะนาว จ.สุโขทัย ภาพ 5(c) ภาพออร์โธสวันมะนาว จ.พิษณุโลก และ ภาพ 5(d) แบบจำลองพื้นผิวเชิงเลข สวันมะนาว จ.พิษณุโลก เพื่อให้ทราบถึงความถูกต้องของข้อมูลทั้ง 4 ชั้นข้อมูล

### 2.2.3 พัฒนาหน้าต่างแสดงผล GUI MATLAB

อัลกอริทึมที่ได้พัฒนาอัลกอริทึมใน MATLAB นั้นมีความซับซ้อนซึ่งไม่เหมาะต่อการใช้งานของกลุ่มบุคคลทั่วไป ทำให้คณะผู้วิจัยได้พัฒนา GUI (Graphical User Interface) ขึ้นมาดังภาพ 6 ซึ่งจะประกอบไปด้วยหน้าต่างสำหรับแสดงภาพถ่ายทางอากาศทั้ง 4 ส่วนที่ได้จากการประมวลผลในแต่ละขั้นตอนที่ได้พัฒนาขึ้นมา โดยมีในส่วนของ Processing ที่ผู้ใช้งานสามารถที่จะกำหนดค่าพารามิเตอร์เพื่อทดสอบค่าความอ่อนไหวหรือ sensitivity เพื่อหาค่าพารามิเตอร์ที่เหมาะสมในการสกัดและตรวจนับต้นไม้ผลในแต่ละภาพที่ต้องการได้



ภาพ 6 หน้าต่างแสดงผล GUI MATLAB

## 2.3 การตรวจสอบความถูกต้องของเทคนิคกับข้อมูลที่ได้นับจริงจากการแปลภาพถ่ายด้วยสายตา

การตรวจสอบความถูกต้องในการประมาณค่าจะใช้วิธีทางสถิติด้วยการหาค่าความคลาดเคลื่อนเฉลี่ยสัมบูรณ์ (Mean Absolute Error; MAE) และค่ารากที่สองของความคลาดเคลื่อนยกกำลังสองเฉลี่ย (Root Mean Square Error; RMSE) ดังสองสมการด้านล่าง

$$MAE = \frac{1}{n} \sum_{i=1}^n |\text{Estimated value} - \text{Observed value}|$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [\text{Estimated value} - \text{Observed value}]^2}$$

โดยที่

Estimated value = จำนวนไม้ผลจากการตรวจนับอัตโนมัติ

Observed value = จำนวนไม้ผลจากการตรวจนับจริง

MAE = ความคลาดเคลื่อนเฉลี่ยสัมบูรณ์

RMSE = ค่ารากที่สองของความคลาดเคลื่อนยกกำลังสองเฉลี่ย

n = จำนวนข้อมูลทั้งหมด

#### ผลและอภิปรายผลการทดลอง

1. ผลการศึกษาและพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติจากข้อมูล google map

#### ตาราง 4 ผลการศึกษาและพัฒนาวิธีการตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูล google map

	ร้อยละความถูกต้อง			
	1. การตรวจนับ จำนวนวัตถุ จากขอบเขตของวัตถุ	2. การตรวจนับ จำนวนวัตถุ จากค่าสีของวัตถุ	3. การตรวจนับ จำนวนวัตถุ จากรัศมีของวัตถุ	ร้อยละ เฉลี่ย
orange	72%	78%	88%	79%
durian	77%	73%	87%	79%
ร้อยละ เฉลี่ย	75%	76%	88%	

จากตาราง 2 ผลการศึกษาและพัฒนาอัลกอริทึมที่ตรวจนับจำนวนไม้ผลอัตโนมัติจากข้อมูล google map จะพบว่าร้อยละความถูกต้องของเทคนิคที่ 1,2 และ 3 คือ 75%,76% และ 88% ตามลำดับ และผลลัพธ์ของเทคนิคที่ 3 มีร้อยละความถูกต้องมากที่สุด รองลงมาคือ เทคนิคที่ 2 และเทคนิคที่ 1 มีร้อยละความถูกต้องน้อยที่สุด ซึ่งจะเห็นได้ว่าผลลัพธ์ที่ได้ แสดงให้เห็นถึงความสำเร็จของอัลกอริทึมที่พัฒนาขึ้น พร้อมร้อยละความถูกต้องของการวิเคราะห์เกิน 70% แม้ว่าเทคนิคที่ 1 และ 2 จะมีร้อยละความถูกต้องน้อย แต่ความสามารถในการแยกสัญญาณยังคงอยู่ในระดับดี

2. ผลการพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

ตาราง 5 ผลการพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ

Area	Data	ร้อยละความถูกต้อง			ร้อยละเฉลี่ย
		1. การตรวจนับจำนวนวัตถุจากขอบเขตของวัตถุ	2. การตรวจนับจำนวนวัตถุจากค่าสีของวัตถุ	3. การตรวจนับจำนวนวัตถุจากรัศมีของวัตถุ	
สวนมะนาว	lemon (RGB)	77%	69%	90%	79%
สวนมะนาว	lemon (DSM)	86%	98%	99%	94%
สวนมะนาว	lemon (RGB)	51%	72%	78%	67%
สวนมะนาว	lemon (DSM)	80%	78%	99%	86%
ร้อยละเฉลี่ย		74%	79%	92%	

จากตาราง 3 ผลการพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับ พบว่าร้อยละความถูกต้องของเทคนิคที่ 1,2,3 คือ 74%,79% และ 92% ตามลำดับ พบว่าเทคนิคที่ 3 ให้ค่าความถูกต้องในแต่ละการทดลองมากที่สุด โดยที่การนำ DSM มาใช้จำแนกทำให้สามารถแยกต้นไม้ผลออกจากพื้นหลังได้ง่ายขึ้น เนื่องจากมีความแตกต่างของความสูงอย่างชัดเจน ส่วนการใช้ภาพออร์โธอย่างเดียวยังคงเป็นปัญหาเนื่องจากพื้นที่สวนมีค่าสีของพื้นหลังใกล้เคียงกับค่าสีของทรงพุ่มจะทำให้อัลกอริทึมไม่สามารถแยกพื้นหลังออกได้ ทำให้การสร้างเส้นขอบของวัตถุที่ใช้ canny มีความผิดพลาด

ผลลัพธ์ที่ได้สอดคล้องกับงานของ Dilek (2018)[3] ที่ได้พัฒนาอัลกอริทึมในการตรวจวัดจำนวนต้นมะนาวจากข้อมูลภาพอากาศยานไร้คนขับซึ่งพบว่าค่าความถูกต้องของการตรวจนับจำนวนต้นมะนาวที่ใช้ข้อมูลจาก DSM มีมากกว่าการใช้ภาพออร์โธ

### สรุปผลการศึกษา

การพัฒนาอัลกอริทึมเพื่อการตรวจนับจำนวนไม้ผลแบบอัตโนมัติด้วยข้อมูลภาพถ่ายจากอากาศยานไร้คนขับของ สวนมะนาว จ.สุโขทัย และสวนมะนาว จ.พิษณุโลก โดยการประมวลผลโฟโตแกรมเมตรี ด้วยโปรแกรม Pix4Dmapper เพื่อให้ได้ ภาพออร์โธ และภาพแบบจำลองพื้นผิวดิจิทัล



DSM แล้วทำการประมวลผลภาพดิจิทัล ด้วยโปรแกรม MATLAB เพื่อตรวจนับจำนวนไม้ผลัดโนมิตีสรุปผลดังนี้

1. การพัฒนาอัลกอริทึมเพื่อตรวจนับจำนวนไม้ผลัดโนมิตีจากข้อมูลภาพ Google เพื่อเป็นฐานในการพัฒนาอัลกอริทึมในภาพถ่ายจากอากาศยานไร้คนขับนั้น พบว่าค่าความถูกต้องในการตรวจนับยังมีค่าที่ต่ำ เนื่องจากการใช้ข้อมูลภาพอย่างเดียวในการจำแนกพื้นหลังกับทรงพุ่มของต้นไม้

2. อัลกอริทึมที่ถูกพัฒนาเพื่อการตรวจนับจำนวนไม้ผลัดโนมิตีจากข้อมูลภาพถ่ายทางอากาศด้วยอากาศยานไร้คนขับมีร้อยละความถูกต้องเฉลี่ยสูงเมื่อเทียบกับภาพ Google เนื่องจากมีการพัฒนาโค้ดเพิ่มเติมและยังนำข้อมูล DSM เข้ามาประกอบในการช่วยจำแนกพื้นหลังภาพกับทรงพุ่ม ทำให้มีการตรวจนับจำนวนไม้ผลัดโนมิตีที่ดีขึ้น

อย่างไรก็ตามยังมีข้อจำกัดบางประการ โดยพบว่าหากภาพที่ใช้ในการจำแนกมีลักษณะทรงพุ่มที่ไม่ชัดเจน หรือพื้นดินมีวัชพืชปกคลุม จะส่งผลให้การตรวจนับจำนวนไม้ผลัดโนมิตีผิดพลาด หากมีการใช้เซนเซอร์ช่วงคลื่นอินฟราเรดใกล้ร่วมกับช่วงคลื่นอินฟราเรดความร้อนจะทำให้การจำแนกทรงพุ่มของพืชสวนกับพื้นดินนั้นมีความถูกต้องมากยิ่งขึ้น

#### กิตติกรรมประกาศ

ขอขอบพระคุณ เจ้าของสวนมะนาว จ.สุโขทัย จ.พิษณุโลก นายดำรงณ์ พรหมมี และ นายประเสริฐ เรืองฉาย ที่ให้ความอนุเคราะห์ในการเข้าถ่ายพื้นที่ศึกษาโดย UAV นอกจากนี้ ขอขอบคุณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ให้ความอนุเคราะห์ซอฟต์แวร์ MATLAB ในการศึกษาครั้งนี้

#### เอกสารอ้างอิง

- [1] ธรารุณี บุญเหลือ. (2556). การประยุกต์ใช้อากาศยานไร้คนขับ เพื่อจัดทำภาพถ่ายทางอากาศสำหรับการออกแบบ ชุมชนเมือง : กรณีศึกษาพื้นที่ มหาวิทยาลัยมหาสารคาม วิทยาเขต ขามเรียง . วารสารมนุษยศาสตร์ และ สังคมศาสตร์มหาวิทยาลัยมหาสารคาม, ปีที่ 34, ฉบับที่ 5 (ก.ย.-ต.ค. 2558). หน้า 55-68.
- [2] มหศักดิ์ เกตุฉ่ำ. (2561) การหาขอบภาพ (Edge Detection). สืบค้นเมื่อ 10 มีนาคม 2562, จาก [www2.it.kmutnb.ac.th/teacher/FileDL/DrMahasak142256116161.pptx](http://www2.it.kmutnb.ac.th/teacher/FileDL/DrMahasak142256116161.pptx)
- [3] Dilek, k. (2018). Automatic citrus tree extraction from UAV images and digital surface models using circular Hough transform. Computers and Electronics in Agriculture, 150: pp.289-301
- [4] Gistda Academy. (2015). ระบบการแจ้งเตือนต้นไม้ในแปลงปลูกการเกษตร. สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน). สืบค้นเมื่อ 15 มีนาคม 2562, จาก <https://gistda.or.th/main/th/node/808>
- [5] Juthawut Chantharamalee. (2016). Digital image Processing. Retrieved March 25, 2019, from [http://dusithost.dusit.ac.th/~juthawut\\_cha/download/Chapter-01-Introduction-to-Digital-Image-Processing%20\(JC\).pdf](http://dusithost.dusit.ac.th/~juthawut_cha/download/Chapter-01-Introduction-to-Digital-Image-Processing%20(JC).pdf)



ลิขสิทธิ์ มหาวิทยาลัยนเรศวร

Copyright by Naresuan University

All rights reserved

## ประวัติผู้วิจัย

ชื่อ – ชื่อสกุล อุไรรัตน์ อุ่นเมือง  
 วัน เดือน ปี เกิด 4 กรกฎาคม 2544  
 ที่อยู่ปัจจุบัน 134/1 หมู่ 2 ตำบลสามพวง อำเภอกีรีมาศ  
 จังหวัดสุโขทัย 64160



### ประวัติการศึกษา

พ.ศ. 2559-ปัจจุบัน วท.บ. (ภูมิศาสตร์) มหาวิทยาลัยนเรศวร เกรดเฉลี่ย 3.21  
 พ.ศ. 2553-2558 ระดับมัธยมศึกษา โรงเรียนคีรีมาศพิทยาคม ตำบลโตนด  
 อำเภอกีรีมาศ จังหวัดสุโขทัย  
 พ.ศ. 2547-2552 ระดับประถมศึกษา โรงเรียนบ้านตะเข้ชานสามัคคีวิทยา  
 ตำบลสามพวง อำเภอกีรีมาศ จังหวัดสุโขทัย

### กิจกรรมที่เข้าร่วม

- 1) เข้าร่วมอบรมเชิงปฏิบัติการหัวข้อ การทำแผนที่ภูมิประเทศ วันที่ 23 กันยายน 2560 ณ คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก
- 2) เข้าร่วมอบรม Drone ideas (gistnext) หัวข้อ Conventional Survey VS Drone Survey วันที่ 12 มีนาคม 2562 ณ คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก
- 3) เข้าร่วมอบรม Let's become OSM Contributor (i-bitz company limited) หัวข้อ OpenStreetMap by JOSM และ Vector Tiled by TileserverGL วันที่ 12 เมษายน 2562 ณ คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก
- 4) เข้าร่วมโครงการเสริมความรู้เฉพาะด้านวิทยาศาสตร์ และเทคโนโลยีสารสนเทศ หัวข้อการประมวลผลภาพดิจิทัล (digital image processing) โดย ดร.ธราวิเชษฐ์ ธิติจรูญโรจน์ วันที่ 23-26 เมษายน 2562 ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ และเทคโนโลยีสารสนเทศ มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก
- 5) เข้าร่วมอบรมหลักสูตรภาษาอังกฤษ Cambridge English Placement Test Preparation (หลักสูตรระยะสั้น) ประจำปีเดือนพฤษภาคม 2562 ณ สถานพัฒนาวิชาการด้านภาษา มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก
- 6) เข้าร่วมอบรมหลักสูตรภาษาอังกฤษ Fundamentals of Writing (หลักสูตรระยะสั้น) ประจำปีเดือนกรกฎาคม 2562 ณ สถานพัฒนาวิชาการด้านภาษา มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก
- 7) เข้าร่วมอบรมหลักสูตรภาษาอังกฤษ Reading for Examinations (หลักสูตรระยะสั้น) ประจำปีเดือนกรกฎาคม 2562 ณ สถานพัฒนาวิชาการด้านภาษา มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

### ประวัติผู้วิจัย (ต่อ)

8) เป็นผู้ช่วยอบรมเรดาร์ตรวจอากาศ ของ ผศ.ดร.นัฐพล มหาวิค วันที่ 5-9 สิงหาคม 2562 ณ กรมอุตุนิยมวิทยา เขตบางนา กรุงเทพมหานคร

9) เข้าร่วมอบรมการเผยแพร่องค์ความรู้ด้านอุตุนิยมวิทยาและการใช้ข้อมูลเรดาร์ในการวิเคราะห์สภาพอากาศ วันที่ 11 กรกฎาคม 2562 ณ ห้องเมนคอนเฟอร์เร้นท์ อาคาร CITCOM มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

10) เป็นผู้ช่วยสอนภาคปฏิบัติการในรายวิชา Research Methodology in Geography (104411) ในหัวข้อ Remote sensing tutorial in ArcMap ของ ผศ.ดร.นัฐพล มหาวิค ประจำปีภาคการศึกษาที่ 1 ปีการศึกษา 2564 วันที่ 18 และ 20 กันยายน 2562 ณ คณะเกษตรศาสตร์ ทรัพยากรธรรมชาติและสิ่งแวดล้อม มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

11) เข้าร่วมอบรมงานการตรวจวัดอากาศของสถานีอุตุนิยมวิทยาพิษณุโลก วันที่ 28 กันยายน 2562 ณ สถานีอุตุนิยมวิทยา จังหวัดพิษณุโลก

12) เป็นผู้ช่วยอบรมการทำแผนที่จากโดรนเชิงปฏิบัติการ ของ ผศ.ดร.นัฐพล มหาวิค วันที่ 6 ตุลาคม 2562 ณ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา จังหวัดพิษณุโลก

13) เข้าร่วมงานสัมมนาเชิงปฏิบัติการ FOSS4G Thailand 2019 วันที่ 4-5 พฤศจิกายน 2562 ณ อาคารเฉลิมราชกุมารี 60 พรรษา (อาคารจามจุรี 10) จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพมหานคร

### รางวัลที่ได้รับ

1) เกียรติบัตรนิสิตที่มีพัฒนาการเรียนดี ประจำปีการศึกษา 2560 ระดับปริญญาตรี สาขาวิชา ภูมิศาสตร์ วันที่ 13 กันยายน 2561 ณ อาคารขวัญเมือง มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

2) เกียรติบัตรนิสิตที่มีพัฒนาการเรียนดี ประจำปีการศึกษา 2561 ระดับปริญญาตรี สาขาวิชา ภูมิศาสตร์ วันที่ 25 กรกฎาคม 2561 ณ อาคารขวัญเมือง มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

3) เกียรติบัตรนิสิตที่มีพัฒนาการเรียนยอดเยี่ยม ประจำปีการศึกษา 2561 ระดับปริญญาตรี สาขาวิชาภูมิศาสตร์ วันที่ 25 กรกฎาคม 2561 ณ อาคารขวัญเมือง มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

4) เกียรติบัตรนิสิตที่มีผลการเรียนดี ประจำปีการศึกษา 2561 ระดับปริญญาตรี สาขาวิชา ภูมิศาสตร์ วันที่ 25 กรกฎาคม 2561 ณ อาคารขวัญเมือง มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก