# การตรวจหาต้นไม้เป็นโรคโดยอัตโนมัติด้วยภาพถ่ายมุมสูงจากโดรน และวิธีการเรียนรู้เชิงลึก

พันจ่าอากาศโท รัตนโชติ พันธุ์วิไล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่
วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2562

# Automated plant disease detection using drones and deep learning

**Rattanachot Phwanwilai**

**A Thesis Submitted in Partial Fulfillment of the Requirements**

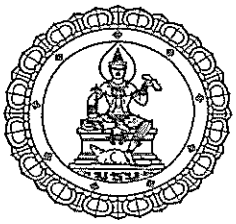
**for the Degree of Master of Engineering**

**Department of Big Data Engineering,**

**College of Innovative Technology and Engineering,**

**Dhurakij Pundit University**

**2019**



# ใบรับรองงานวิทยานิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ การตรวจหาต้นไม้เป็นโรคโดยอัตโนมัติด้วยภาพถ่ายมุมสูงจากโดรน และวิธีการเรียนรู้เชิงลึก

เสนอโดย พันจ่าอากาศโทรัตนโชติ พันธุ์วิไล

สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.ธนภัทร ฆังคะจิตร

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.กฤษณะ ไวยมัย)

.....กรรมการและอาจารย์ที่ปรึกษาหลัก

(ดร.ธนภัทร ฆังคะจิตร)

.....กรรมการ

(ดร.สรรพฤทธิ์ มฤคทัต)

.....กรรมการ

(ดร.เอกสิทธิ์ พัชรวงศ์ศักดา)

**วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว**

.....

(ผู้ช่วยศาสตราจารย์ ดร.ณรงค์เดช กีรติพรานนท์)

คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

วันที่ 28 เดือน พฤศจิกายน พ.ศ. 2562

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การตรวจหาต้นไม้เป็นโรคโดยอัตโนมัติด้วยภาพถ่ายมุมสูงจากโดรนและวิธีการเรียนรู้เชิงลึก |
| ชื่อผู้เขียน | พันจ่าอากาศโท รัตนโชติ พันธุ์วิไล |
| อาจารย์ที่ปรึกษา | ดร. ธนภัทร ฆังคะจิตร |
| สาขาวิชา | วิศวกรรมข้อมูลขนาดใหญ่ |
| ปีการศึกษา | 2562 |

## บทคัดย่อ

การทำการเกษตรสมัยใหม่ถือเป็นส่วนสำคัญส่วนหนึ่งที่ทำให้ประเทศไทยมีเศรษฐกิจที่แข็งแกร่งและมั่นคง ด้วยเทคโนโลยีที่ก้าวหน้าโดยเฉพาะการใช้โดรนช่วยในการทำเกษตรกรรมสามารถลดการใช้แรงงานในภาคเกษตรซึ่งส่งผลดีในการลดต้นทุนและการขาดแคลนแรงงานในภาคการเกษตรได้เป็นอย่างดี อย่างไรก็ตามการใช้ภาพถ่ายมุมสูงจากโดรนเพื่อสร้างโมเดลตรวจจับต้นไม้ที่เป็นโรคในสวนผลไม้ ประสบปัญหาในเรื่องข้อมูลไม่สมดุลย์เนื่องจากจำนวนข้อมูลที่มีน้อยมากของต้นไม้ที่เป็นโรค รวมถึงสภาพแวดล้อมที่แตกต่างกันของแต่ละสวนและมักจะมีต้นไม้หลายชนิดที่เกษตรกรปลูกร่วมกันในสวน ดังนั้นในงานวิจัยฉบับนี้จึงได้นำเทคนิคทางคอมพิวเตอร์วิทัศน์มาช่วยตรวจจับต้นไม้ที่เป็นโรคโดยอัตโนมัติโดยประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning) ซึ่งใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันกับภาพที่ได้จากโดรนที่ถ่ายจากมุมสูงภายในสวนของเกษตรกร โดยเสนอโมเดลแบบลำดับขั้น (Cascade Model) ในการตรวจจับต้นไม้ที่เป็นโรคใน 2 ขั้นตอน ขั้นตอนแรกทำการตรวจจับตำแหน่งต้นไม้ในรูปภาพ (Object detection) โดยใช้เทคนิค Mask R-CNN ขั้นตอนที่สองทำการจำแนกต้นไม้ที่เป็นโรคจากต้นไม้ที่ตรวจจับได้ในขั้นตอนแรก (Classification) โดยการใช้เทคนิคโอนถ่ายการเรียนรู้จากฐานข้อมูล ImageNet ของโมเดล VGG16 (Transfer Learning) ร่วมกับการใช้ฟังก์ชั่น Focal Loss ซึ่งให้ผลดีกับข้อมูลแบบไม่สมดุลย์ ผลการทดลองจากภาพถ่ายมุมสูงจากสวนทุเรียนตัวอย่างแสดงให้เห็นว่าโมเดลแบบลำดับขั้นที่นำเสนอให้ความแม่นยำที่สูงและครบถ้วนกว่าการใช้โมเดลในการตรวจจับต้นไม้ที่เป็นโรคเพียงอย่างเดียว รวมถึงลดความแตกต่างของค่าความน่าจะเป็นในกรณีที่มีการทำนายผิด ซึ่งทำให้สามารถนำไปแจ้งเตือนให้เกษตรกรตรวจสอบต้นไม้เหล่านี้อย่างละเอียดได้ในภายหลัง

| | |
|---|---|
| Thesis Title | Automated plant disease detection using drones and deep learning. |
| Author | FS2 Rattanachot Phwanwiali |
| Thesis Advisor | Dr. Thanapat Kangkachit |
| Department | Big Data Engineering |
| Academic Year | 2019 |

## ABSTRACT

Modern agriculture is one of the major parts which leads Thailand's economic becomes strong and stable. An advance technology especially agricultural drone could reduce using of labor which is beneficial in reduction of cost and labor shortages. However, using top-view images from drone for constructing a disease-tree detection model could face many problems i.e. imbalanced data (due to a small number of disease trees) and a variety of environments in the farms. In this research, the computer vision techniques were applied to detect the disease trees. A deep learning technique using a convolution neural network (CNN) together with the top-view images from drone was applied. In particular, a cascade CNN model is proposed to effectively detect disease trees in 2 steps. Firstly, a CNN object-detection model to detect the trees in top-view images is constructed using Mask R-CNN technique. Secondly, a CNN model to classify the disease trees using the pre-trained VGG16 model from ImageNet dataset with focal loss function (transfer learning technique). The results from the durian farms show that our proposed cascade CNN model provides higher precision and recall of detecting disease durian trees compared to using only a Mask R-CNN model. In addition, it also reduces the difference of probability in the misclassified trees which can be alerted to the farmers in order to further investigate.

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยการให้ความช่วยเหลือแนะนำของ ดร.ธนภัทร ฆังคะจิตร ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้กรุณาที่ให้คำแนะนำข้อคิดเห็นตรวจสอบ และแก้ไขร่างวิทยานิพนธ์มาโดยตลอด ผู้เขียนจึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้

ผู้เขียนขอกราบขอบพระคุณ รองศาสตราจารย์ กฤษณะ ไวยมัย ที่กรุณาให้เกียรติเป็นประธานโดยมี ดร.สรรพฤทธิ์ มฤคทัต เป็นกรรมการในการสอบวิทยานิพนธ์ ซึ่งได้กรุณาตรวจแก้ไขวิทยานิพนธ์ฉบับนี้ให้ถูกต้องสมบูรณ์ยิ่งขึ้น และ ดร.เอกสิทธิ์ พัชรวงศ์ศักดา ที่ให้คำปรึกษาแนะนำในทุก ๆ ด้านในการเขียนวิทยานิพนธ์ ตลอดจน นางสาวกุลธิดา รอดบุญ รวมถึงเจ้าหน้าที่บัณฑิตมหาวิทยาลัยธุรกิจบัณฑิตย์ ทุกท่านที่ให้ความสะดวกด้านอำนวยการ และประสานงาน ในการทำวิทยานิพนธ์ให้ผู้เขียนตลอดมาตลอดจนค้นคว้าหาข้อมูลในการจัดทำวิทยานิพนธ์ของผู้เขียนครั้งนี้สำเร็จลุล่วงไปด้วยดี

ท้ายนี้ผู้เขียนขอน้อมรำลึกถึงอำนาจบารมีของคุณพระศรีรัตนตรัย และสิ่งศักดิ์สิทธิ์ทั้งหลายที่อยู่ในสากลโลก อันเป็นที่พึ่งให้ผู้เขียนมีสติปัญญาในการจัดทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี ผู้เขียนขอให้เป็นกตเวทิตาแด่บิดา มารดา ครอบครัวของผู้เขียน ตลอดจนผู้เขียนหนังสือ และบทความต่าง ๆ ที่ให้ความรู้แก่ผู้เขียนจนสามารถให้วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

รัตนโชติ พันธุ์วิไล

# สารบัญ


| | หน้า |
|---|---|
| บทคัดย่อภาษาไทย | ฆ |
| บทคัดย่อภาษาอังกฤษ | ง |
| กิตติกรรมประกาศ | จ |
| สารบัญตาราง | ซ |
| สารบัญภาพ | ฌ |
| บทที่ | |
| 1. บทนำ | 1 |
| 1.1 ที่มาและความสำคัญของปัญหา | 1 |
| 1.2 วัตถุประสงค์ของการวิจัย | 2 |
| 1.3 ประโยชน์ที่คาดว่าจะได้รับ | 2 |
| 1.4 ขอบเขตงานวิจัย | 2 |
| 1.5 นิยามศัพท์ | 3 |
| 2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง | 4 |
| 2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง | 4 |
| 2.1.1 วิธีการเรียนรู้เชิงลึก (Deep Learning) | 4 |
| 2.1.2 การตรวจจับวัตถุ (Object Detection) | 4 |
| 2.1.3 โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network) | 7 |
| 2.1.4 Mask R-CNN | 14 |
| 2.1.5 Transfer Learning การเรียนรู้แบบถ่ายโอน | 16 |
| 2.1.6 การวัดผล | 19 |
| 2.2 งานวิจัยที่เกี่ยวข้อง | 28 |
| 3. ระเบียบวิธีวิจัย | 35 |
| 3.1 การสร้างแบบจำลองโมเดล | 35 |
| 3.1.1 วิธีการเก็บรวบรวมข้อมูล (Data Collection) | 36 |

3.1.2 การเตรียมข้อมูล (Data Preparation) 37
3.1.3 การสร้างแบบจำลอง (Model Training) 40
3.1.4 การประเมินผล (Model Evaluation) 43
3.2 การนำแบบจำลองโมเดลไปใช้งาน 44
4. ผลการศึกษา 46
4.1 ผลการปรับแต่งโมเดล 46
4.1.1 การปรับแต่ง Mask R-CNN 48
4.1.2 การปรับแต่ง Mask R-CNN + VGG16 46
5. บทสรุปและข้อเสนอแนะ 56
5.1 สรุปผลศึกษา 56
5.2 อภิปรายผลการศึกษา 56
5.3 ข้อเสนอแนะ 56
6. บรรณานุกรม 57
7. ภาคผนวก 62
7.1 ภาพจากผลการทดลอง 63
7.2 ผลงานทางวิชาการ 64
7.3 ประวัติผู้เขียน 70

# สารบัญตาราง


| ตารางที่ | หน้า |
|---|---|
| 2.1 ตารางตัวอย่างการหาค่า Precision Recall | 23 |
| 2.2 ตารางการหาค่า Precision Recall | 26 |
| 3.1 แทนชื่อต้นทุเรียนโดยใช้ชื่อภาษาอังกฤษ | 38 |
| 3.2 จำนวนของ Train, Val และ Test ที่ได้จากการ Mask | 38 |
| 3.3 ตัวอย่างการทำ Augmentation process ทั้ง 5 แบบ | 39 |
| 3.4 การตั้งค่าพารามิเตอร์ Mask R-CNN | 41 |
| 3.5 การตั้งค่าพารามิเตอร์ Transfer Learning | 42 |
| 4.1 ผลการทดลองของ Mask R-CNN กับ Mask R-CNN + VGG16 | 54 |

# สารบัญภาพ


| ภาพที่ | หน้า |
|---|---|
| 2.1 ตัวอย่างการทำ Sliding Window | 5 |
| 2.2 ตัวอย่างการทำ Region proposal | 6 |
| 2.3 R-CNN | 6 |
| 2.4 โครงสร้างของ Neuron | 8 |
| 2.5 Feed-Forward Neural Network | 8 |
| 2.6 Convolutional Neural Network | 9 |
| 2.7 Input Filter/Kernel | 10 |
| 2.8 Feature Map | 10 |
| 2.9 การทำ Padding | 11 |
| 2.10 การทำ Max pooling | 12 |
| 2.11 การทำ Min pooling | 12 |
| 2.12 การทำ Average pooling | 13 |
| 2.13 การทำ Fully Connected | 13 |
| 2.14 Mask R CNN | 14 |
| 2.15 Instance Segmentation | 14 |
| 2.16 residual modules | 15 |
| 2.17 Region proposal Network | 16 |
| 2.18 Transfer Learning | 17 |
| 2.19 โครงสร้าง VGG16 | 18 |
| 2.20 VGG16 Architecture | 18 |
| 2.21 area of IoU | 20 |
| 2.22 ตัวอย่างการซ้อนทับ | 20 |
| 2.23 การหา IoU | 21 |
| 2.24 IoU > 0.5 = TP | 22 |
| 2.25 IoU > 0.5 and Wrong Class = FN | 22 |

## สารบัญภาพ(ต่อ)


| ภาพที่ | หน้า |
|---|---|
| 2.26 IoU < 0.5 and Duplicated BB | 23 |
| 2.27 การ pot Precision Recall curve | 24 |
| 2.28 ตัวอย่างตาราง Confusion Matrix | 25 |
| 2.29 ภาพจากการทำนาย | 25 |
| 2.30 ภาพจาก Ground Truth | 25 |
| 2.31 ภาพจาก IoU | 26 |
| 2.32 Precision - Recall Curve | 27 |
| 2.33 งานวิจัยที่เกี่ยวข้องที่ 1 | 29 |
| 2.34 งานวิจัยที่เกี่ยวข้องที่ 2 | 29 |
| 2.35 งานวิจัยที่เกี่ยวข้องที่ 3 | 30 |
| 2.36 งานวิจัยที่เกี่ยวข้องที่ 4 | 31 |
| 2.37 งานวิจัยที่เกี่ยวข้องที่ 5 | 32 |
| 2.38 งานวิจัยที่เกี่ยวข้องที่ 6 | 33 |
| 2.39 งานวิจัยที่เกี่ยวข้องที่ 7 | 34 |
| 2.40 งานวิจัยที่เกี่ยวข้องที่ 8 | 36 |
| 3.1 แผนผังการสร้างแบบจำลอง (Trained Model) โดยรวม | 35 |
| 3.2 ตัวอย่างไฟล์ที่ 1 | 36 |
| 3.3 แสดงการรวบรวมและแปลงข้อมูล | 37 |
| 3.4 ภาพตัวอย่างที่เป็นสวนลำใย | 39 |
| 3.5 แผนผังการสร้างแบบจำลอง (Trained Model) ในส่วนของ Mask R-CNN | 40 |
| 3.6 แผนผังการสร้างแบบจำลอง (Trained Model) ในส่วนของ VGG16 | 42 |
| 3.7 ตัวอย่างการ Mask ที่ใช้แปรแกรม via-1.0.6 | 42 |
| 3.8 ภาพตัวอย่างการ Mask ต้นทุเรียนที่เป็นโรคและไม่เป็นโรค | 43 |
| 3.9 ภาพที่ได้จาก Ground Truth | 43 |

3.10 ภาพจาก IoU Mask R-CNN 44

3.11 แผนผังการนำแบบจำลองไปใช้งาน (Testing) 44

3.12 ผลลัพธ์จากโมเดลที่ 2 ที่แยะแยะว่าเป็นต้นอะไร 45

4.1 แสดงผลของ mAP, Recall และ f1 ของการ mask ทุกต้นในรูป และ mask เท่ากัน 46

4.2 เปรียบเทียบผลลัพธ์ Confusion matrix ของ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss 47

4.3 เปรียบเทียบผลลัพธ์ Precision, Recall และ F1-Score ของ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดย ใช้ ฟังก์ชั่น Loss เป็น focal loss 47

4.4 ตัวอย่างการแยกแยะต้นที่เป็นโรค ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss 48

4.5 เปรียบเทียบผลลัพธ์ ตัวอย่างการแยกแยะต้นมังคุด ของ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดย ใช้ ฟังก์ชั่น Loss เป็น focal loss 49

4.6 ตัวอย่างการแยกแยะต้นที่ไม่เป็นโรคของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss มีค่ามากกว่า[1] 49

4.7 ตัวอย่างการแยกแยะต้นที่ไม่เป็นโรค ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss มีค่ามากกว่า[2] 50

4.8 เปรียบเทียบผลลัพธ์ จำนวนการทายผิด - ถูก ของโมเดล Mask R–CNN +VGG16 โดย ใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss 50

4.9 เปรียบเทียบผลลัพธ์ ช่วงที่มี Confidence ที่ทำนายผิด ของโมเดล Mask R–CNN +VGG16 โดย ใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss มีค่ามากกว่า 51

4.10 แสดงการเปรียบเทียบจำนวนการทายผิด ของโมเดล Mask R–CNN +VGG16 52

โดย ใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ ฟังก์ชั่น Loss เป็น focal loss มีค่ามากกว่า[1]

4.11 แสดงการเปรียบเทียบ Probability Categorical Cross entropy 52

4.12 แสดงการเปรียบเทียบของช่วงที่มี Probability Focal loss 53

4.13 แสดงกราฟแผนภูมิคอลัมน์แบบเรียงซ้อน 100% พร้อมทั้งตัด Probability 53

4.14 รูปที่ Mask R-CNN ทำการค้นหาและระบุตำแหน่ง 54

4.15 รูปที่ Mask R-CNN ทำการค้นหาและระบุตำแหน่ง และตัดมาทีละต้น 54

# บทที่ 1
# บทนำ

**1.1** ที่มาและความสำคัญของปัญหา

ประเทศไทยโดยพื้นฐานแล้วเป็นประเทศเกษตรกรรม ประเทศไทยมีพื้นที่ประมาณ 321 ล้านไร่ หรือประมาณ 513,000 ตร.กม. โดยมีพื้นที่สำหรับการเกษตรประมาณ 43 % หรือ (138 ล้านไร่) โดยมีรายได้ภาคเกษตรคิดเป็น 10% ของ GDP (ข้อมูลปี 2561) ด้วยพื้นที่ทางการเกษตรในประเทศไทยที่ได้กล่าวมาข้างต้นมีพื้นที่กว้างขวาง การที่เกษตรจะไปตรวจสอบหรือบำรุงรักษาพืชผลทางการเกษตร นั้นเป็นไปได้ยากลำบากเพราะต้องใช้กำลังคนและสิ้นเปลืองงบประมาณทำให้ต้องกำหนดราคาสินค้าการเกษตรที่สูงขึ้น ในปัจจุบันมีเทคโนโลยีที่ก้าวหน้า ทำให้เกษตรกรลดต้นทุนและการขาดแคลนแรงงานในภาคการเกษตรเป็นอย่างดี โดยการประยุกต์ใช้โดรนเพื่อตรวจสอบพืชผลทางการเกษตร

ปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) ได้มีการพัฒนาก้าวหน้าไปอย่างรวดเร็วโดยมีการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning) โดยเฉพาะการใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Networks) ซึ่งมีโครงสร้างที่ใช้ในการสกัดคุณลักษณะสำคัญ (Feature extraction) แล้วจึงนำคุณลักษณะเหล่านั้นมาช่วยในการสร้างโมเดลที่มีประสิทธิภาพสูงในการทำนายวัตถุและตำแหน่งการพบวัตถุในภาพหรือวิดีโอ

ปัจจุบันมีหลายงานวิจัยที่ใช้เทคโนโลยีทางด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) ทำการตรวจหาโรคพืชจากภาพถ่ายใบไม้โดยประยุกต์ใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน ดังเช่น งานวิจัย การวินิจฉัยจำแนกโรคใบองุ่นจากภาพถ่ายโดยใช้จีเนติกส์อัลกอริทึมและแผนผังจัดการตนเองเชิงโครงสร้างปรับตัวได้ กันยายน ปี 2018 โดยใช้ อัลกอริทึม GA-SASOM (ชมพูทรัพย์ ปทุมสิน, อาทิตย์ ศรีแก้ว. (2562)), งานวิจัยเป็นงานวิจัยที่ ตรวจหาโรคพืช 2018 เป็นการใช้ CNN Architecture จำนวน 17 model (M. Brahimi, et al ., M. Arsenovic, et al ., S. Laraba, et al ., S. Sladojevic, et al ., K. Boukhalfa, et al ., A. Moussaoui. (2018)), งานวิจัยเป็นงานวิจัยที่ตรวจหาศัตรูพืชโดยใช้ภาพถ่ายจากโดรนแล้วใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน Caffe Net ในการจำแนกศัตรูโรคพืชที่เป็นไม้พุ่ม Y. Bouroubi, et al ., P. Bugnet, et al ., T.Nguyen-Xuan, et al ., C.Gosselin, et al ., C. Bélec, et al ., L.Longchamps, et al ., P.Vigneault.(2018), ซึ่งยังไม่สารมารถใช้กับพืชผลทางการเกษตรของประเทศไทยซึ่งมีลักษณะเป็นไม้ยืนต้น งานวิจัยที่กล่าวมาข้างต้นให้ผลลัพธ์ค่อนข้างดี แต่ยังไม่มีการขยายส่วนข้อมูลอินพุตภาพถ่ายใบไม้ (Augmentation) ทำให้

อาจจะไม่ทนทานต่อภาพที่มีสิ่งรบกวน (noise), มีข้อจำกัดเรื่องการใช้กลุ่มตัวอย่างที่น้อยและไม่สามารถตรวจจับตำแหน่งที่เกิดโรคในใบของพืชได้

งานวิจัยนี้ได้นำเสนอโมเดลที่เป็นแบบลำดับขั้น (Cascade model) เพื่อตรวจจับต้นไม้ที่เป็นโรคโดยใช้ภาพถ่ายมุมสูงจากโดรนโดยการประยุกต์การตรวจจับการใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Networks) โดยโมเดลแรกจะเป็นการค้นหาและระบุตำแหน่งต้นไม้ที่เป็นโรคโดยใช้ CNN ที่ชื่อว่า Mask R-CNN ส่วนโมเดลที่สองจะเป็นการแยกแยะต้นไม้ที่เป็นโรค และไม่เป็นโรค จากการทดสอบภาพตัวอย่างจากสวนของเกษตรกรในจังหวัดจันทบุรีพบว่า โมเดลสามารถตรวจเจอต้นไม้ที่เป็นโรคซึ่งได้ผลลัพธ์เป็นที่น่าพอใจ โดยมีค่า precision เป็น 0.86%, recall เป็น 0.85% และ f1 เป็น 0.85% (ในส่วนของต้นไม้ที่เป็นโรค) แต่อย่างไรก็ตามสามารถพัฒนาวิธีการให้มีผลลัพธ์ได้ดีขึ้น

**1.2** วัตถุประสงค์ของงานวิจัย

1.2.1 จัดการข้อมูล ต้นที่เป็นโรค และ ไม่เป็นโรค ที่มีสัดส่วนไม่สมดุลย์กัน

1.2.2 ลดจำนวน False Positive จากการทำนาย ต้นไม้ที่ไม่เป็นโรค เป็น ต้นที่เป็นโรค

**1.3** ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 ประโยชน์ทางด้านเกษตรกรรม

- ช่วยเกษตรกรในการตรวจสอบเฝ้าระวังการเกิดโรคของต้นไม้
- ช่วยเกษตรกรในการทำนายต้นไม้ที่เป็นโรค
- ช่วยเกษตรกรในการลดต้นทุน, ลดค่าใช้จ่าย และลดแรงงานในภาคการเกษตร
- ช่วยเพิ่มรายได้ของเกษตรกร
- เกษตรมีความมั่นคงทางอาชีพอย่างยั่งยืน
- การสูญเสียทรัพยากรธรรมชาติและสิ่งแวดล้อมจากการผลิตทางการเกษตรลดลง

1.3.2 ประโยชน์ทางด้านทฤษฎี

- ได้วิธีการสร้างชุดข้อมูลภาพถ่ายมุมสูงจากโดรนที่เหมาะสมเพื่อเป็นแนวทางสำหรับผู้ที่สนใจได้ประยุกต์และนำไปใช้ให้เกิดประโยชน์ในอนาคตได้
- ได้วิธีการออกแบบและสร้างโมเดลเพื่อใช้กับข้อมูลภาพถ่ายมุมสูงจากโดรน
- ได้วิธีการแก้ปัญหาสภาพแวดล้อมต่าง ๆ ที่มีผลต่อการ โมเดล เช่น ข้อมูลไม่สมดุลย์, ลักษณะของต้นไม้, แสงเงา, มุม และความสูง
- ได้ทราบวิธีการวัดผลแบบต่าง ๆ

**1.4** ขอบเขตงานวิจัย

1.4.1 ขอบเขตของข้อมูล อธิบายถึงข้อมูลที่นำมาใช้ในงานวิจัย

ภาพที่ได้ต้องแปลงจากไฟล์ VDO ที่มีขนาด 2701 * 1152

ภาพถ่ายมุมสูงจากโดรนเท่านั้น

ภาพที่มีลักษณะเป็น Bird eye view

ภาพที่มีนามสกุล jpg ที่มีขนาด 1152 * 1920

ความสูงของโดรนระหว่างต้นไม้ 3 – 4 เมตร

เวลาการบินเพื่อบันทึกภาพจากโดนช่วงเวลากลางวัน

**1.5** นิยามศัพท์

1.5.1 **GDP** หมายถึง ผลิตภัณฑ์รวมในประเทศเป็นมูลค่าของสินค้าในประเทศขั้นสุดท้ายของประเทศนั้น ๆ

1.5.2 **Computer Vision** หมายถึง การฝึกฝนคอมพิวเตอร์ และระบบให้สามารถเข้าใจและตอบสนองต่อข้อมูลภาพได้อย่าง ชาญฉลาด ด้วยภาพดิจิทัลจากกล้องถ่ายภาพและวิดีโอ

1.5.3 **Deep learning** หมายถึง ซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบการทำงานของระบบโครงข่ายประสาท(neurons)ในสมองมนุษย์

1.5.4 **Convolution Neural Networks** หมายถึง โครงข่ายประสาทแบบคอนโวลูชัน เป็นโครงข่ายประสาทเทียมชนิดหนึ่ง

1.5.5 **Precision** หมายถึง เป็นคำที่นิยมใช้และแสดงความหมายใกล้เคียงกับความถูกต้องแม่นยำ

1.5.6 **Recall** หมายถึง ค่าความถูกต้อง นำค่า True positive มาเทียบกับ False negative

1.5.7 **F1** หมายถึง ค่าเฉลี่ยของ Precision และ Recall

1.5.8 **Augmentation process** หมายถึง ขยายส่วนข้อมูลอินพุตภาพที่ได้รับเข้ามาเพื่อให้มีข้อมูลมีความหลากหลาย

1.5.9 **Cascade model** หมาถึง โมเดลเป็นที่เป็นระดับชั้น ภายในอาจจะประกอบด้วยโมเดลหลายๆ โมเดลทำงานร่วมกัน

1.5.10 **Transfer learning** หมายถึง การส่งผ่านการเรียนรู้โดยใช้โมเดลที่มีการพัฒนา

1.5.11 **Ground Truth** หมายถึง ผลเฉลยของชุดข้อมูล ที่ใช้เปรียบเทียบคำตอบกับโมเดล

1.5.12 **Class** หมายถึง ประเภทของชุดข้อมูลที่ใช้ในการสอนโมเดล

1.5.13 **Imbalanced data** หมายถึง ชุดข้อมูลที่มี Class ไม่เท่ากันหรือต่างกันมาก ๆ

1.5.14 **Under sampling** หมายถึง การลดจำนวนชุดข้อมูลในกรณี ที่เป็น Imbalanced data

1.5.15 **Over sampling** หมายถึง การเพิ่มจำนวนชุดข้อมูลในกรณี ที่เป็น Imbalanced data

1.5.15 **False Positive** หมายถึง ผลทดสอบที่ออกมาว่าเป็นบวกแต่นั้นเป็นผลที่ไม่ถูกต้อง

# บทที่ 2
# แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

## 2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

### 2.1.1 วิธีการเรียนรู้เชิงลึก (Deep Learning)

วิธีการเรียนรู้เชิงลึก เป็นซอฟต์แวร์คอมพิวเตอร์ที่เลียนแบบให้คล้ายกับเครือข่ายเซลล์ประสาทในสองของมนุษย์ (Network of Neuron) ซึ่ง Deep Learning นั้นเป็นส่วน หนึ่งเล็ก ๆ อยู่ใน Machine Learning ส่วน Algorithm ของ deep learning ถูกสร้างขึ้นจากการนำเอา Neural Network หลายๆ layer มาต่อกัน โดย layer แรกสุดจะทำหน้าที่ในการรับข้อมูล (Input layer) layer สุดท้ายจะทำหน้าที่ส่งผลลัพธ์การประมวลผลออกมา (Output layer) ส่วน layer ระว่าง layer แรกสุด และ layer สุดท้าย จะถูกเรียกว่า Hidden layer โดยที่ Hidden layer ของแต่ละ layer จะเปรียบเสมือนว่าประกอบด้วย เซลล์ประสาท (neural) จำนวนมาก มีหน้าที่ในการประมวลผล รับข้อมูลจาก layer ที่อยู่เหนือกว่า และส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยัง layer ที่อยู่ล่างกว่า ข้อดีของการส่งข้อมูลแบบนี้ก็คือ layer แต่ละ layer สามารถที่จะมี ค่าถ่วงน้ำหนัก (weight) ค่าความเอนเอียงของข้อมูล (bias) และ วิธีการประมวลผลทางคณิตศาสตร์ (activation function) ที่เป็นอิสระต่อกัน เมื่อมีข้อมูล input ให้กับ model มากเท่าไหร่ layer แต่ละ layer ก็จะสามารถสกัด feature ที่มีความซับซ้อนมากยิ่งขึ้น

### 2.1.2 การตรวจจับวัตถุ (Object Detection)

การที่จะให้คอมพิวเตอร์มีความสามารถจำแนกวัตถุที่อยู่ในรูปภาพออกมาแล้วทำนายว่าวัตถุเป็น Class อะไร ต้องอาศัยหลักการของ Object Classification และการที่จะให้คอมพิวเตอร์หาตำแหน่งของวัตถุที่อยู่ต้องอาศัยหลักการของ Object Detection การทำ Object Classification จะสรุปได้ดังนี้ Input Image ⟶ Feature Extractor ⟶ Classifier ⟶ Output Class โดยมีขั้นตอนดังนี้เป็น นำข้อมูลเข้า เพื่อสกัดหาคุณลักษณะที่สำคัญ แล้วทำ Classifier เพื่อให้ได้ Output ของ Class ขั้นตอนถัดมาเข้าสู่กระบวนการ Object Detection ซึ่งมีหลายแบบดังนี้

- Sliding Window

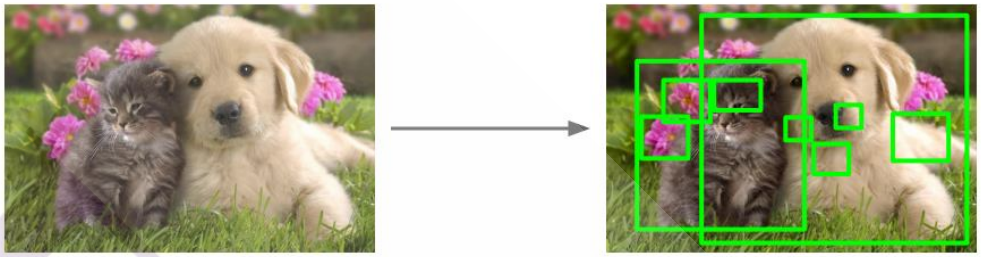
  เป็นวิธีการทำ Object Detection ที่นิยมมากในสมัย10 ปีที่ผ่านมา มีการทำงานจะนำเอาผลลัพธ์จากการทำ Object Classification ดังนี้ ทำการตีกรอบแล้วขยับไปเรื่อยๆ จากนั้นภาพที่ได้จากการขยับ เข้าไปทำ Feature Extractor และทำการ Classifier ว่าตรงกับ Class อาจจะมีการขยับทีละ 1 pixel ไปเรื่อยๆ จนหมด จะเกิดปัญหาขึ้นเพราะถ้าเรากำหนดกรอบ ขนาด (32X32) และมีภาพขนาด (512X512) เวลาทำ Sliding Window จะต้องใช้พลังงานของเครื่องมหาศาลถึงจะเสร็จ ซึ่งในปัจจุบันมีการแก้ปัญหาแล้ว มีการย่อขยายรูปทุกครั้งที่มีการทำงานเสร็จ ย่อรูปขนาดตั้งแต่ 0.1,0.2……,0.9 ดังภาพที่ 2.1

**ภาพที่ 2.1** ตัวอย่างการทำ Sliding Window

- Selective Search

  เป็นอัลกอริทึมข้อเสนอ region proposal ที่ใช้ในการตรวจจับวัตถุ มันถูกออกแบบมาให้รวดเร็วพร้อมการเรียกคืนที่สูงมาก ขึ้นอยู่กับการคำนวณการจัดกลุ่มลำดับชั้นของภูมิภาคที่คล้ายกันขึ้นอยู่กับสีพื้นผิวขนาดและความเข้ากันได้ของรูปร่าง การค้นหาแบบเลือกเริ่มต้นด้วยการแบ่งส่วนภาพตามความเข้มของ

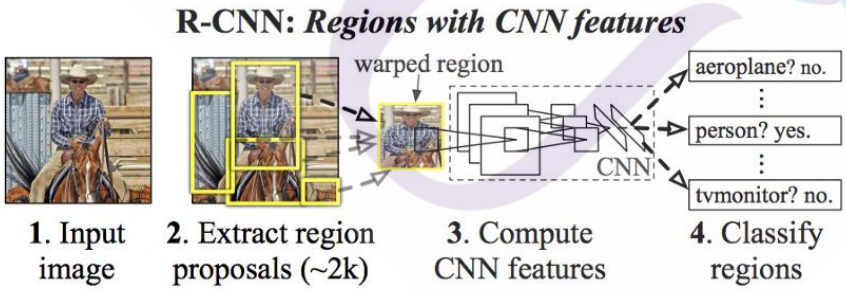
พิกเซลโดยใช้วิธีแบ่งกลุ่มตามกราฟโดย Felzenszwalb และ Huttenlocher ดังภาพที่ 2.2



**ภาพที่ 2.2** ตัวอย่างการทำ region proposal

- R-CNN

  เป็นการนำ region proposals จำนวน 2,000 แปรสภาพเป็นสี่เหลี่ยมจัตุรัสและป้อนเข้าสู่โครงข่ายประสาทเทียมที่สร้างเวกเตอร์ที่มีคุณลักษณะ 4096 มิติเป็นเอาต์พุต CNN ทำหน้าที่เป็นตัวแยกคุณลักษณะและเลเยอร์หนาแน่นของเอาต์พุตประกอบด้วยคุณลักษณะที่ดึงออกมาจากรูปภาพและฟีเจอร์ที่แยกแล้วจะถูกป้อนเข้าสู่ SVM เพื่อจำแนกการมีอยู่ของวัตถุภายใน ดังภาพที่ 2.3



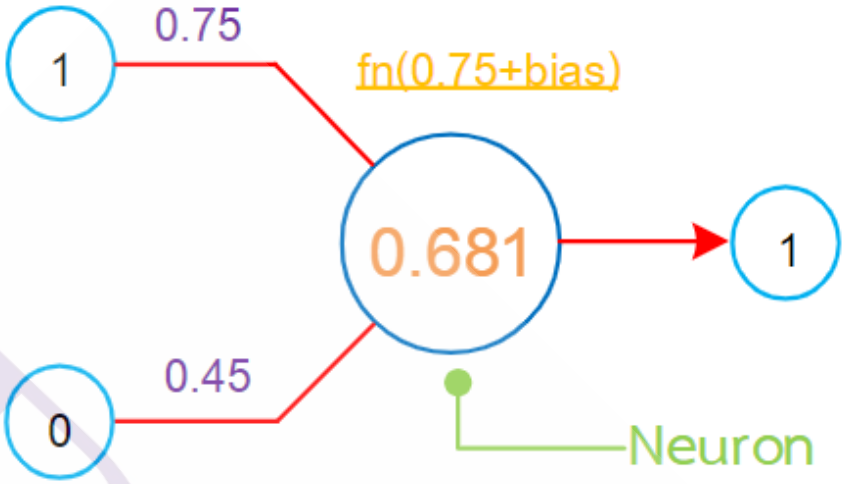

**ภาพที่ 2.3** R-CNN

### 2.1.3 โครงข่ายประสาทเทียมแบบคอนโวลูชั่น (Convolutional Neural Network)

**Neural Network** เป็น โมเดลทางคณิตศาสตร์ที่จำลองกระบวนการคิดจากสมองมนุษย์ โดยสมองนั้นจะมีหน่วยประมวลผลขนาดเล็กอยู่เยอะมาก และเชื่อมโยงกันด้วยโครงข่ายประสาทมากมายทำให้มนุษย์เรียนรู้และคิดวิเคราะห์ได้อย่างรวดเร็ว แต่คอมพิวเตอร์นั้นไม่ได้มีโครงข่ายที่ซับซ้อนเหมือนกับสมองของมนุษย์ คอมพิวเตอร์มีหน้าที่แค่รันโปรแกรมตามคำสั่งของเท่านั้น เมื่อให้คอมพิวเตอร์ทำการเรียนรู้อะไรซักอย่าง จึงเป็นเรื่องยากในรูปแบบปกติ จึงเกิดการจำลองแนวทางการเรียนรู้ของคน ไปสู่คอมพิวเตอร์ด้วย Neural Network ส่วนที่เล็กที่สุดคือ Neuron ซึ่งทำหน้าที่คำนวน มีส่วนประกอบดังนี้

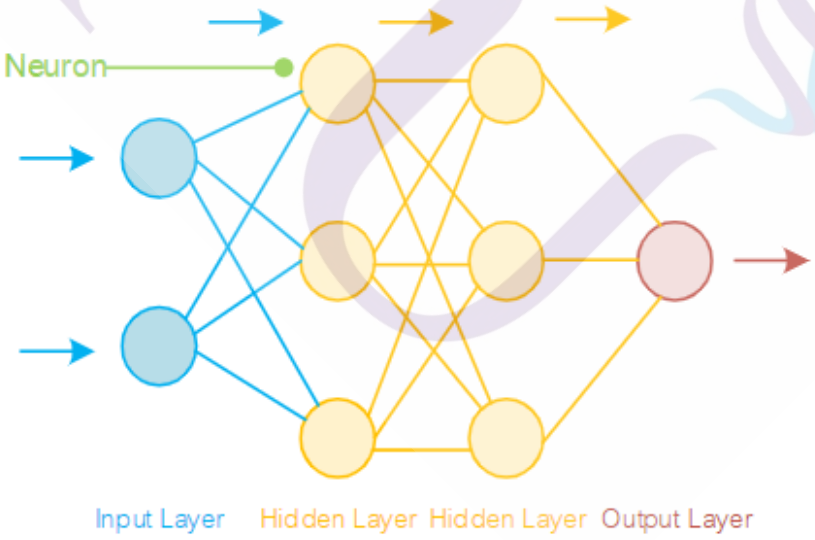
- Input เป็นค่าที่ส่งเข้ามาที่ Neuron โดยจะมีขาที่เข้ามาได้หลายขา โดยทั่วไปมักจะเท่ากับจำนวน Class
- Weight เป็นการให้น้ำหนักของขาแต่ละที่ส่งเข้ามา โดยมีค่าระหว่าง 0-1 โดยที่ทำการสุ่มขึ้นมาจากนั้น Neuron เมื่อทำการเรียนรู้เรื่อย ๆ ก็จะเป็นการปรับ weight เพื่อให้ได้คำตอบที่ใกล้เคียงที่สุด
- Bias คือค่าที่จะช่วยเข้ามาทำให้ค่าที่เข้ามาอยู่ในระหว่าง 0 - 1 ได้ โดยจะเป็นเลขที่สุ่ม และปรับไปเรื่อย ๆ ทุกครั้งที่เรียนรู้
- Output คือผลลัพธ์
- Back Propagation คือการที่ Neuron นำค่า Error ของ Output ที่ได้ กับ Output ที่เรียนรู้ นำไปปรับ Weight และ Bias ให้เกิดผลลัพธ์ที่ถูกต้องตามที่ได้เรียนรู้

นั้นเมื่อป้อน Input และ Output ให้มัน เพื่อทำการเรียนรู้ ตัว Neuron ก็จะทำการบวกตัว Input ด้วย Weight คูณกับค่าที่มาของแต่ละขา จากนั้นบวกค่า bias แล้วนำไปเข้าฟังก์ชันที่ตัว Neuron กำหนดไว้ ก็จะเป็น Output โดยฟังก์ชันก็จะมีหลายแบบขึ้นอยู่รูปแบบที่จะนำไปใช้งาน เช่น Sigmoid, Hyperbolic Tangens, Hard Limit, Rectified Linear Unit เป็นต้น ดังภาพที่ 2.4



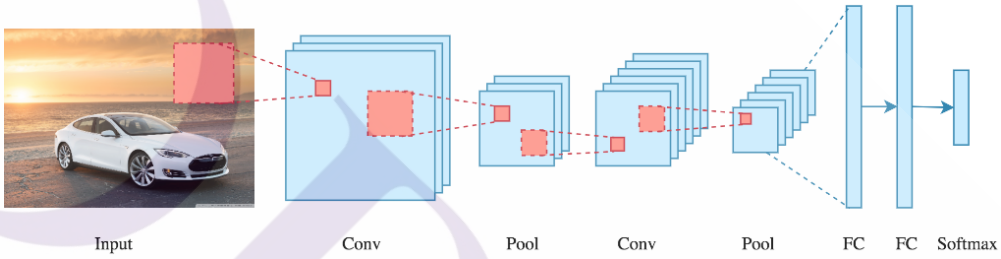

**ภาพที่ 2.4** โครงสร้างของ Neuron

Feed-Forward Neural Network รูปแบบของ Neural Network ที่ง่ายที่สุดคือ Feed-Forward Neural Network โดยจะแบ่ง Neuron ออกเป็นกลุ่มๆ โดยแต่ละกลุ่มจะเรียกเป็น Layer โดยข้อมูลที่เข้ามาจะไหลไปในทิศทางเดียว ไม่ไหลย้อนกลับ จาก Layer หนึ่งสู่อีก Layer ดังภาพที่ 2.5




**ภาพที่ 2.5** Feed-Forward Neural Network

**Convolutional Neural Network(CNN)** เป็น โครงข่ายประสาทเทียมแบบคอนโวลูชัน เป็นกระบวนการที่สกัดเอาลักษณะที่สำคัญของภาพออกมา โดยใช้ค่า Pixel ซึ่งได้จากข้อมูลอินพุต มีทั้งหมด 3 แชนแนว (Channael) ได้แก่ สีแดง, น้ำเงิน และเขียว สามารถใช้เลข 0 ถึง 255 เพื่อเป็นค่าแทนความเข้มของสี เนื่องจาก CNN ไม่ค่อยเป็นที่นิยมเพราะต้องใช้ประสิทธิภาพของเครื่องคอมพิวเตอร์อย่างมากแต่ปัจจุบันมีการพัฒนาหน่วยประมวลผลกราฟฟิก (GPU) ให้รองรับการคำนวณจึงทำให้ CNN ประสบความสำเร็จในการแก้ปัญหา Classification ที่เกี่ยวข้องกับรูปภาพ อีกทั้งในปัจจุบันได้นำแนวคิดไปใช้ต่อยอดมากมาย ดังภาพที่ 2.6

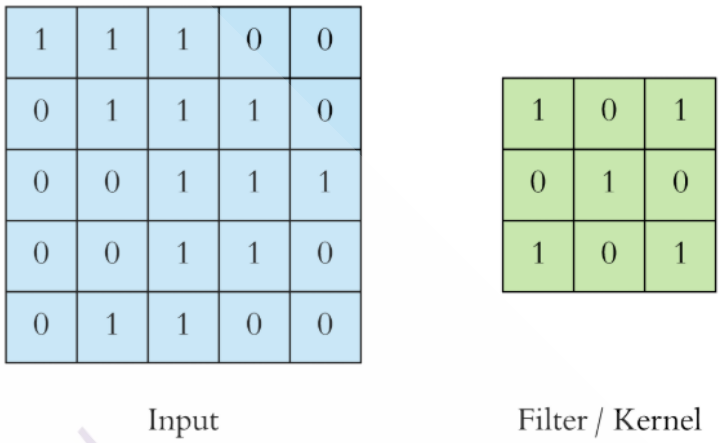



**ภาพที่ 2.6** Convolutional Neural Network

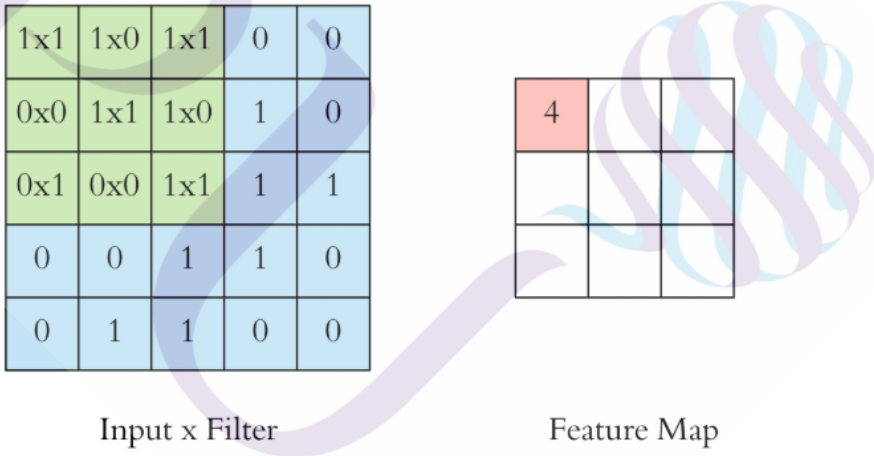
โครงสร้างของ Convolutional Neural Network ประกอบด้วยดังนี้

- Convolutional เป็น Layer หลักของ CNN ทำหน้าที่รับ Input เข้ามา แปลงภาพให้เป็น pixel ที่กำหนดให้ปีน 0 – 255 หลังจากนั้นใช้การดำเนินการทางคณิตาสตร์แปลงเป็นข้อมูลเพื่อหา Feature Extraction เพื่อนำมาคูณกับ ตัวกรอง (filter) หรือ เคอร์เนล (kernel) ที่ทำหน้าที่ดึงคุณลักษณะที่ใช้ในการรู้จำวัตถุออก โดยปกติตัวกรอง/เคอร์เนลอันหนึ่งจะดึงคุณลักษณะที่สนใจออกมาหรือที่เรียกว่า Feature Map ตัวอย่างการหา Feature Map หาได้ดังนี้ Feature Map[0,0] = (1*1) + (1*0) + (1*1) + (0*0) + (1*1) + (1*0) + (0*1) + (0*0) + (1*1) = 4 ส่วนการที่จะขยับตัวกรองหรือ Filter เป็นขนาดเท่าไหร่ จะถูกกำหนดด้วย Stride ซึ่งสามารถกำหนดค่าของ Stride ให้มากขึ้นก็ได้ ถ้าต้องการให้คำนวนหาคุณลักษณะมีพื้นที่ทับซ้อนกันน้อยขึ้น แต่อย่างไรก็ตามการกำหนดค่าของ

Stride ที่มากขึ้นจะทำให้เราคุณลักษณะ (Feature Map) ที่มีขนาดเล็กลง ดังภาพที่ 2.7 – 2.8

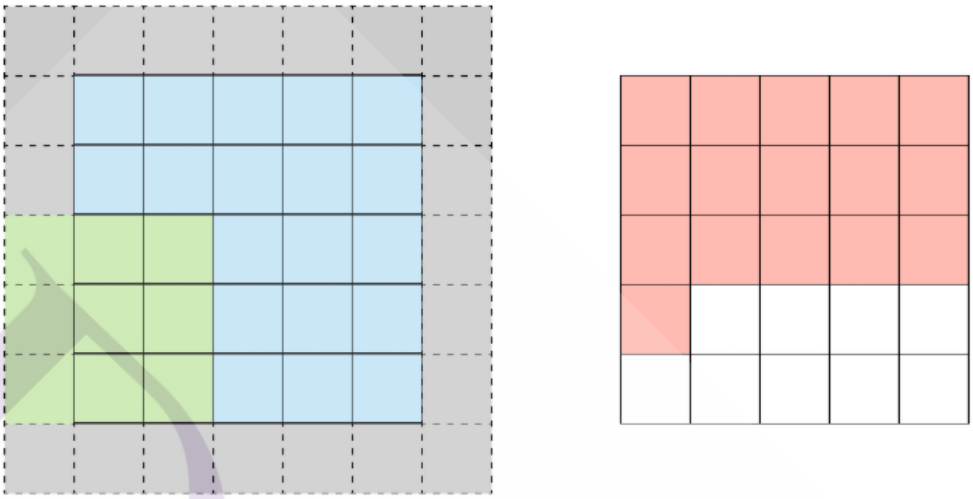



**ภาพที่ 2.7** Input Filter/Kernel




**ภาพที่ 2.8** Feature Map

- **Padding** เป็นการกำหนด Feature Map ให้มีขนาดเท่ากับ input ส่วนมากมักจะเติมค่า 0 (สีเทา) หรือค่าต่าง ๆ เข้าไป ส่วนเวลาเขียน code มักจะใช้ Padding = same ดังภาพที่ 2.9



**ภาพที่ 2.9** การทำ Padding

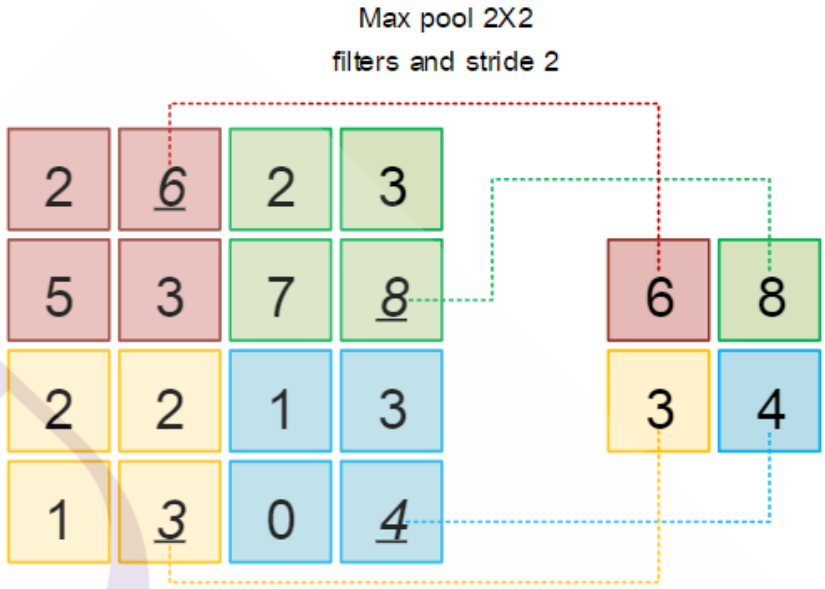
- **Pooling** เป็นกระบวนการที่ทำหน้าที่ลดขนาดของ Feature Map ที่ได้มาจากการทำ CNN มีวัตถุประสงค์ในการลดจำนวนของพารามิเตอร์ที่มีมากเกินไป และลดระยะเวลาในการ Train ข้อมูล Pooling นั้นมีหลายแบบดังนี้

  Max pooling

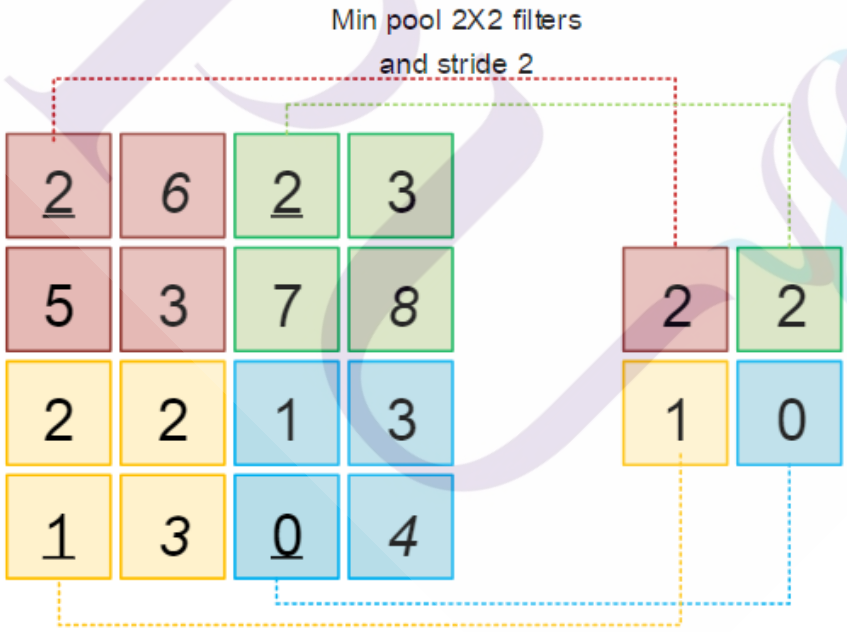
  Min polling

  Average polling

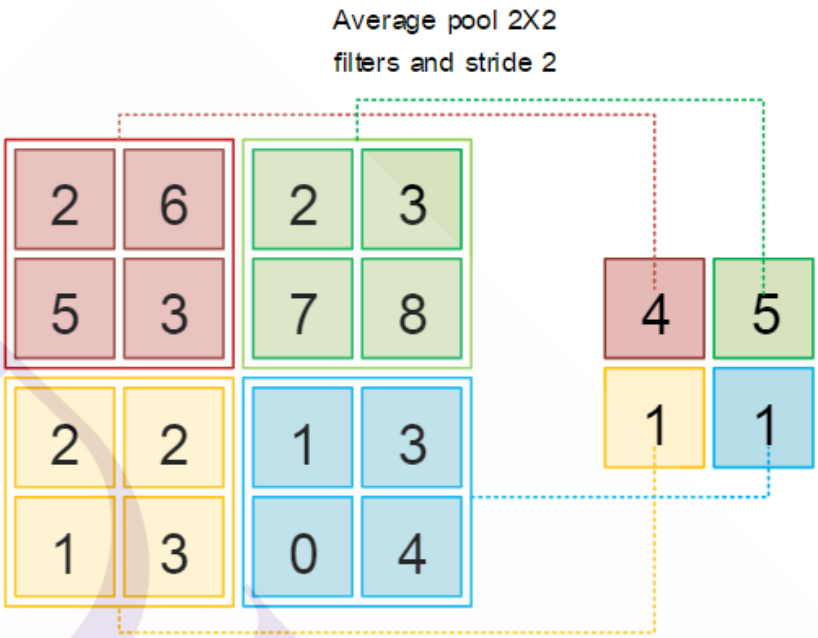
  แต่ที่เป็นที่นิยมใช้คือ Max pooling โดยหาค่ามากที่สุดของแต่ละ pixel ดังภาพที่ 2.10 – 2.12




**ภาพที่ 2.10** การทำ Max pooling

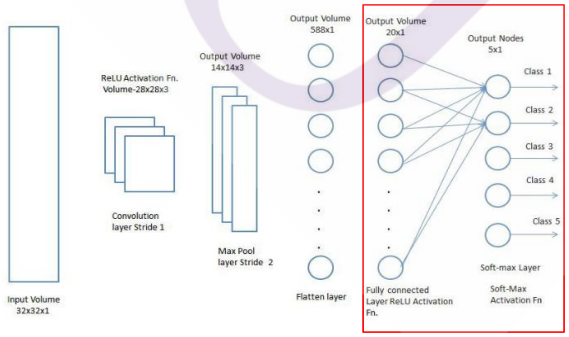



**ภาพที่ 2.11** การทำ Min pooling




**ภาพที่ 2.12** การทำ Average pooling

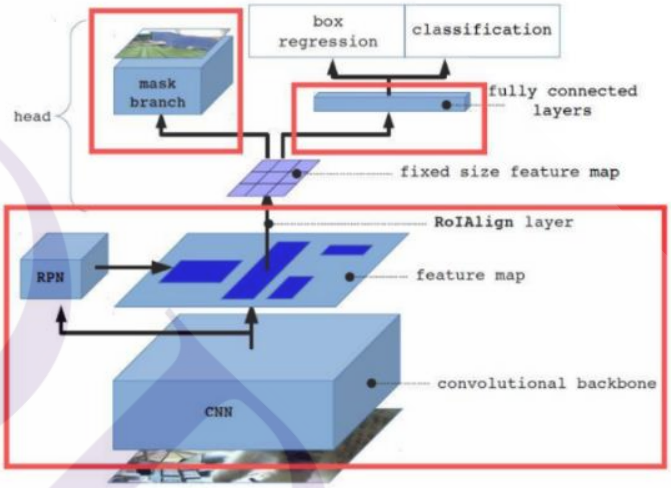
- Fully Connected เป็น layer สุดท้ายของการทำ CNN ส่วนนี้ จะทำหน้าที่นำเอาคุณลักษณะสำคัญไปสร้างเป็น Neural Network สำหรับการเรียนรู้ จดจำรูปแบบ และการทำนายประเภท โดยใช้เทคนิคที่ชื่อว่า Softmax เพื่อทำ Classification ต่อไป ดังภาพที่ 2.13



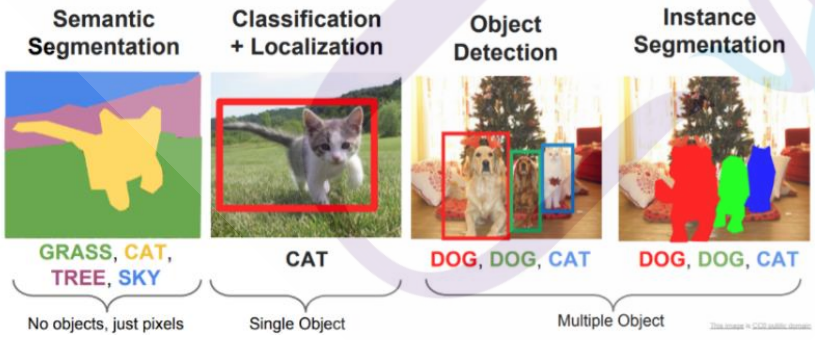

**ภาพที่ 2.13** การทำ Fully Connected

### 2.1.4 Mask R-CNN

Mask R-CNN ถูกเผยแพร่ในปี 2018 เป็นแนวคิดที่เรียบง่ายในการตรวจจับและแบ่งส่วนวัตถุ มีการพัฒนาต่อยอดมาจาก R-CNN, Faster R-CNN, Faster R-CNN ส่วนมากจะเป็นการทำ Object Detection ซึ่งมีข้อจำกัดหลายอย่าง ต่อมามีการพัฒนาเพื่อแก้ไขข้อจำกัด Mask R-CNN จะเป็น Instance Segmentation ที่เป็นแบบ multiple object ดังภาพที่ 2.14 - 2.15



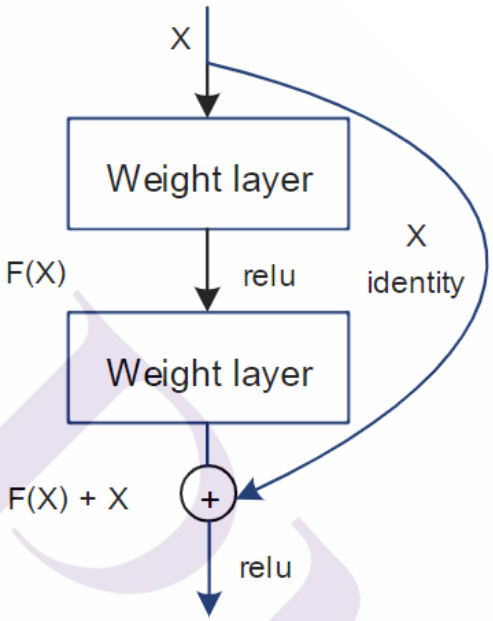

**ภาพที่ 2.14** Mask R CNN




**ภาพที่ 2.15** Instance Segmentation

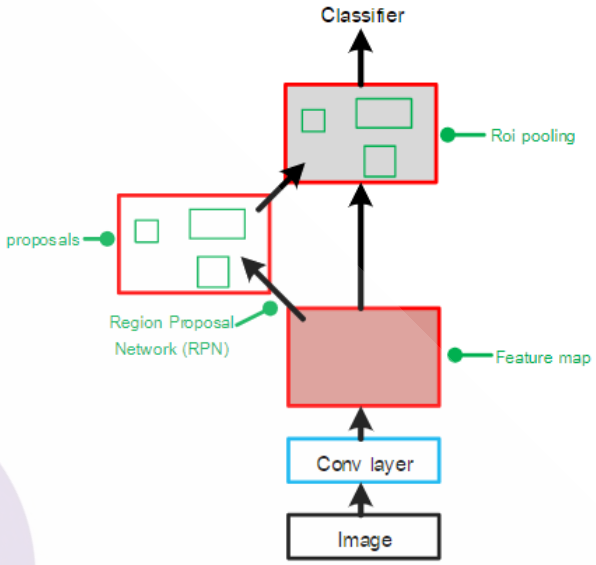
โครงสร้างของ Mask R CNN ประกอบไปด้วย

- Convolutional ResNet backbone: ResNet เป็น CNN ชนิดหนึ่งซึ่งเป็นวิธี การนำเอา residual modules มาต่อกัน และใช้ stochastic descent gradient มาเทรนตัว input จะถูก preprocess โดยการแบ่งเป็น patch เล็ก ๆ ก่อนถูกนำเข้ามาในโมเดล ในงานวิจัยนี้ใช้ ResNet101 ดังภาพที่ 2.16




**ภาพที่ 2.16** residual modules

- Region proposal Network (RPN) เป็น layer หนึ่งของ Mask R-CNN ทำหน้าที่สกัดคุณสมบัติที่คาดว่าจะเป็น วัตถุ ออกจาก feature map จากนั้นนำส่วนที่คาดว่าจะเป็นวัตถุเข้า CNN เพื่อคำนวณหา feature ที่สามารถใช้แยกแยะได้ว่าพื้นที่ใดเป็นวัตถุ โดยต้องทำหลังจาก Selective Search ดังภาพที่ 2.17




**ภาพที่ 2.17** Region proposal Network

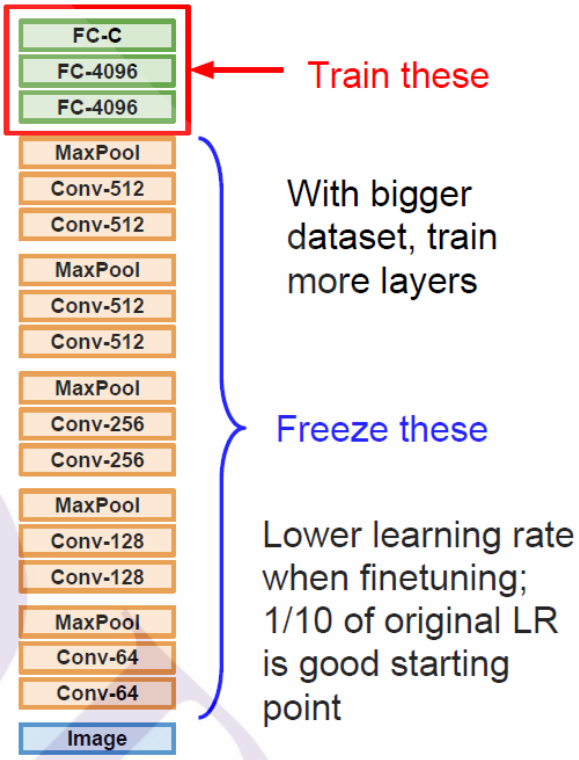
- ROI Pooling เป็นขั้นตอนการใช้พิกัดตำแหน่งบน feature map เพื่อหา feature vector ให้มีขนาดคงที่แล้วส่งไปยัง fully connected layer กับ softmax เพื่อทำ Classification ว่า proposed region feature นั้นจะตอบว่าเป็นคลาสอะไร

### 2.1.5 Transfer learning การเรียนรู้แบบถ่ายโอน

การเรียนรู้แบบถ่ายโอนเป็นเทคนิคการเรียนรู้ของเครื่อง ซึ่งใช้การเรียนรู้ที่มีอยู่แล้วมาเป็นโครงสร้างแล้วนำข้อมูลของใหม่มาสอน แล้วปรับตรงชั้น CNN ในส่วนของ Fully Connect ให้มีผลลัพธ์เป็นจำนวน Class ให้ตรงกับข้อมูล ซึ่งมีประโยชน์อย่างมากเพราะช่วยประหยัดเวลาในการสร้าง CNN และการทำ Optimization เป็น ส่วนโครงสร้างภายในจะเป็น CNN Architecture ที่เป็นแบบต่าง ๆ ดังนี้ Xception, VGG16[2], VGG19 และ ResNet, ResNetV2, ResNeX เป็นต้น

การสร้างการเรียนรู้แบบถ่ายโอนมี 2 แนวทางดังนี้ การพัฒนาการเรียนรู้ (Develop Model Approach) และการใช้วิธีการฝึกอบรมรุ่นก่อน (Pre-trained Model Approach)

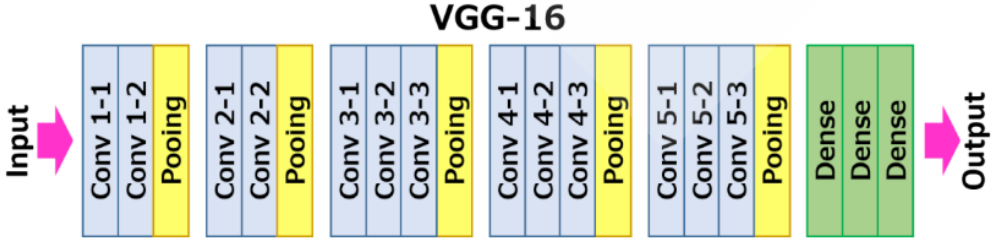
ข้อดีของ Transfer Learning ไม่ต้องเขียน CNN เอง ง่ายและรวดเร็วในส่วนของงานวิจัยฉบับนี้จะ Train ฉะเพราะชั้น FC เท่านั้น ดังภาพที่ 2.18




**ภาพที่ 2.18** Transfer Learning

ในงานวิจัยฉบับนี้ใช้ VGG16

VGG16 ปีนรูปแบบโครงข่ายประสาทเทียมที่เสนอโดย K. Simonyan และ A. Zisserman จาก University of Oxford แบบจำลองนี้มีความแม่นยำในการทดสอบสูงสุด 5 อันดับแรก 92.7% ใน ImageNet ซึ่งเป็นชุดข้อมูลของรูปภาพมากกว่า 14 ล้านภาพที่เป็นของคลาส 1,000 รายการ มันเป็นหนึ่งในโมเดลที่มีชื่อเสียงที่ส่งถึง ILSVRC-2014 มันทำให้การปรับปรุงให้ทำงานดีกว่า AlexNet โดยแทนที่ตัวกรองเคอร์เนลขนาดใหญ่ (11 และ 5 ในชั้นแรกและครั้งที่สองตามลำดับ) ด้วยตัวกรองขนาดเคอร์เนล 3 × 3 ดังภาพที่ 2.19 – 2.20



**ภาพที่ 2.19** โครงสร้าง VGG16

Summary of VGG16 Architecture

| | Layer | Feature Map | Size | Kernel Size | Stride | Activation |
|---|---|---|---|---|---|---|
| Input | Image | 1 | 224 x 224 x 3 | - | - | - |
| 1 | 2 X Convolution | 64 | 224 x 224 x 64 | 3x3 | 1 | relu |
| | Max Pooling | 64 | 112 x 112 x 64 | 3x3 | 2 | relu |
| 3 | 2 X Convolution | 128 | 112 x 112 x 128 | 3x3 | 1 | relu |
| | Max Pooling | 128 | 56 x 56 x 128 | 3x3 | 2 | relu |
| 5 | 2 X Convolution | 256 | 56 x 56 x 256 | 3x3 | 1 | relu |
| | Max Pooling | 256 | 28 x 28 x 256 | 3x3 | 2 | relu |
| 7 | 3 X Convolution | 512 | 28 x 28 x 512 | 3x3 | 1 | relu |
| | Max Pooling | 512 | 14 x 14 x 512 | 3x3 | 2 | relu |
| 10 | 3 X Convolution | 512 | 14 x 14 x 512 | 3x3 | 1 | relu |
| | Max Pooling | 512 | 7 x 7 x 512 | 3x3 | 2 | relu |
| 13 | FC | - | 25088 | - | - | relu |
| 14 | FC | - | 4096 | - | - | relu |
| 15 | FC | - | 4096 | - | - | relu |
| Output | FC | - | 1000 | - | - | Softmax |

**ภาพที่ 2.20** VGG16 Architecture

**เลเยอร์ที่หนึ่งและสอง**: อินพุตเป็นภาพ RGB 224x224x3 ซึ่งผ่านเลเยอร์ convolutional **ชั้นที่หนึ่ง** และที่**สอง**พร้อมกับ 64 feature map หรือ filter ที่มีขนาด 3 × 3 และการรวมขนาดเดียวกันโดยมี stride เท่ากับ 14 ขนาดของรูปภาพเปลี่ยนเป็น 224x224x64 จากนั้น VGG16 จะใช้เลเยอร์ร่วมสูงสุดหรือเลเยอร์การสุ่มตัวอย่างย่อยด้วยขนาดตัวกรอง 3 × 3 และก้าวหนึ่งของทั้งสอง ขนาดภาพที่ได้จะลดลงเป็น 112x112x64

**ชั้นที่สามและสี่**: ถัดไปมีสองเลเยอร์ convolutional พร้อมกับฟีเจอร์ 128 แผนที่ที่มีขนาด 3 × 3 และทำstride ที่ 1 จากนั้นจะมีเลเยอร์รวมสูงสุดอีกครั้งที่มีขนาดตัวกรอง 3 × 3 และทำ stride ที่ 2 เลเยอร์นี้เหมือนกับเลเยอร์รวมก่อนหน้านี้ยกเว้นมี feature map คุณสมบัติ 128 ชุดดังนั้นผลลัพธ์จะลดลงเหลือ 56x56x128

**ชั้นที่ห้าและหก**: เลเยอร์ที่ห้าและหกเป็นเลเยอร์ convolutional ที่มีขนาดตัวกรอง 3 × 3 และก้าวหนึ่ง ทั้งสองใช้แผนที่ฟีเจอร์ 256 รายการเลเยอร์ convolutional สองเลเยอร์จะตามด้วยเลเยอร์สูงสุดรวมกับขนาดตัวกรอง 3 × 3, ทำ stride ของ 2 และมี 256 feature map

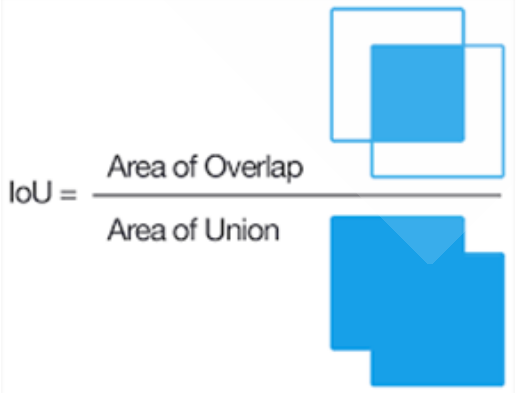
**ชั้นที่เจ็ดถึงชั้นสิบสอง:** ถัดไปคือชุด convolutional 3 เลเยอร์สองชุดตามด้วยชั้นการรวมสูงสุด เลเยอร์ convolutional ทั้งหมดมีตัวกรอง 512 ขนาด 3 × 3 และก้าวหนึ่ง ขนาดสุดท้ายจะลดลงเป็น 7x7x512

**ชั้นที่สิบสาม:** เอาท์พุทเลเยอร์ Convolutional จะแผ่ผ่านเลเยอร์ที่เชื่อมต่ออย่างเต็มที่ด้วย feature map 25088 แต่ละขนาด 1 × 1

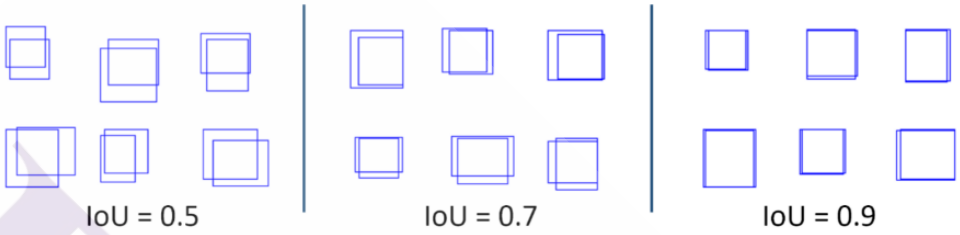
**เลเยอร์ที่สิบสี่และสิบห้า:** ถัดไปเป็นอีกสองเลเยอร์ที่เชื่อมต่ออย่างสมบูรณ์ด้วย 4096 หน่วย
**เลเยอร์เอาท์พุท:** ในที่สุดก็มีชั้นเอาท์พุท softmax กับ 1,000 ค่าที่เป็นไปได้

### 2.1.6 การวัดผล

- Intersection over Union (IoU) เป็นการวัดประสิทธิภาพของของโมเดลเป็นที่นิยมในการทำ Object detection ซึ่งหาได้จาก อัตราส่วนระหว่าง area ที่เป็น intersection ของ 2 bounding box หารด้วย area รวมของกรอบทั้งสอง หรืออาจจะเรียกว่า ดัชนี Jaccard เป็นหลักวิธีการในการหาจำนวนเปอร์เซ็นต์ที่ทับซ้อนกันระหว่างผลเฉลย (Ground Truth) และผลจากการทำนาย (Predict) ถ้าได้ค่า IoU ที่มากกว่า 0.5 ถึงว่ายอมรับได้ ดังสมการที่ 2.1 ตามภาพที่ 2.21 – 2.22



**ภาพที่ 2.21** area of IoU




**ภาพที่ 2.22** ตัวอย่างการซ้อนทับ

- AP และ mean Average Precision (mAP) เป็นการหาค่าเฉลี่ยของความแม่นยำ (Precision) และความลึก (Recall) ของวัตถุในรูปนั้นๆโดยโดยที่ ค่าความแม่นยำหาได้

  True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่าจริง และคนบอกว่ามันจริง

  True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง และคนบอกว่ามันไม่จริง

  False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่าจริง แต่คนบอกว่าไม่จริง

  False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่าไม่จริง แต่คนบอกว่าจริง

  Precision หาได้จากสมการที่ 2.2

  $$Precision = \frac{TP}{TP+FP} \quad (2\text{-}2)$$

  Recall หาได้จากสมการที่ 2.3

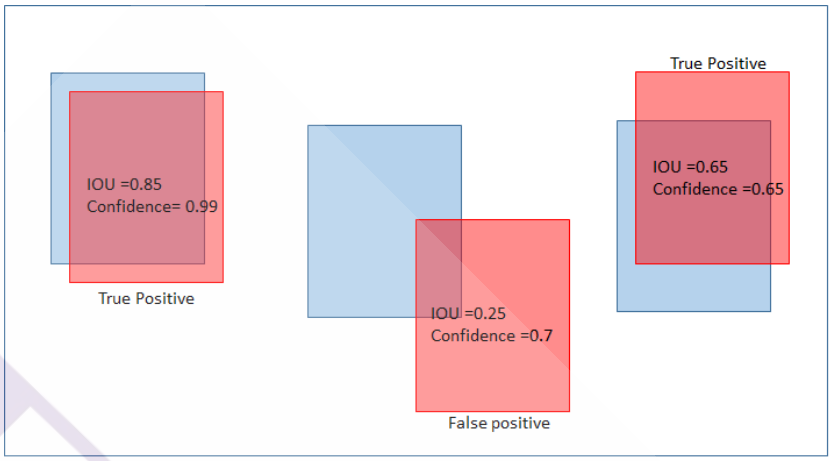
  $$Recall = \frac{TP}{TP+FN} \quad (2\text{-}3)$$

- การหา TP, TP+FP และ TP+FN ที่ IoU >0.5 ตามภาพที่ 2.23

  TP มีค่าเท่ากับจำนวน Class ที่มีค่า IoU >0.5

  TP+FP มีค่าเท่ากับจำนวน Class ทีทำนายได้ (Predict)

  TP+FN มีค่าเท่ากับจำนวนผลเฉลย (Ground Truth)




$$Precision = \frac{True\ Postive}{True\ Positive + False\ Negative} = \frac{count(True\ Postive)}{count(all\ red\ boxes)} = \frac{2}{3}$$

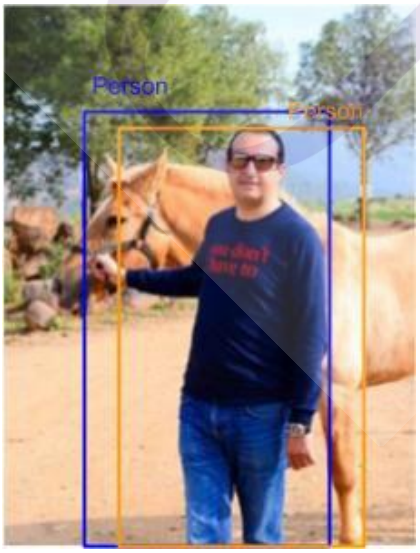
$$Recall = \frac{True\ Postive}{All\ Positives} = \frac{count(True\ Positives)}{count(all\ blue\ boxes)} = \frac{2}{3}$$

**ภาพที่ 2.23** การหา IoU

ดังภาพที่ 2.17 สามารถหาค่า AP ได้ดังนี้

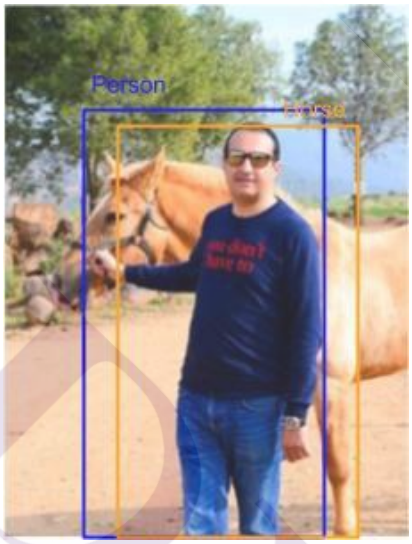
I. หาค่าการทำนายที่มี confidence แล้วเรียงจากมากไปหาน้อย

II. พิจารณาหาค่า IoU ถ้ามากกว่า 0.5 และมีการทำนาย Class ถูก ให้กำหนดค่าเป็น TP ดังภาพที่ 2.24



**ภาพที่ 2.24** IoU > 0.5 = TP

III. พิจารณาหาค่า IoU ถ้ามากกว่า 0.5 และมีการทำนาย Clsss ผิดให้กำหนดค่าเป็น FN ดังภาพที่ 2.25




**ภาพที่ 2.25** IoU > 0.5 and Wrong Class = FN

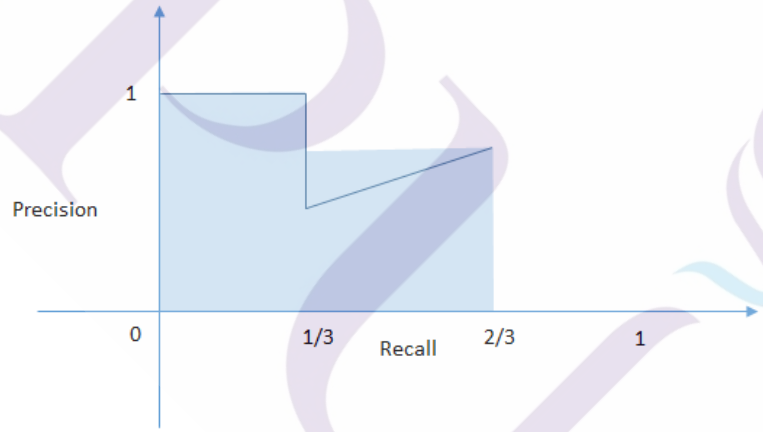
IV. พิจารณาหาค่า IoU ถ้าน้อยกกว่า 0.5 หรือ IoU = 0 ให้กำหนดค่าเป็น FP ดังภาพที่ 2.26

**ภาพที่ 2.26** IoU < 0.5 and Duplicated BB

สร้างตารางจากข้อมูลข้างต้น ตามตารางที่ 2.1 และ plot I. ดังภาพที่ 2.27

**ตารางที่ 2.1** ตารางตัวอย่างการหาค่า Precision Recall

| Confidence | IoU>0.5 | TP/FP | Precision | Recall |
|---|---|---|---|---|
| 0.99 | 0.85 | TP | 1/1 = 1 | 1/3 = 0.33 |
| 0.7 | 0.25 | FP | 1/2 = 0.5 | 1/3 = 0.33 |
| 0.65 | 0.65 | TP | 2/3 = 0.66 | 2/3 = 0.66 |

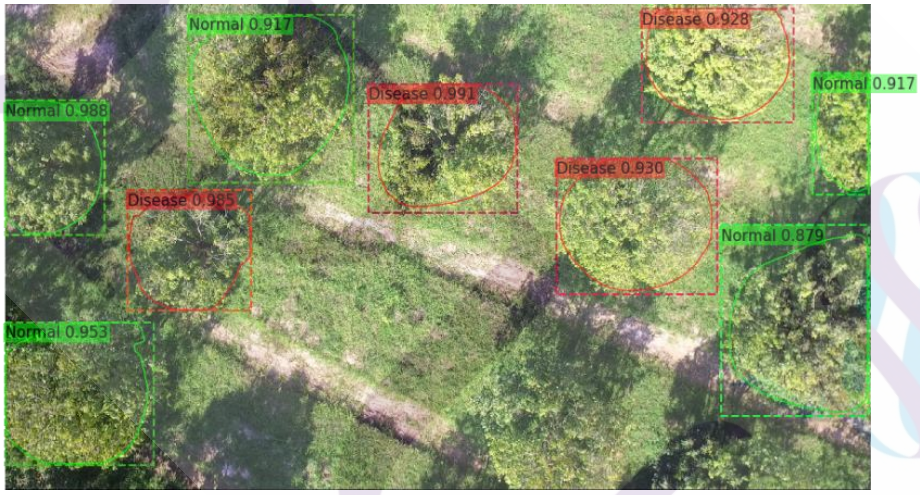



**ภาพที่ 2.27** การ pot Precision Recall curve

V. การหา Precision และ Recall ใช้วิธีการของ Confusion Matrix ซึ่งเป็นตาราง Matrix เพื่อประเมินผลลัพธ์การทำนายเพื่อเทียบกับผลลัพธ์จริง ๆ ตัวอย่างข้อมูลการทำนายที่แปลงเป็น Confusion Matrix ซึ่งมีข้อมูลดังนี้สามารถแปลงเป็นตารางเพื่อหาค่า TP, TN, FP และ FN ดังภาพที่ 2.28

| | Predicted class POSITIVE (spam ✉) | Predicted class NEGATIVE (normal 🕵) | |
|---|---|---|---|
| Actual class POSITIVE (spam ✉) | TRUE POSITIVE (TP) ✉✉ 320 | FALSE NEGATIVE (FN) ✉🕵 43 | $Recall = \frac{TP}{TP+FN} = \frac{320}{320+43} = 0.882$ |
| Actual class NEGATIVE (normal ✉) | FALSE POSITIVE (FP) ✉✉ 20 | TRUE NEGATIVE (TN) ✉🕵 538 | |
| | $Precision = \frac{TP}{TP+FP} = \frac{320}{320+20} = 0.941$ | | |

**ภาพที่ 2.28** ตัวอย่างตาราง Confusion Matrix




**ภาพที่ 2.29** ภาพจากการทำนาย

**ภาพที่ 2.30** ภาพจาก Ground Truth

**ภาพที่ 2.31** ภาพจาก IoU

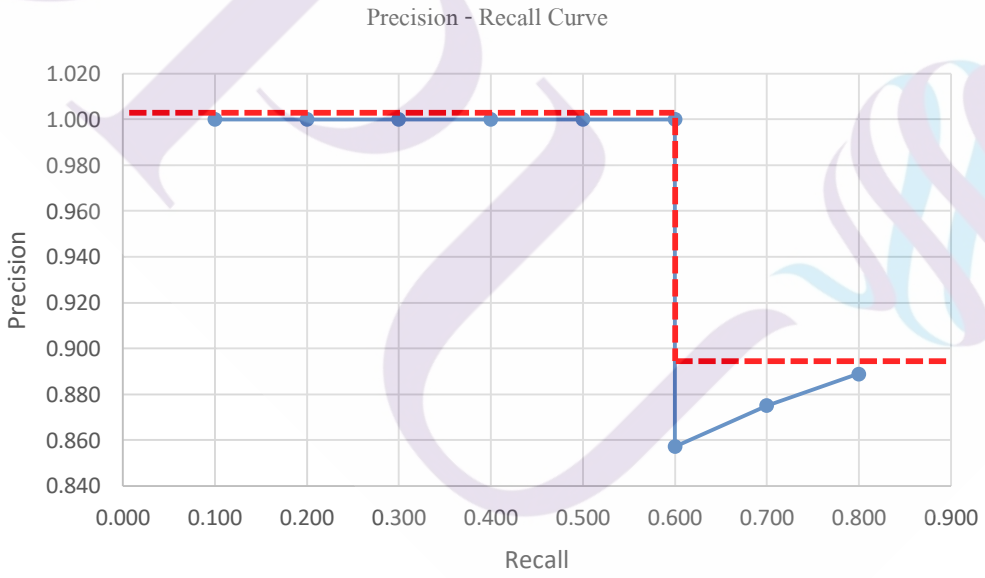
จากภาพ 2.29 , 2.30 และ 2.31 สามารถหา Precision , Recall ได้ดังนี้ โดยพิจารณาจากรูป หาค่า TP, TN, FP และ FN เป็นตัวอย่าง

**ตารางที่ 2.2** ตารางการหาค่า Precision Recall

| Detection | confidence | IoU | GT | TP/FP/FN | TP | FP | FN | Precision | Recall |
|---|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Disease | 0.991 | 0.884 | Disease | TP | 1 | | | 1.000 | 0.100 |
| Normal | 0.988 | 0.810 | Normal | TP | 2 | | | 1.000 | 0.200 |
| Disease | 0.985 | 0.883 | Disease | TP | 3 | | | 1.000 | 0.300 |
| Normal | 0.953 | 0.879 | Normal | TP | 4 | | | 1.000 | 0.400 |
| Disease | 0.930 | 0.801 | Disease | TP | 5 | | | 1.000 | 0.500 |
| Disease | 0.928 | 0.884 | Disease | TP | 6 | | | 1.000 | 0.600 |
| Normal | 0.917 | 0.813 | Disease | FN | 6 | | 1 | 0.857 | 0.600 |
| Normal | 0.971 | 0.656 | Normal | TP | 7 | | | 0.875 | 0.700 |
| Normal | 0.879 | 0.896 | Normal | TP | 8 | | | 0.889 | 0.800 |

จาก ตารางที่ 2.2 นำค่า Precision และ Recall มา plot Precision - Recall Curve เพื่อหาพื้นที่ใต้กราฟ (AUC) เมื่อได้พื้นที่ใต้กราฟก็จะหา Mean Average Precision (mAP) เพราะ mAP มีค่าเท่ากับ AUC = (1*0.70) + 0.89*(0.80-0.60) = 0.878 ดังภาพที่ 2.32




**ภาพที่ 2.32** Precision - Recall Curve

การหา Precision และ Recall แยก Class ใช้วิธีการของ Confusion Matrix ซึ่งเป็นตาราง Matrix เพื่อประเมินผลลัพธ์การทำนายเพื่อเทียบกับผลลัพธ์จริง ๆ ใช้ข้อมูลจากภาพ จากภาพ 2.29 , 2.30 และ 2.31

Reference คือสิ่งที่เป็นคำตอบจริง ๆ จาก Training Data

Predict คือสิ่งที่ Model ตอบให้ว่าคือคำตอบอะไร

Recall คือการส่งคำตอบจริง ๆไป Model ตอบถูกจริง ๆเท่าไหร่

Precision คือการคำนวณเทียบกับคำตอบที่ผิดข้างเคียง ว่าตัวเองถูกมากน้อยแค่ไหน

**สรุป ต้นเป็นโรค**

TP มีค่าเท่ากับจำนวน Class ที่มีค่า IoU >0.5 และ Class **เท่ากับ** 4

TP+FP มีค่าเท่ากับจำนวน Class ทีทำนายได้ (Predict) **เท่ากับ** 4

TP+FN มีค่าเท่ากับจำนวนผลเฉลย (Ground Truth) **เท่ากับ** 4 + 1

$$Precision = \frac{TP}{TP + FP} = \frac{4}{4 + 0} = 1$$

$$Recall = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.8$$

**สรุป ต้นเป็นไม่เป็นโรค**

TP มีค่าเท่ากับจำนวน Class ที่มีค่า IoU >0.5 และ Class **เท่ากับ** 4

TP+FP มีค่าเท่ากับจำนวน Class ทีทำนายได้ (Predict) **เท่ากับ** 5

TP+FN มีค่าเท่ากับจำนวนผลเฉลย (Ground Truth) **เท่ากับ** 4

TN ทายผิด **เท่ากับ 1**

$$Precision = \frac{TP}{TP + FP} = \frac{4}{4 + 1} = 0.8$$

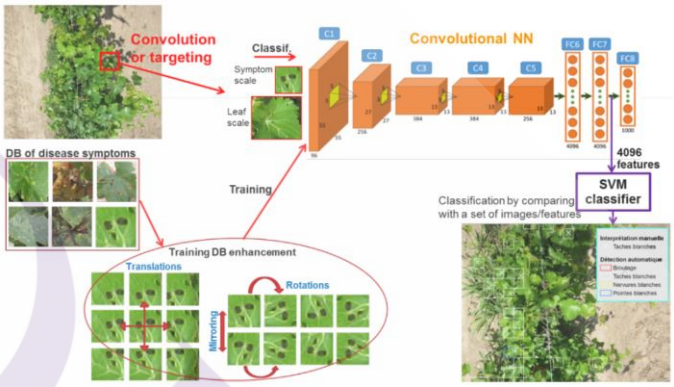
$$Recall = \frac{TP}{TP + FN} = \frac{4}{4 + 0} = 1$$

**2.2 งานวิจัยที่เกี่ยวข้อง**

มีงานวิจัยที่นำเสนอการใช้ Mask R CNN เพื่อแก้ไขปัญหา ต่าง ๆ รวมถึงการประชุกต์ใช้ CNN เพื่อตรวจหาพืชที่เป็นโรค และการวัดผล ที่เกี่ยวข้องแบ่งเป็นงานวิจัยดังนี้

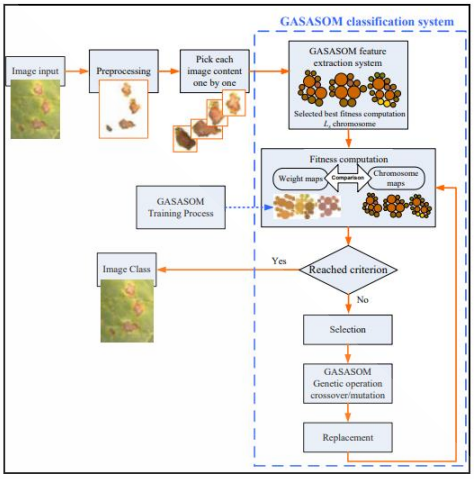
(Y. Bouroubi, et al ., P. Bugnet, et al ., T.Nguyen-Xuan, et al ., C.Gosselin, et al ., C. Bélec, et al ., L.Longchamps, et al ., P.Vigneault (2018)) งานวิจัยนี้ได้กล่าวถึงการหาศัตรูพืช

โดยใช้ UAV กับพืชที่ไม่สูงเช่น สตอเบอรี่ ใช้การเรียนรู้เชิงลึกประเภท CaffeNet ที่มีข้อมูล input เป็นภาพที่มีขนาด 24 x 24 pixels, 50 x 50 pixels, 100 x 100 pixels, 150 x 150 pixels, 200 x 200 pixels, 250 x 250 pixels and 500 x 500 pixels เป็นข้อมูลชุดสอนและมีการทำ Object detection ดังภาพที่ 2.33



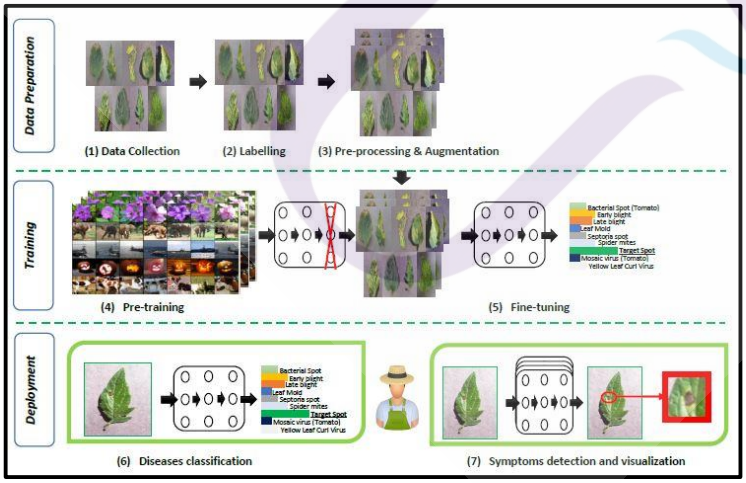

**ภาพที่ 2.33** งานวิจัยที่เกียวข้องที่ 1

(ชมพูทรัพย์ ปทุมสิน และ อาทิตย์ ศรีแก้ว (2018)) งานวิจัยได้กล่าวถึงได้กล่าวถึงการใช้ dataset จำนวน 122 รูป มีการทำ 2 ส่วน ส่วนแรกแยกองุ่นที่เป็นโรคออกจากองุ่นที่ไม่เป็นโรค แล้วแยกส่วนใบที่เป็นโรคออกจากพื้นหลัง และทำการจำแนกรูปแบบของจุดโรคบนใบว่าเป็นรูปแบบใดโดยใช้ อัลกอริทึม GA-SASOM มีข้อดี ใช้วิธีการประยุกต์ระหว่าง 2 วิธีในการ Classification และ มีวิธีการทำ 2 ส่วนทำให้เกิดความแม่นยำมากขึ้น มีข้อด้อย dataset น้อยไป และไม่มีการเปรียบเทียบกับ Model แบบอื่น ๆ และ ไม่ได้ทำ Augmentation process แยกได้แต่ไม่รู้ว่าใบไม้อยู่ตำแหน่งไหนของรูป ดังภาพที่ 2.34



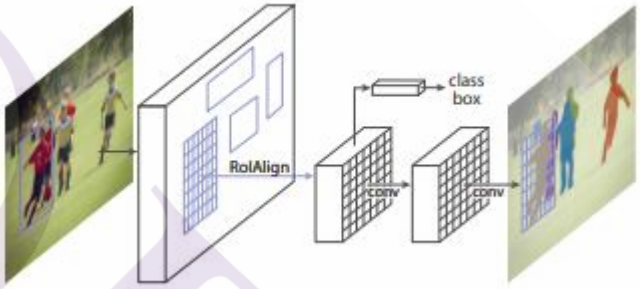

**ภาพที่ 2.34** งานวิจัยที่เกียวข้องที่ 2

(M. Brahimi, et al ., M. Arsenovic, et al ., S. Laraba, et al ., S. Sladojevic, et al ., K. Boukhalfa, et al ., A. Moussaoui, (2018)) งานวิจัยได้กล่าวถึงการใช้ CNN Architecture จำนวน 17 แบบ ในการตรวจจับใบไม้ที่เป็นโรคจำนวน 38 ชนิด โดยมีกระบวนการ 3 ขั้นตอน มีข้อดี ใช้ Dataset ที่หลากหลายใช้ Model ที่หลากหลายทำให้เกิดข้อเปรียบเทียบในการทำงานและทำ Augmentation process ทำ Object detection ได้มีข้อด้อยใช้เวลาในการทำงานนาน, ไม่มีเทคนิคใหม่ และไม่มีการเพิ่มประสิทธิภาพโมเดล ดังภาพที่ 2.35




**ภาพที่ 2.35** งานวิจัยที่เกียวข้องที่ 3

(Kaiming He, et al ., G. Gkioxari, et al ., P. Doll´ar, et al ., R. Girshick ( 2018)) งานวิจัยได้กล่าวถึงแนวคิดที่เรียบง่ายในการตรวจจับและแบ่งส่วนวัตถุที่พัฒนาต่อยอดมาจาก R-CNN, Faster R-CNN และ Faster R-CNN โดยมากจะเป็นการทำ Object Detection มีข้อจำกัดหลายอย่าง ต่อมามีการพัฒนามาเป็น Mask R-CNN ที่ได้รับการพิสูจน์และเป็นที่ยอมรับและใช้งานแพร่หลายเพราะเป็นวิธีที่ทำ Object Detection ได้ดีและทันสมัย ดังภาพที่ 2.36

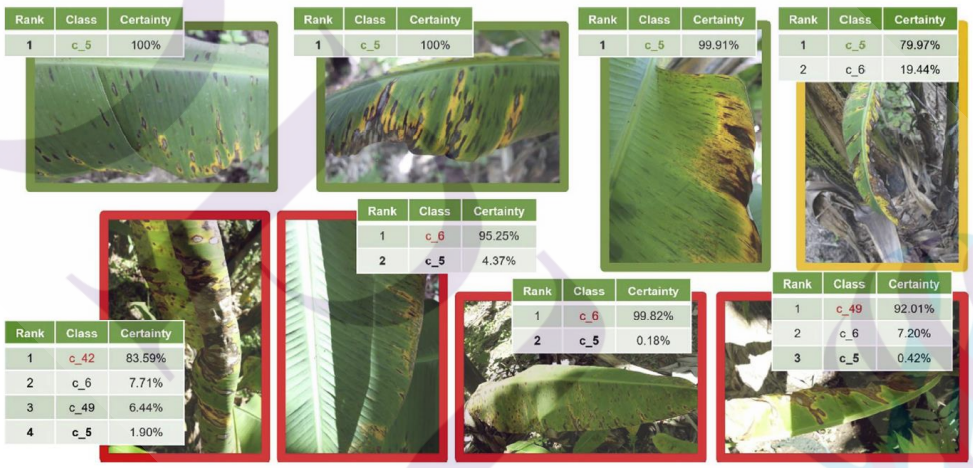



**ภาพที่ 2.36** งานวิจัยที่เกียวข้องที่ 4

(M. Danish (2018)) งานวิจัยได้กล่าวถึงการประยุกต์ใช้ Mask R-CNN เพื่อตรวจดูพฤติกรรมของ วัวเพศเมียฟาร์มปศุสัตว์ใช้เพิ่มผลผลิตโดยมีอินพุตเป็นภาพจาก กล้องวิดีโอวัดผลได้ AP@0.5 มีค่าเท่ากับ 0.632 ต้องปรับปรุง เช่น ค่าพารามิเตอร์, โมเดลการสอน และทำ Augmentation ดังภาพที่ 2.37

**ภาพที่ 2.37** งานวิจัยที่เกียวข้องที่ 5

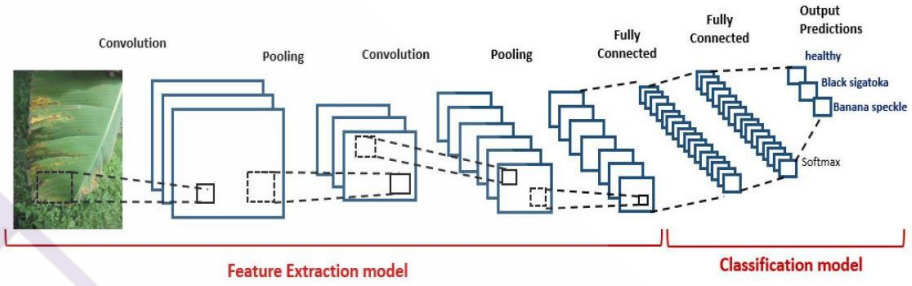
(Konstantinos P. Ferentinos (2018)) งานวิจัยได้กล่าวถึง การใช้ใบของพืชที่เป็นโรค เป็นชุดข้อมูลสอนและ ชุดข้อมูลทดสอบ จำนวน 57 กลุ่ม จำนวน 87,848 รูป แบ่งเป็น train 80% และ test 20% โดยใช้ CNNs Architecture จำนวน 5 model ดังนี้ AlexNet, AlexNetOWTBn, VGG, GoogLeNet, Overfeat ได้ผลการทดลองสรุปได้ว่า VGG ได้ผลดีที่สุด มีข้อดี Dataset มีปริมาณมากใช้ CNN Architecture จำนวน 5 model เปรียบเทียบกัน จำนวน 58 Class จำนวน 87,848 รูป แบ่งเป็น train 80% และ test 20% มีข้อด้อย ไม่มีการทำ Augmentation process ยังไม่สามารถทำ object detection เพียงแค่ทำ Classification เท่านั้น ดังภาพที่ 2.38




**ภาพที่ 2.38** งานวิจัยที่เกียวข้องที่ 6

(J. Amara, et al .,B. Bouaziz, et al ., A. Algergawy (2017)) งานวิจัยได้กล่าวถึง การใช้กับข้อมูลที่เป็นใบกล้วยที่เป็นโรคเช่น Black Sigatoka จำนวน 240 รูป, Banana Speckle จำนวน 1,817 รูป และ ใบที่ไม่เป็นโรค 1,643 รูป โดยใช้ CNNs ที่ชื่อว่า LeNet model โดยมีการแบ่งข้อมูลเป็น train:test 20%:80%, 40%:60%, 50%:50%, 60%:40%, 80%:20% และ แยกทำทั้ง ภาพสี และภาพขาวดำ มีข้อดี ใช้ Model ที่มีชื่อว่า LeNet model มีการแบ่งข้อมูลที่หลากหลาย train:test เป็น 20%:80%, 40%:60%, 50%:50%, 60%:40%, 80%:20% และ ทำทั้ง ภาพสี และ ภาพขาวดำ มีการบอกพารามิเตอร์การทำ optimizer (SGD) มีข้อด้อย มีข้อมูลไม่หลายกหลาย Dataset มีเฉพาะใบ

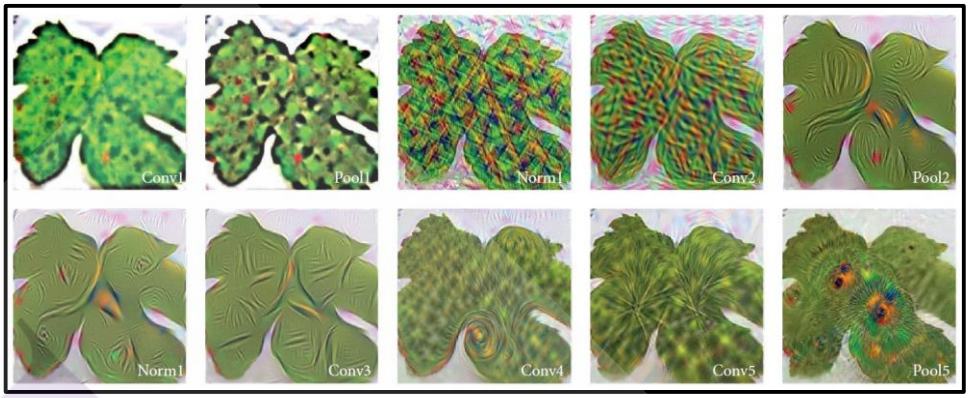
กล้วยอย่างเดียวเท่านั้น และไม่มีการทำ Augmentation process ถ้ามีใบไม้ 2 ใบในรูป แยกได้แต่ไม่รู้ว่าใบไม้อยู่ตำแหน่งไหนของรูป ดังภาพที่ 2.39




**ภาพที่ 2.39** งานวิจัยที่เกียวข้องที่ 7

(S. Sladojevic, et al ., M. Arsenovic, et al ., A. Anderla, et al ., D. Culibrk, et al ., D. Stefanovic (2016)) งานวิจัยได้กล่าวถึงการใช้ CNN ดังภาพที่ 2.40 โดยมีขั้นตอนดังนี้

- Dataset 15 Class เป็นใบพืชที่เป็นโรค จำนวน 4,483 รูป ใช้เป็นข้อมูลชุดทดสอบจำนวน 2,589 รูป และทำ augmented process ของรูปในแต่ละ Class
- ทำ Image Preprocessing and Labelling ปรับขนาดของรูปให้เป็น 256 X 256
- ทำ Augmentation Process
- Neural Network Training โดยใช้ Caffe Net
- Performed Tests โดยใช้ 10-fold cross validation technique
- Fine-Tuning
- ข้อดี Dataset มีปริมาณมาก มีการทำ Augmentation Process กระบวนการทำงานเป็นขั้นตอน แต่ละ step มี output ให้เห็น ข้อด้อย มี Class น้อย ไม่มีการเทียบผลลัพธ์กับ Model อื่น แยกได้แต่ไม่รู้ว่าใบไม้อยู่ตำแหน่งไหนของรูป



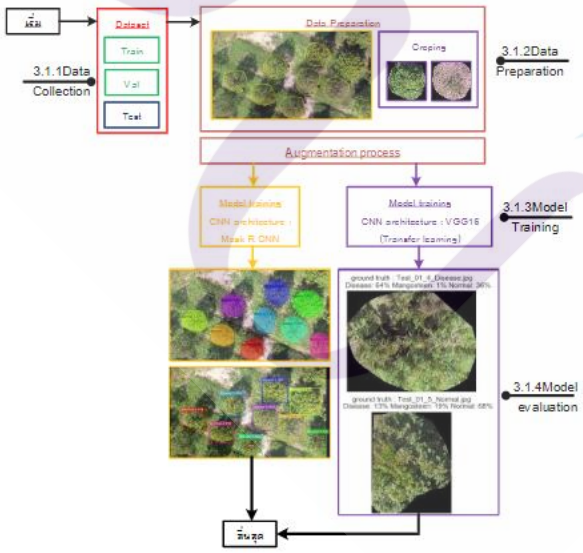

**ภาพที่ 2.40** งานวิจัยที่เกียวข้องที่ 8

# บทที่ 3
# ระเบียบวิธีวิจัย

การศึกษาวิจัยครั้งนี้ เป็นการวิจัยเพื่อหาวิธีการตรวจจับต้นไม้ที่เป็นโรคอัตโนมัติด้วยภาพถ่ายมุมสูงจากโดรนและวิธีการเรียนรู้เชิงลึก โดยเสนอโมเดลที่เป็นแบบลำดับขั้น (Cascade Model) โดยมีการใช้โครงข่ายประสาทเทียม แบบคอนโวลูชัน ทำงานร่วมกัน โมเดลแรกใช้ Mask R-CNN ทำหน้าที่ ตรวจจับตำแหน่งและหาต้นไม้ทุกต้นอยู่ในภาพ โมเดลที่สอง ทำหน้าที่ แยกแยะต้นไม้ว่าเป็นโรค หรือไม่ โดยใช้เทคนิคการโอนถ่ายความรู้(Transfer learning) ของโครงข่ายประสาทเทียม แบบคอนโวลูชันที่เรียกว่า VGG16 มีขั้นตอนดังนี้ การสร้างแบบจำลองโมเดล และการนำโมเดลไปใช้งาน

## 3.1 การสร้างแบบจำลองโมเดล

การสร้างแบบจำลองโมเดลในส่วนของ เป็น Mask R-CNN จะคอยทำหน้าตรวจจับตำแหน่งและหาต้นไม้ทุกต้นอยู่ในภาพ ในงานวิจัยนี้ใช้ Resnet 101 และใช้ชุดข้อมูลเป็นแบบ Mask ทุกต้น 100% ส่วนคู่ขนานจะ สร้างแบบจำลองโมเดลขึ้นมาอีก 1 ชุดโดยใช้ ชุดข้อมูล ที่ได้จาก Mask R-CNN ทำการ Transfer learning โดยใช้ VGG16 และมีการปรับฟังก์ชั่นสูญเสียเป็นแบบ focal loss จะทำหน้าที่แยกแยะต้นไม้ที่เป็นโรค และไม่เป็นโรคออกจากกัน



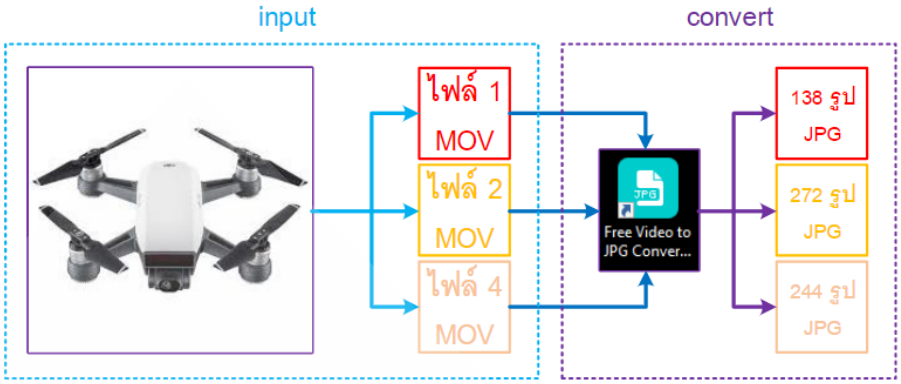

**ภาพที่ 3.1** แผนผังการสร้างแบบจำลอง (Trained Model) โดยรวม

**3.1.1 วิธีการเก็บรวบรวมข้อมูล (Data Collection)**

**3.1.1.1 กระบวนการเก็บข้อมูล** ข้อมูลอินพุตเป็นไฟล์วิดีโอที่มีการบันทึกจากโดรน จำนวน 4 มีคุณสมบัติดังนี้ เป็นไฟล์มีนามสกุล Mov ที่มีความละเอียดเป็น 2K (2704 X 1520) อัตราภาพต่อวินาที เป็น 29.97 Frame/Second มีขนาดด้านกว้าง 2704 pixel และความสูง 1520 pixel มีความยาวเฉลี่ย 3.12 นาทีที่มีขนาดเฉลี่ย 970MB ไฟล์ที่ 1 และ 2 เป็นการถ่ายจากบริเวณเดียวกันซึ่งจะมีต้นทุเรียนที่เป็นโรคและไม่เป็นโรคปะปนกัน, ไฟล์ที่ 3 จะถ่ายที่บริเวณหนึ่งซึ่งจะมีต้นมังคุดมากจึงไม่ใช้ไฟล์นี้เป็นข้อมูล, ไฟล์ที่ 4 เป็นการถ่ายบริเวณหนึ่งซึ่งจะมีต้นทุเรียนที่เป็นโรคและไม่เป็นโรคปะปนกัน และ **ตัวอย่างภาพไฟล์ที่ 2 – 4 อยู่ในภาคผนวก**

**ภาพที่ 3.2** ตัวอย่างไฟล์ที่ 1

**3.1.1.2 กระบวนการแปลงข้อมูล** มีการแปลงข้อมูลที่ได้รับมาเป็นไฟล์วิดีโอทำการแปลงเป็นไฟล์รูปภาพเพื่อใช้ในการ Train ข้อมูล เนื่องจากไฟล์มีขนาด 30 Frame/Second ใช้โปรแกรม Free Video to JPG Converter เลือกเมนู Every แล้วปรับไปที่ 30 Frame (ตามคุณสมบัติของไฟล์) เพื่อแปลงวิดีโอให้เป็นรูปที่มีนามสกุล jpg ที่มีขนาด 2028 X 1140, ไฟล์ที่ 1 จะได้รูปจำนวน 138 รูป, ไฟล์ที่ 2 จะได้รูปจำนวน 272 รูปและไฟล์ที่ 4 จะได้รูปจำนวน 244 รูป ดังภาพที่ 3.4




**ภาพที่ 3.3** แสดงการรวบรวมและแปลงข้อมูล

### 3.1.2 การเตรียมข้อมูล (Data Preparation)

**3.1.2.1 การทำความสะอาดข้อมูล** เนื่องจากรูปที่ได้จากกระบวนการแปลงข้อมูลมีส่วนที่ไม่เกี่ยวข้องมีกระบวนการจัดการรูปดังนี้รูปในส่วนที่เป็นสิ่งปลูกสร้าง, รูปในส่วนที่เป็นภาพซ้ำกันออก (1 ภาพ จะเหมือนกันอยู่ 4 ภาพ ตัดออกทุก 2 ภาพที่เหมือน), รูปในส่วนที่เป็นต้นลำไย และมังคุด ที่ไม่เกี่ยวข้องกับต้นทุเรียน, รูปในส่วนที่ไม่ชัด, รูปในส่วนที่ต้นไม้ไม่เต็มต้น ได้ผลลัพธ์ดังนี้ **ไฟล์ที่ 1 จะเหลือรูปจำนวน 50 รูป, ไฟล์ที่ 2 จะเหลือรูปจำนวน 86 รูป, ไฟล์ที่ 4 จะเหลือรูปจำนวน 164 รูป** ตัวอย่างภาพที่ 3.4 และ ภาพตัวอย่างที่ตัดทิ้งอยู่ในภาคผนวก

**3.1.2.2 การแบ่งข้อมูล** การ sampling ข้อมูลในการแบ่งรูปเพื่อใช้ Train, Validation และ Test จะใช้หลักการสุ่มตามหลักสถิติจากเอกสารของสำนักสถิติแห่งชาติ โดยใช้การเลือกแบบ **“การเลือกหน่วยตัวอย่างโดยใช้ความน่าจะเป็นเท่ากัน”** ซึ่งมี 2 วิธีดังนี้ 1. การเลือกกลุ่มตัวอย่างอย่างง่าย (simple random sampling : SRS) 2 วิธีการเลือกหน่วยตัวอย่างแบบมีระบบ (systematic sampling : SYS)

การบ่งข้อมูลจะใช้ การเลือกกลุ่มตัวอย่างอย่างง่าย(simple random sampling : SRS) โดยวิธีการใช้คอมพิวเตอร์ในการสุ่ม มีการแบ่งข้อมูลเป็นดังนี้

- มีการแบ่งข้อมูลเป็น Train, Validation และ Test โดยมีอัตราส่วนเป็น 60:30:10
- ไฟล์ที่ 1 Train:30 รูป, Validation:15 รูป และ Test:5 รูป
- ไฟล์ที่ 2 Train:52รูป, Validation:25รูป และ Test:9รูป
- ไฟล์ที่ 3 Train:98รูป, Validation:49รูป และ Test:17รูป
- นำไฟล์ที่ 1 – 3 ของ Train, Validation และ Test มารวมตามชนิดของข้อมูลจะได้ผลลัพธ์ดังนี้ Train:180รูป, Validation:89รูป และ Test:31รูป

**3.1.2.3 การ Mask ข้อมูล** แทนค่าต้นทุเรียนที่เป็นโรค แทนค่าเป็น 1 และ ต้นทุเรียนไม่เป็นโรค แทนค่าเป็น 2 โปรแกรม via-1.0.6 ทำ label รูปภาพ เนื่องด้วยรูปที่แปลงออกมาจากไฟล์วิดีโอ จะพบปัญหาดังนี้จำนวนของต้นทุเรียนที่เป็นโรคมีน้อยกว่าต้นที่ไม่เป็นโรค, บางรูปไม่มีต้นทุเรียนที่เป็นโรค, บางรูปมีต้นเป็นโรคที่อยู่ด้วยกัน ส่งผลทำให้ข้อมูลข้างต้นอาจเกิดไม่สมดุลกัน หรือ imbalance dataset จึงมีการแบ่งการ Mask ต้นทุเรียนที่เป็นโรคและไม่เป็นโรคเป็น 2 แนวทาง โดยมีจำนวนข้อมูลดังนี้**ข้อมูล Train:180รูป, Validation:89รูป และ Test:31รูป ดังรูปที่ 3.8 – 3.9** ดังนี้

**แนวทางที่ 1** สุ่ม Mask ต้นทุเรียนที่ไม่เป็นโรคให้เท่ากับต้นที่เรียนที่เป็นโรค โดยการสุ่มพิจารณาจากจำนวนต้นทุเรียนที่เป็นโรคในภาพนั้น ๆ แล้วสุ่ม Mask ต้นที่ไม่เป็นโรคถ้าในบางภาพมีต้นทุเรียนที่ไม่เป็นโรคมีอยู่ชนิดเดียวในภาพให้ดูจำนวนรวมของต้นไม้ที่เป็นโรคแล้ว Mask ต้นไม่ไปโรคให้เท่ากับจำนวนของต้นไม้ที่เป็นโรคในขณะนั้น แนวทางที่ 1 จะใช้กับข้อมูล Train ตามตารางที่ 3.2

**แนวทางที่ 2 Mask** ต้นทุเรียนที่ไม่เป็นโรคและต้นที่เรียนที่เป็นโรคในแต่ละภาพให้ครบทุกต้นหรือ ทำการ Mask 100% โดยไม่สนจำนวนต้นไม่ที่เป็นโรคและไม่เป็นโรคจะมีจำนวนเท่าไหร่ ตามตารางที่ 3.2

**ตารางที่ 3.1** แทนชื่อต้นทุเรียนโดยใช้ชื่อภาษาอังกฤษ

| ชื่อต้นไม้ | ชื่อ label | ชื่อในการทำนาย |
|---|---|---|
| ต้นทุเรียนเป็นโรค | 1 | Disease |
| ต้นทุเรียน | 2 | Normal |

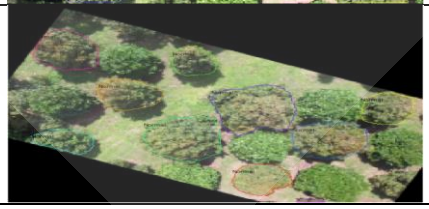
**ตารางที่ 3.2** จำนวนของ Training data, Validation data และ Testing data ที่ได้จากการ Mask

| แนวทางที่ | Training Data | | Validation Data | | Testing Data | |
|---|---|---|---|---|---|---|
| | Disease | Normal | Disease | Normal | Disease | Normal |
| 1 | 367 ต้น | 367 ต้น | 162 ต้น | 248 ต้น | 62 ต้น | 175 ต้น |
| 2 | 367 ต้น | 884 ต้น | | | | |

**ภาพที่ 3.4** ภาพตัวอย่างที่เป็นสวนลำไย

**3.1.2.4 การทำ Augmentation process** เป็นการขยายส่วนข้อมูลอินพุตภาพที่ได้รับเข้ามาเพื่อให้มีข้อมูลมีความหลากหลายมีการทำทั้งหมด 5 แบบ ตามตารางที่ 3.3

**ตารางที่ 3.3** ตัวอย่างการทำ Augmentation process ทั้ง 5 แบบ

| | |
|---|---|
| iaa.Fliplr(0.5) | |
| ContrastNormalization((0.75, 1.5) | |
| Affine |  |

| | | |
|---|---|---|
| Multiply | | |
| AdditiveGaussianNoise(loc=0,<br>scale=(0.0, 0.05*255),<br>per_channel=0.5) | | |

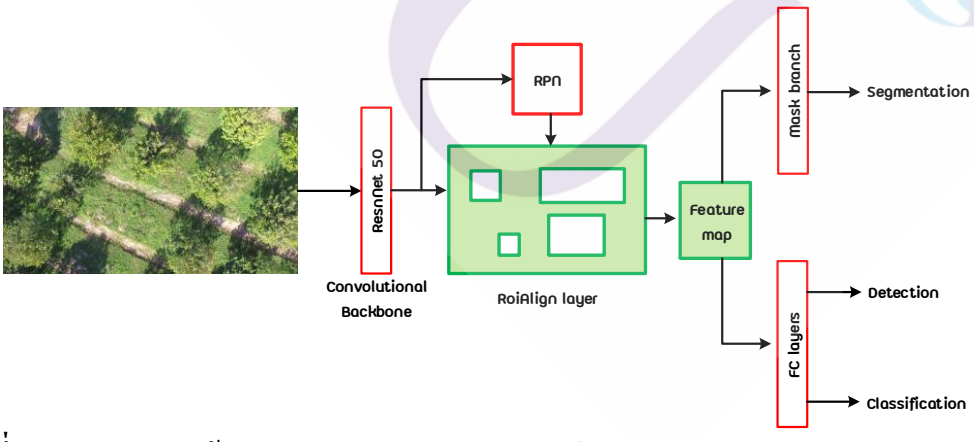
* ข้อสังเกต ชุดข้อมูล ที่ใช้เป็น imbalance data ในส่วนการ train ของ mask จะไม่ไม่มีการทำ Under sampling เพื่อปรับลดจำนวน dataset ในการแก้ปัญหา imbalance data เพราะ การที่มี ชุดข้อมูล ที่มาก เราจะหา feature ที่สำคัญของภาพได้มากพอ ถ้าตัดหรือ ลด จำนวนรูปอาจจะสูญเสีย feature ที่สำคัญไปดังนั้นจึงไม่แก้ไขปัญหา imbalance data

### 3.1.3 การสร้างแบบจำลอง (Model Training)

ประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึก (Deep learning) ซึ่งใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันโดยใช้เทคนิค Mask R-CNN และ VGG16 มีการ สร้างโมเดลเป็นแบบลำดับชั้นแบบลำดับ (Cascade model) โดยทำการฝึกสอนให้เป็นแบบคู่ขนาน

#### 3.1.3.1 Mask R-CNN

มีการสร้างโมเดล ฝึกสอน โดยกำหนดใช้ข้อมูลตามตารางที่ 3.2 ในแนวทางที่ 2 (Mask ต้นที่เป็นโรค และไม่เป็นโรคหมดทุกต้น) และใช้การตั้งค่าพารามิเตอร์ตามตาราง 3.4




**ภาพที่ 3.5** แผนผังการสร้างแบบจำลอง (Trained Model) ในส่วนของ Mask R-CNN

**ตารางที่ 3.4** การตั้งค่าพารามิเตอร์

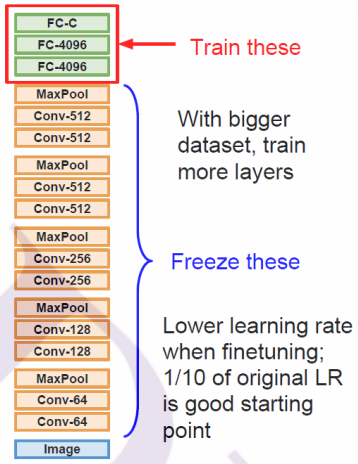
| Config | Description |
|---|---|
| ResNet Architecture | ResNet101 (All) |
| Learning rate momentum | 0.09 |
| Image min dimension | 1152 |
| Image max dimension | 1920 |
| Detection min confidence | 0.9 |
| Number of Epoch | 20 |
| Steps per epoch | 1251 |
| Validation steps | 410 |
| Mask shape | 28 X 28 |
| Num classes | 1+2(background + Disease + Normal) |

**3.1.3.2 VGG16 + Mask R-CNN**

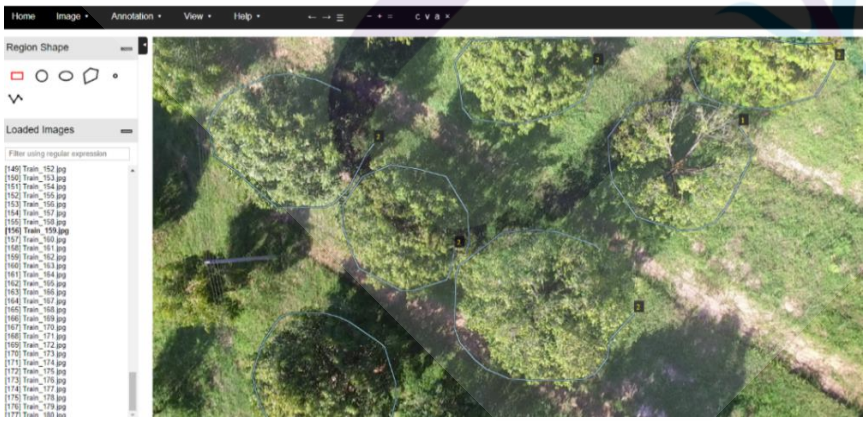
ใช้ขั้นตอนของการส่งต่อการเรียนรู้ หรือ Transfer learning ใช้ CNN architecture แบบ VGG16 ซึ่งเป็นที่นิยมในปัจจุบัน มีโครงสร้างการทำงานที่ไม่ซับซ้อน ไม่ต้องการใช้ประสิทธิภาพสูงมากนัก ซึ่งในงานวิจัยฉบับนี้ใช้ของมูลจาก 3.1.3.1 และเพิ่มจำนวน Class เพื่อขึ้นอีก 1 Class เนื่องจากเกษตรกรมักปลูกมังคุดกับทุเรียนติดกัน ดังนั้น มีจำนวนรูปภาพรวม 2931 รูป แบ่งเป็น Disease(591) : Mangosteen(853) : Normal(1,487) ดังรูปที่ 3.10 **โดยใช้ข้อมูลชุดสอน ของ Mask R-CNN การ Transfer learning ในภาพ 3.7 จะทำการ Train ใหม่เฉพาะส่วนในส่วน FC ชั้นสุดท้ายเป็นซึ่งชั้นอื่นไม่มีการแก้ไขตามสี่เหลี่ยม สีเขียว** มีขั้นตอนการสร้างแบบจำลองโมเดลดังนี้

**ตารางที่ 3.5** การตั้งค่าพารามิเตอร์ step 2 ส่วน Transfer Learning

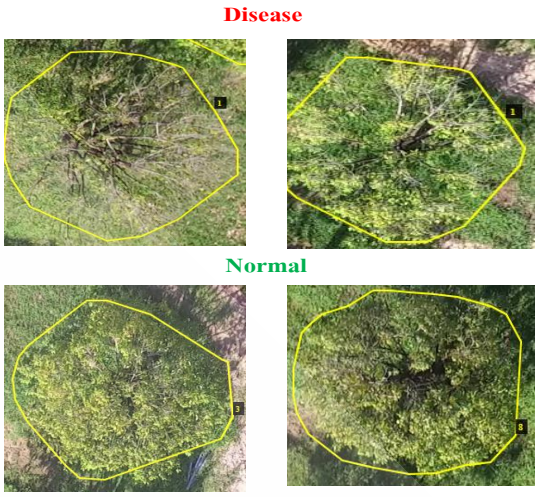
| Setting | Description |
|---|---|
| CNN Architecture | VGG16 |
| Dataset | 2931 รูป |
| Batch_size | 8 |
| loss | focal loss |
| optimizers | Adamax |
| epochs | 30 |



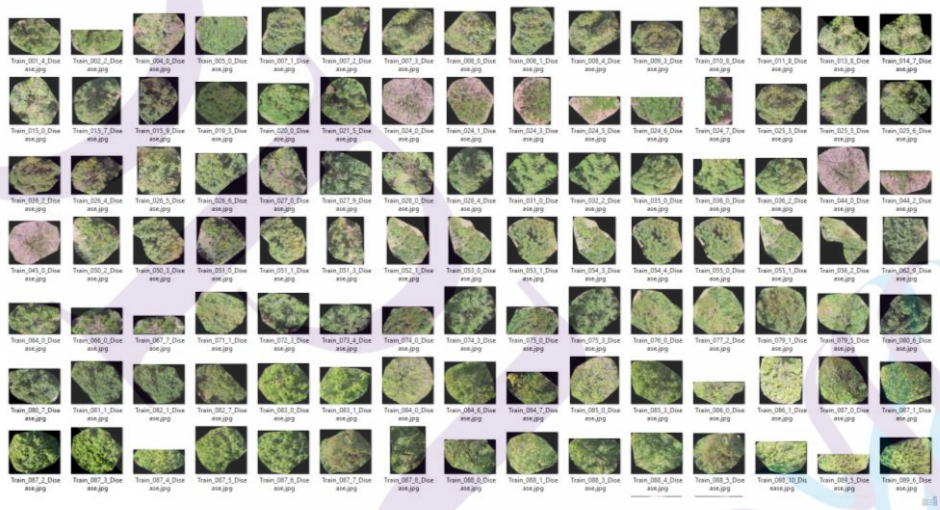

**ภาพที่ 3.6** แผนผังการสร้างแบบจำลอง (Trained Model) ในส่วนของ VGG16




**ภาพที่ 3.7** ตัวอย่างการ Mask ที่ใช้โปรแกรม via-1.0.6




**ภาพที่ 3.8** ตัวอย่างการ Mask ต้นทุเรียนทีเป็นโรค และไม่เป็นโรค



**ภาพที่ 3.9** ภาพที่ได้จาก Ground Truth

### 3.1.4 การประเมินผล (Model Evaluation)

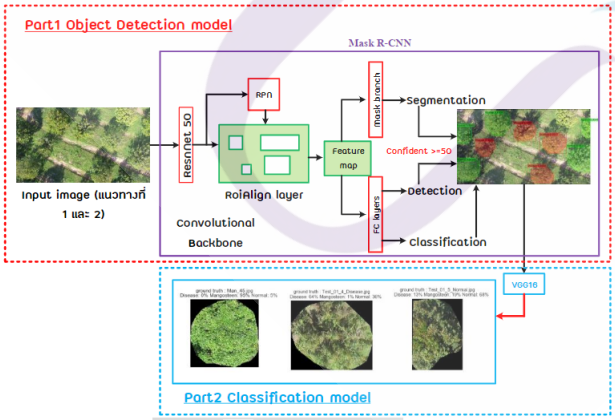
การวัดผลจะอ้างอิงจากบทที่ 2 หัวข้อ 2.1.6 การหาค่า Intersection over Union (IoU), การหาค่าเฉลี่ยของความแม่นยำ (Precision), การหาค่าความลึก (Recall) ของข้อมูลชุด Test ที่มีจำนวนรูป 31 รูป มี Disease (ต้นไม้เป็นโรค) จำนวน 70 รูป Normal (ต้นไม้ไม่เป็นโรค) จำนวน

164 รูป ที่ทำ Ground Truthไว้ แต่เนื่องจากในภาพนั้นมีทั้งต้นที่เป็นโรคและไม่เป็นโรคอยู่การวัดผลจะแยก Class และ รวม Class

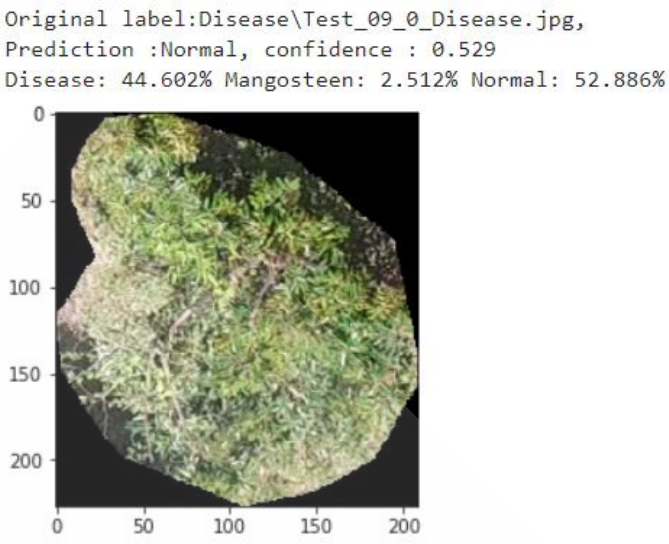
**ภาพที่ 3.10** ภาพจาก IoU Mask R-CNN

### 3.2 การนำโมเดลไปใช้งาน

ในหัวข้อนี้การทำงานจะเริ่มจาก นำภาพเข้า Mask R – CNN เพื่อทำการค้นหาและระบุตำแหน่งข้องต้นไม้ต้องปรับแต่งค่า min confidence ให้สามารถค้นหาต้นไม้ให้ครบทุกต้นจากในรูปแล้วส่งรูปภาพต้นไม้เข้าสู่โมเดลที่สองเพื่อทำการแยกแยะว่าต้นไม้ต้นไหนที่เป็นโรค, ไม่เป็นโรค หรือ มังคุด ดังภาพที่ 3.12




**ภาพที่ 3.11** แผนผังการนำแบบจำลองไปใช้งาน (Testing)



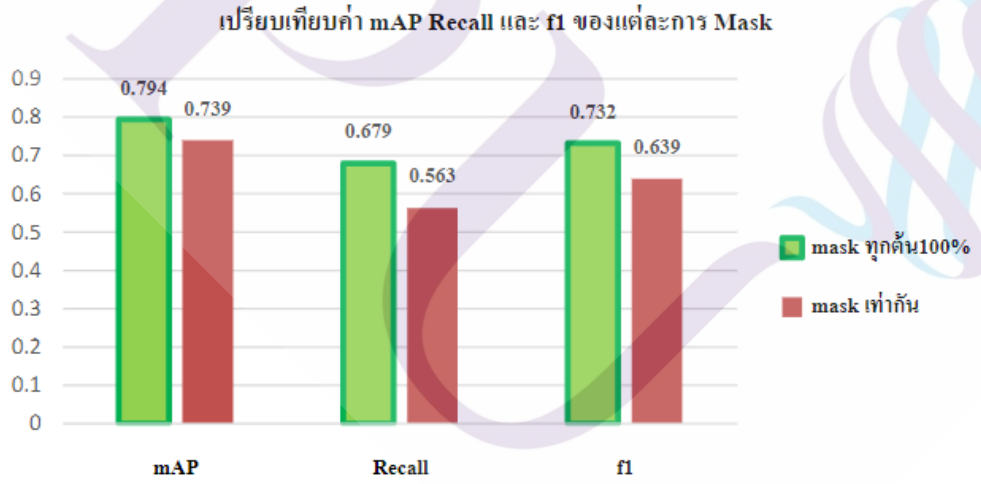

**ภาพที่ 3.12** ผลลัพธ์จากโมเดล 2 ที่แยกแยะว่าเป็นต้นไม่เป็นโรค

# บทที่ 4
# ผลการศึกษา

จากการวิจัยการตรวจหาต้นไม้เป็นโรคโดยอัตโนมัติด้วยภาพถ่ายมุมสูงจากโดรน และวิธีการเรียนรู้เชิงลึก สามารถแบ่งการวิเคราะห์ ผลการทดลอง ดังนี้

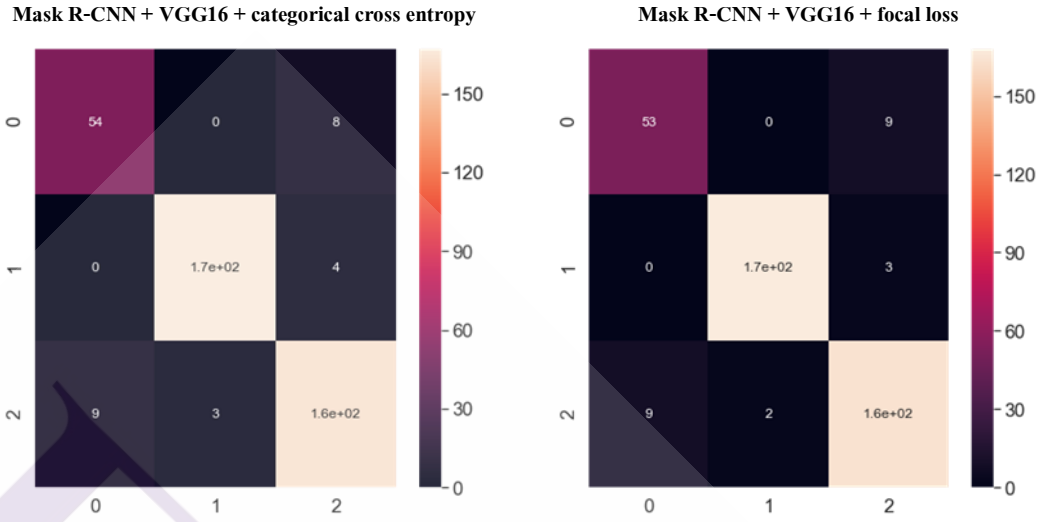
**4.1 ผลการปรับแต่งโมเดล** แบบลำดับขั้น ตามหัวข้อ 3.1.3 การสร้างแบบจำลอง (Model Training) ซึ่งได้ผลการทดลองดังนี้

**4.1.1 ผลการปรับแต่ง Mask R-CNN** ผลจากการปรับแต่งเป็นการเปรียบเทียบชุดข้อมูลในที่ใช้สอนระบบจากข้อ 3.1 ผลการทดลองสรุปว่า **mask ใน แนวทางที่ 2 Mask 100% ทุกต้น** ให้ค่า mAP = 0.749, Recall = 0.679 และ F1 = 0.732 มากที่สุด เป็นเพราะการ mask ในแนวทางที่ 2 ซึ่ง mask หมดทุกต้นเป็นการได้ข้อมูล Input หลากหลายทำให้ส่วนที่เป็น Convolution ซึ่งเป็นหัวใจของการสร้างโมเดลสามารถสกัดคุณลักษณะที่สำคัญออกมาได้ทั้งหมดสารมารถตรวจจับต้นทุเรียนที่เป็นโรคและไม่เป็นโรคได้ ถึงแม้ชุดข้อมูลจะเป็น imbalanced data ซึ่งดีกว่าการ mask ในแนวทางที่ 1 ซึ่ง Mask เท่ากัน ดังภาพที่ 4.1




**ภาพที่ 4.1** แสดงผลของ mAP, Recall และ f1 ของการ mask ทุกต้นในรูป และ mask เท่ากัน

**4.1.2 ผลการปรับแต่ง Mask R-CNN + VGG16** เปรียบเทียบโมเดลที่ได้จาก Train ในหัวข้อ 3.1 เป็นแบบ VGG16 ซึ่งใช้ loss function ที่เป็น focal loss เปรียบเทียบกับ VGG16 ที่ใช้ loss function ที่เป็น categorical cross entropy ดังภาพที่ 4.2 และ 4.3




**ภาพที่ 4.2** เปรียบเทียบผลลัพธ์ Confusion matrix ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss (0 เป็น ต้นไม้ที่เป็นโรค, 1 เป็น ต้นมังคุด, 2 เป็น ต้นไม้ไม่เป็นโรค)

**Mask R-CNN+VGG16+categorical cross entropy**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Disease | 0.86 | 0.87 | 0.86 | 62 |
| Mangosteen | 0.98 | 0.98 | 0.98 | 171 |
| Normal | 0.93 | 0.93 | 0.93 | 175 |
| micro avg | 0.94 | 0.94 | 0.94 | 408 |
| macro avg | 0.92 | 0.93 | 0.92 | 408 |
| weighted avg | 0.94 | 0.94 | 0.94 | 408 |

**Mask R - CNN + VGG16 + focal loss**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Disease | 0.85 | 0.85 | 0.85 | 62 |
| Mangosteen | 0.99 | 0.98 | 0.99 | 171 |
| Normal | 0.93 | 0.94 | 0.93 | 175 |
| micro avg | 0.94 | 0.94 | 0.94 | 408 |
| macro avg | 0.92 | 0.92 | 0.92 | 408 |
| weighted avg | 0.94 | 0.94 | 0.94 | 408 |

**ภาพที่ 4.3** เปรียบเทียบผลลัพธ์ Precision, Recall และ F1-Score โมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss

จากผลการทดลองโดยใช้โมเดล **Mask R - CNN + VGG16 + categorical cross entropy** สรุปได้จาก data set ที่เป็นข้อมูลทดสอบสรุปได้ดังนี้

ต้นที่เป็นโรคมีจำนวน 64 ต้น ทายผิดจำนวน **8** ต้น

ต้นมังคุดมีจำนวน 171 ต้น ทายผิดจำนวน **4** ต้น

ต้นไม่เป็นโรคมีจำนวน 175 ต้น ทายผิดจำนวน **11** ต้น

รวมทั้งหมด 408 ต้น ทายผิด **24** ต้น

เมื่อนับจำนวนที่ทายผิดแล้วแยกประเภทต้นไม้ที่ทำนายผิด เป็น ต้นทุเรียนที่ไม่เป็นโรค **12**, ต้นทุเรียนที่เป็นโรค **9** ต้นและ ต้นมังคุด **3** ต้น ที่เป็นแบบนี้เพราะข้อมูลในการ Train มีต้นไม้ที่ไม่เป็นโรคมากที่สุดเกิดข้อมูลไม่สมดุลย์กัน

จากผลการทดลองโดยใช้โมเดล Mask R - CNN +VGG16 + focal loss สรุปได้ จาก data set ที่เป็นข้อมูลทดสอบสรุปได้ดังนี้

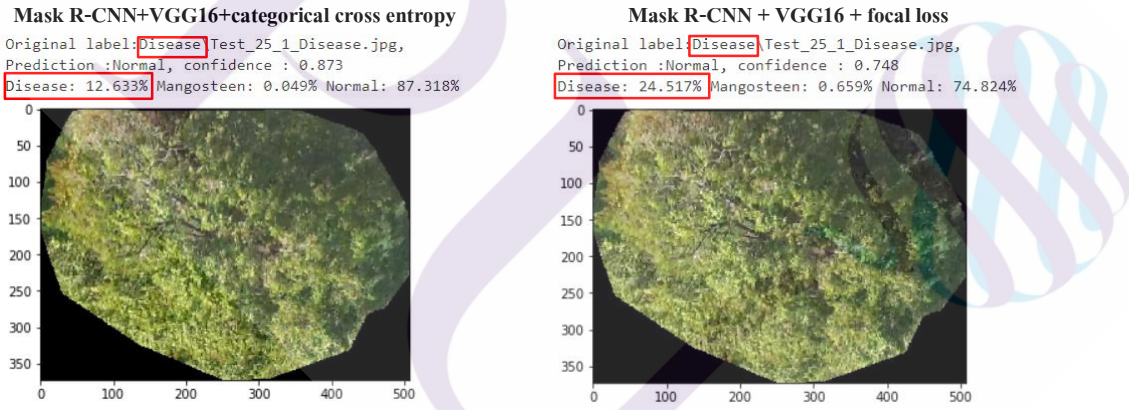
ต้นที่เป็นโรคมีจำนวน 64 ต้น ทายผิดจำนวน **9** ต้น

ต้นมังคุดมีจำนวน 171 ต้น ทายผิดจำนวน **3** ต้น

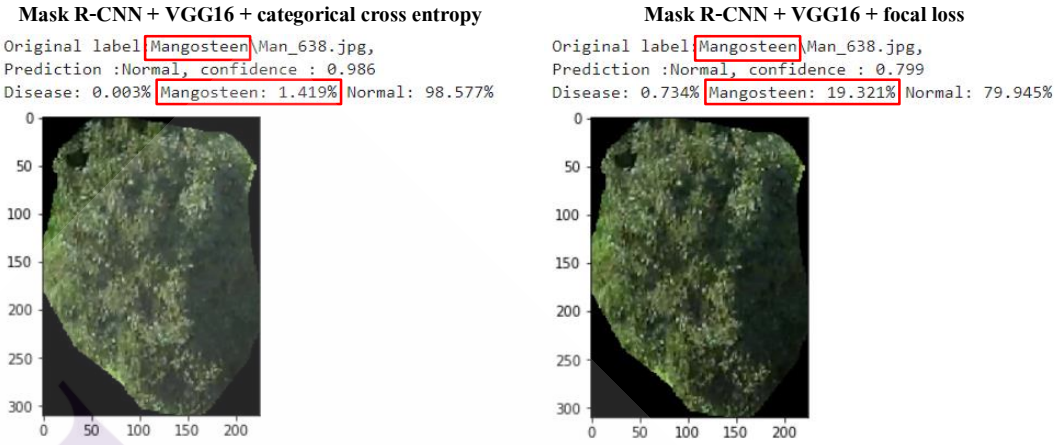
ต้นไม่เป็นโรคมีจำนวน 175 ต้น ทายผิดจำนวน **11** ต้น

รวมทั้งหมด 408 ต้น ทายผิด **23** ต้น

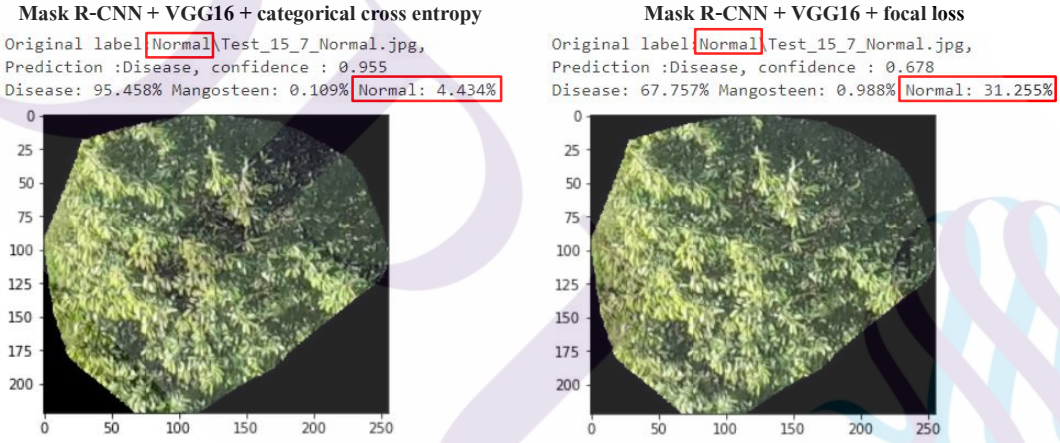
เมื่อนับจำนวนที่ทายผิดแล้วแยกประเภทต้นไม้ที่ทำนายผิด เป็น ต้นทุเรียนที่ไม่เป็นโรค 11, ต้นทุเรียนที่เป็นโรค 9 ต้นและ ต้นมังคุด 2 ต้น จากผลการผลการทดลองพบว่า focal loss สามารถแก้ไขปัญหา Imbalanced data ได้ในระดับหนึ่งที่ไม่ต้องทำ Under sampling และ over sampling ดังรูปที่ 4.4 – 4.7



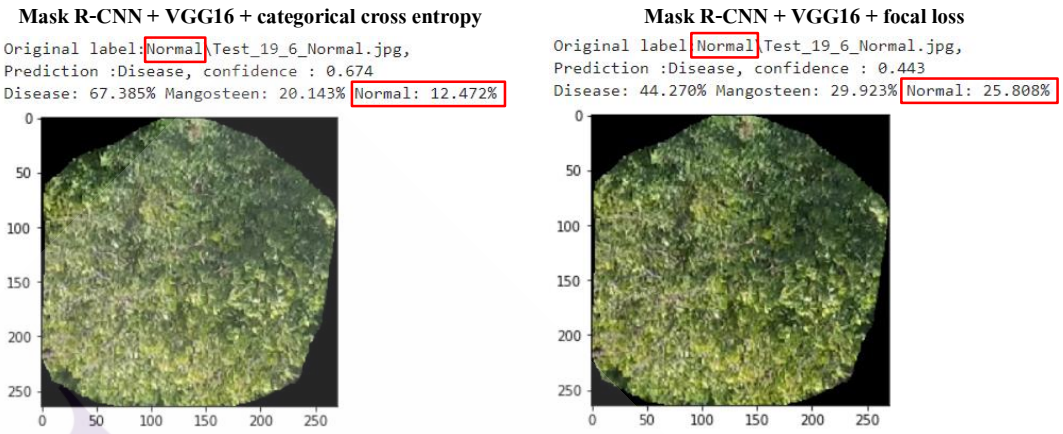

**ภาพที่ 4.4** ตัวอย่างการแยกแยะต้นที่เป็นโรค ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss




**ภาพที่ 4.5** เปรียบเทียบผลลัพธ์ ตัวอย่างการแยกแยะต้นมังคุด ของ ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss



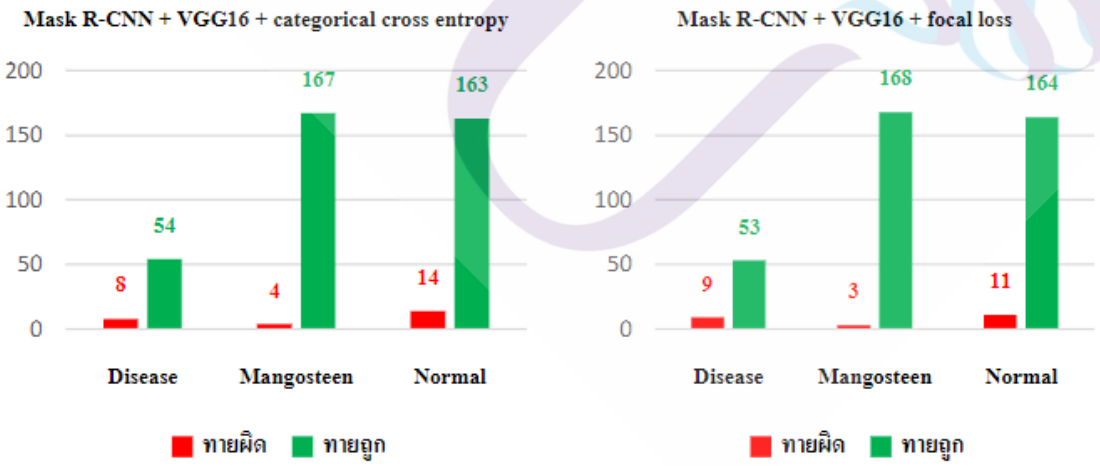

**ภาพที่ 4.6** ตัวอย่างการแยกแยะต้นที่ไม่เป็นโรค ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss มีค่ามากกว่า[1]




**ภาพที่ 4.7** ตัวอย่างการแยกแยะต้นที่ไม่เป็นโรค ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss มีค่ามากกว่า[2]

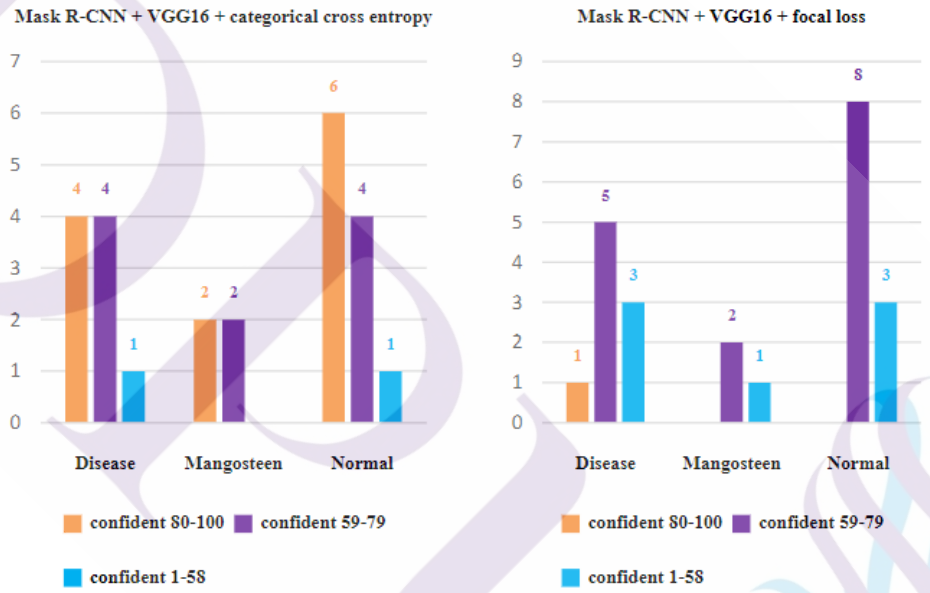
เมื่อวิเคราะห์จากภาพ 4.7 ทำให้ทราบว่าเมื่อมีการทำนายผิดเช่น รูปเป็นต้นไม่เป็นโรค Normal **Categorical cross entropy** ให้ค่า Probability ทำนาย Normal = 12.427%, Disease = 67.385%, Mangosteen = 20.143%

**focal loss** ให้ค่า Probability ทำนาย Normal = 25.808%, Disease = 44.27%, Mangosteen = 29.923% เมื่อเปรียบเทียบกันแล้วถึงแม้จะทำนายผิด ถ้าใช้ focal loss ค่า Probability สูงขึ้นเทียบกับการตอบถูกเพิ่มขึ้น ดังภาพที่ 4.8

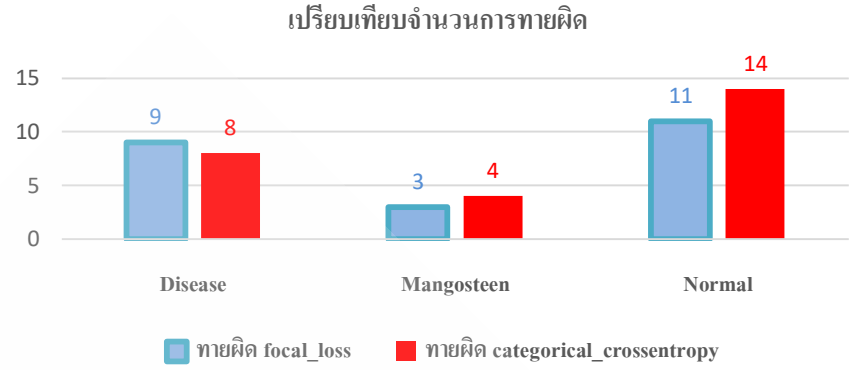



**ภาพที่ 4.8** เปรียบเทียบผลลัพธ์ จำนวนการทายผิด - ถูก ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss

เมื่อพิจารณา จากภาพ 4.8 ของ Mask R–CNN +VGG16 + categorical cross entropy สรุปได้ว่าช่วงที่สนใจคือ ช่วงที่มี confidence80-100 มีจำนวนต้นที่ทายผิดที่มากที่สุดซึ่งหมายความว่าโมเดลมีการเอนเอียงไปทางใดทางหนึ่งอาจเป็นเพราะ **ข้อมูลไม่สมดุลย์** และ Mask R–CNN + VGG16 + focal loss สรุปได้ว่าช่วงที่สนใจคือ ช่วงที่มี Confidence 80 -100 มีต้นทุเรียนที่เป็นโรคจากโมเดลเดิมมี 4 ต้น ลดลง 3 ต้นเหลือ 1 ต้น, มังคุดจากโมเดลเดิมมี 2 ต้นลดลง 2 ต้น และ ต้นทุเรียนที่ไม่เป็นโรคโมเดลเดิมมี 6 ต้นลดลง 6 ต้น จึงสรุปได้ว่าการเอนเอียงไปในทาง Confidence 80 – 100 ลดลงเป็นอย่างมาก วิธีการ focal loss จึงอาจจะเป็นหนึ่งในวิธีการแก้ปัญหา **ข้อมูลไม่สมดุลย์** ดังภาพที่ 4.9 - 4.10

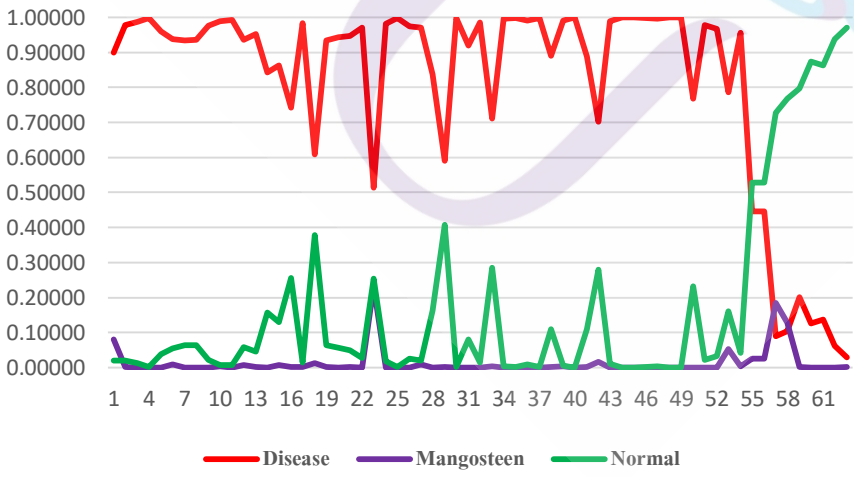



**ภาพที่ 4.9** เปรียบเทียบผลลัพธ์ ช่วงที่มี Confidenceที่ทำนายผิด ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss

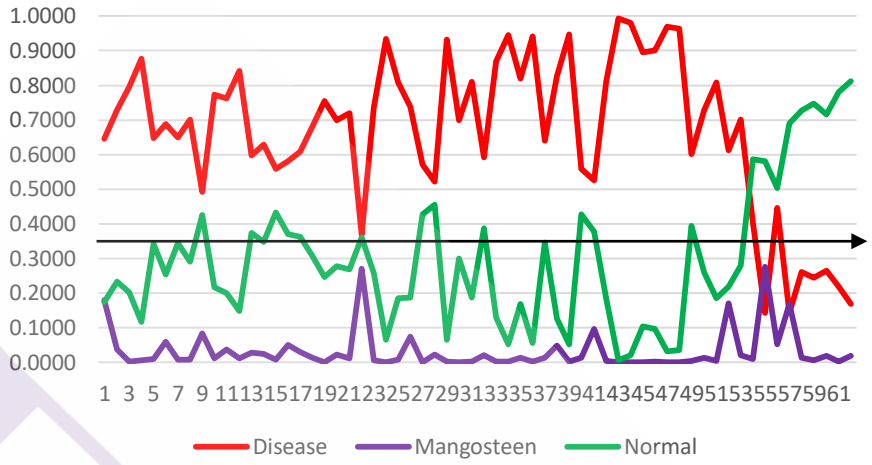



**ภาพที่ 4.10** แสดงการเปรียบเทียบจำนวนการทายผิดของ ของโมเดล Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy และ Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss

การทายของโมเดลจะเป็น Probability ข้อสังเกตต้นที่เป็นโดยส่วนมากในกรณีที่ทายผิดมักจะมี Probability ที่มีความใกล้เคียงกันมากเช่น Mangosteen(10.28%) Normal(48.48%), Disease(41.27%), ทั้งๆ ที่รูปดังกล่าวเป็น Disease อยู่แล้ว ด้วยวัตถุประสงค์ของงานวิจัยฉบับนี้มุ่งเน้นเพื่อหาต้นทุเรียนที่เป็นโรค อาจจะต้องกำหนดค่า Probability ที่เหมาะสมสำหรับต้นทุเรียนที่เป็นโรคในกรณีที่รูปเป็น ต้นที่เป็นโรค โดยพิจารณาจากภาพที่ 4.11 - 4.12 หาค่า Probability ที่เหมาะสมและเป็นการเพิ่มจำนวนต้นทุเรียนที่เป็นโรค ด้วย

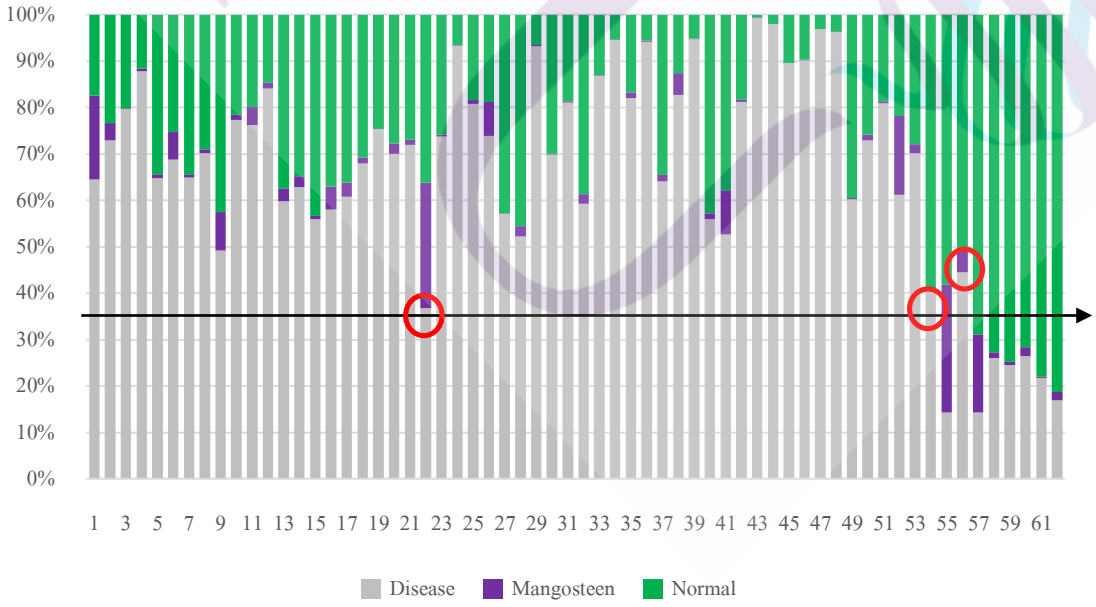



**ภาพที่ 4.11** แสดงการเปรียบเทียบ Probability Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น Categorical cross Entropy



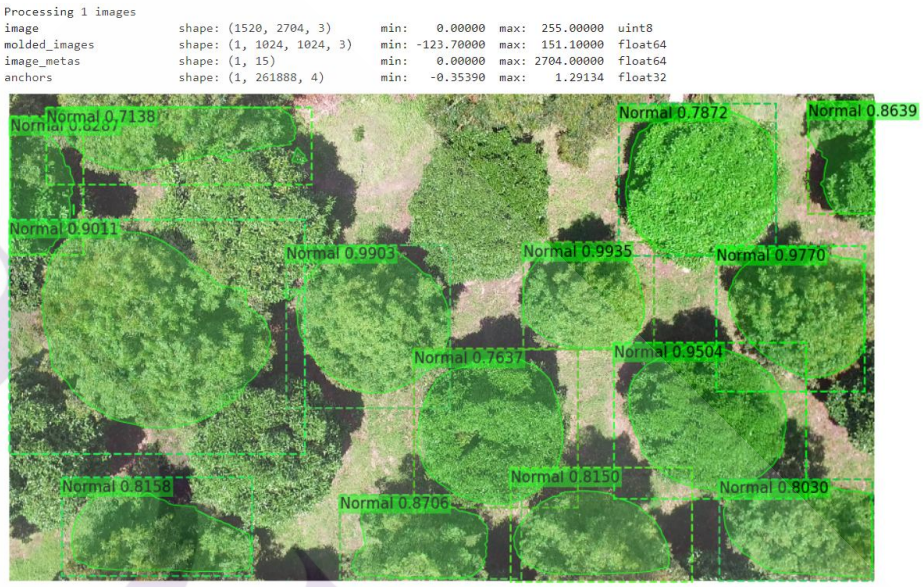

**ภาพที่ 4.12** แสดงการเปรียบเทียบของช่วงที่มี Probability Mask R–CNN +VGG16 โดยใช้ฟังก์ชั่น Loss เป็น focal loss

แปรงรูป 4.12 เป็นกราฟแผนภูมิคอลัมน์แบบเรียงซ้อน 100% ตามรูปเพื่อที่จะดูค่า Probability โดยรวม จากภาพที่ 4.9 ถ้าใช้อ้างอิงตาม Probability ที่มากที่สุดให้ตอบ Class นั้นจะตรวจจับต้นที่เป็นโรคได้ 53 ต้น แต่ถ้าปรับ Probability ที่เหมาะสมและต้นไม้ที่เป็นโรคจริง ๆ แล้ว เมื่อลอง Probability ที่ 35% จะได้ต้นที่เป็นโรคเพิ่มขึ้นอีก 3 ต้นเป็น 56 ต้น ดังภาพที่ 4.13 ดังนั้นวิธีการ Mask R–CNN + VGG16 + focal loss เป็นวิธีการดีที่สุด

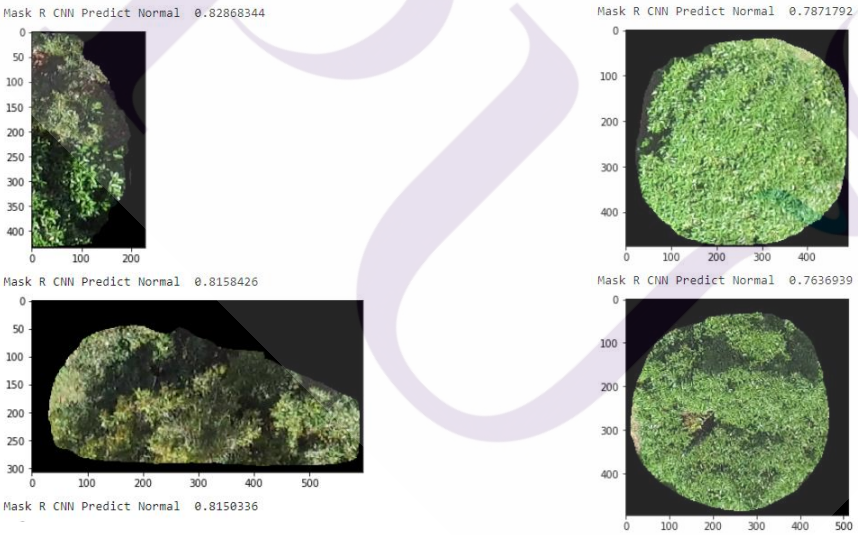



**ภาพที่ 4.13** แสดงกราฟแผนภูมิคอลัมน์แบบเรียงซ้อน 100% พร้อมทั้งตัด Probability

ตัวอย่างการทำงาน ของ Cascade Model ตามรูป ดังภาพที่ 4.14 – 4.15

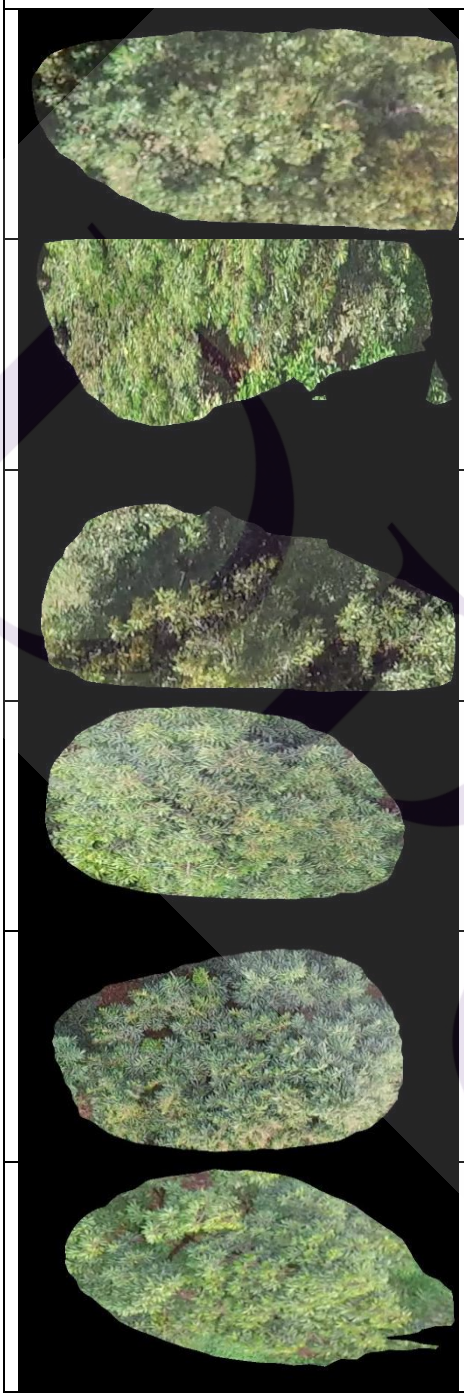



**ภาพที่ 4.14** รูปที่ Mask R-CNN ทำการค้นหาและระบุตำแหน่ง




**ภาพที่ 4.15** รูปที่ Mask R-CNN ทำการค้นหาและระบุตำแหน่ง และตัดมาทีละต้น

**ตารางที่ 4.1** ผลการทดลองของ Mask R-CNN กับ Mask R-CNN + VGG16

| รูป | Mask R - CNN | Mask R-CNN + VGG16 |
|---|---|---|
|  | Actual = Normal<br>Predict = Normal (80.29%) | Actual = Normal<br>Predict = Disease (6.343%)<br>Mangosteen (2.50%)<br>Normal (91.157%) |
| | Actual = Normal<br>Predict = Normal (71.38%) | Actual = Normal<br>Predict = Disease (3.420%)<br>Mangosteen (19.912%)<br>Normal (76.668%) |
| | Actual = Normal<br>Predict = Normal (81.58%) | Actual = Normal<br>Predict = Disease (4.781%)<br>Mangosteen (16.642%)<br>Normal (80.578%) |
| | Actual = Normal<br>Predict = Normal (74.20%) | Actual = Normal<br>Predict = Disease (3.207%)<br>Mangosteen (22.018%)<br>Normal (74.775%) |
| | Actual = Normal<br>Predict = Normal (73.11%) | Actual = Normal<br>Predict = Disease (3.273%)<br>Mangosteen (16.085%)<br>Normal (80.642%) |
| | Actual = Normal<br>Predict = Normal (80.09%) | Actual = Normal<br>Predict = Disease (4.193%)<br>Mangosteen (10.582%)<br>Normal (85.225%) |

# บทที่ 5
# บทสรุปและข้อเสนอแนะ

งานวิจัยในวิทยานิพนธ์เล่มนี้ได้กล่าวถึงการตรวจจับต้นทุเรียนที่เป็นโรคโดยใช้ภาพถ่ายมุมสูงจากโดรน โดยใช้การเรียนรู้เชิงลึกมีวิธีการทำงานเป็นแบบ (Cascade model) หรือโมเดลแบบลำดับขั้น โดยสรุปผลการศึกษาได้ดังนี้

## 5.1 สรุปผลการศึกษา

**5.1.1 ผลการปรับแต่ง Mask R-CNN** มีความสามารถในการตรวจจับต้นที่เป็นโรคได้ดีในระดับหนึ่งแต่ยังมีการทำนายผิดพลาด เป็นเพราะการ Train ข้อมูลเป็นไปได้ยากเพราะผู้ทำการศึกษาวิจัยนั้นมีทรัพยากรด้าน Hardware **ซึ่งวัดค่าได้ mAP = 0.794, recall = 0.679 และ f1 = 0.732** ซึ่งอยู่ในเกณฑ์ยอมรับได้

**5.1.2 ผลการปรับแต่ง Mask R-CNN + VGG16** มีความสามารถในการคัดแยกต้นเป็นโรคที่ได้ดีกว่า Mask R-CNN **ซึ่งวัดค่าได้ precision = 0.94, recall = 0.94 และ f1 = 0.94** ซึ่งอยู่ในเกณฑ์ยอมรับได้

## 5.2 อภิปรายผลการศึกษา

การทำงานวิจัยนี้ ควรใช้ หลาย ๆ โมเดลมาช่วยกันหาคำตอบ เพราะโมเดลเดียวยังไม่สามารถในการแก้ไขปัญหาได้บางส่วน ในงานวิจัยชิ้นนี้ใช้ 2 โมเดลที่เป็นแบบลำดับขั้น หรือ Cascade model โดยมีการแยกส่วนการทำงาน โดยโมเดลแรกใช้ระบุและหาตำแหน่งต้นไม้ และโมเดลที่สองใช้ในการแยกแยะต้นไม้ ซึ่งวิธีนี้จะให้ผลลัพธ์ที่ดี

## 5.2 ข้อเสนอแนะ

5.3.1 ไม่ควรลดขนาดของรูปที่ใช้ในการฝึกสอน

5.3.2 ควรใช้ กราฟิกการ์ด ที่มีประสิทธิภาพสูงในการฝึกสอน

5.3.3 ควรแบ่งข้อมูลเป็น Training data, Validation data และ Testing data และใช้ Validation data เป็นตัวปรับแต่งโมเดล

# บรรณานุกรม

# บรรณานุกรม

## ภาษาไทย

ชมพูทรัพย์ ปทุมสิน, อาทิตย์ ศรีแก้ว. (2562). การวินิจฉัยจำแนกโรคใบอ่อนจากภาพถ่ายโดยใช้จีเนติกส์อัลกอริทึมและแผนผังจัดการตนเองเชิงโครงสร้างปรับตัวได้. *วารสารวิชาการเทคโนโลยีอุตสาหกรรม ปี ที่ 14 ฉบับที่ 3 กันยายน –ธันวาคม 2561.* สืบค้น 12 มกราคม 2561 จาก http://178.128.214.142:81/storage/attachments/035/2_25611403_04_1-001-61_p44-61.pdf

สรุปแนวคิด Neural Network แบบไม่มี Math. (2560). สืบค้น 3 กุมภาพันธ์ 2561 จาก https://coladev.com/machine-learning/neural-network/2017/02/22/neural-network-basic

อนาคต เกษตร ในไทยแลนด์ 4.0. (2561). สืบค้น 7 มีนาคม 2561 จาก https://marketeeronline.co/ https://marketeeronline.co/archives/7375

## ภาษาต่างประเทศ

Amara.J, Bouaziz.B. & Algergawy.A. (2017) A Deep Learning-based Approach for Banana Leaf Diseases Classification. *Lecture Notes in Informatics (LNI).* Retrieved June 23, 2018. From http://btw2017.informatik.uni-stuttgart.de/slidesandpapers/E1-10/paper_web.pdf

Arden.D. (2017). Applied Deep Learning - Part 4: Convolutional Neural Networks. Retrieved Junly 18, 2018. From https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Bouroubi.Y, Bugnet.P, Nguyen-Xuan.T, Gosselin.C, Blec.C, Longchamps.C, & Vigneault.P,. (2018). Pest Detection on UAV Imagery using a Deep Convolutional Neural Network, Retrieved January 9, 2018. From https://ispag.org/proceedings/? action=download &item=5136

Brahimi.M, Arsenovic.M, Laraba.S, Sladojevic.S, Boukhalfa.K & Moussaoui.A. ( 2018). Deep Learning For Plant Diseases: Detection and Saliency map Visualization. Retrieved January 12, 2019. From https://www.researchgate.net/publication /325651831_Deep_Learning_for_Plant_Diseases_Detection_and_Saliency_Map_Visualisation

Danish.M. (2018). Beef Cattle Instance Segmentation Using Mask R-Convolutional Neural Network. Retrieved March 12, 2019. From https://www.semanticscholar.org/ paper/Beef-Cattle-Instance-Segmentation-Using-Mask-Neural-Danish/ea4b19569721f4b6ca2d5ddb5eda45a9c5bdad3b

Fei-Fei.L, Justin.J, & Serena.Y. (2017). Lecture 11: Detection and Segmentation. Retrieved May 27, 2018. From http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Fei-Fei.L, Justin.J, & Serena.Y. (2017). Lecture 7:Training Neural Networks Part 2. Retrieved May 28, 2018. From http://cs231n.stanford.edu/slides/2017/cs231n_2017_ lecture7.pdf

Hugegene. (2013). Implementation of Mean Average Precision (mAP) with Non-Maximum Suppression (NMS). Retrieved November 28, 2018. From https://towardsdatascience .com/implementation-of-mean-average-precision-map-with-non-maximum-suppression-f9311eb92522

Jonathan.H. (2018). mAP (mean Average Precision) for Object Detection. Retrieved August 28, 2019. From https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

Kaiming.H, Georgia.G, Piotr.D & Ross.G. (2018). Mask R-CNN. *arXiv:1703.06870*. Retrieved August 28, 2019. From https://arxiv.org/abs/1703.06870

Konstantinos P. Ferentinos. (2018). Deep learning models for plant disease detection and. diagnosis. Retrieved January 5, 2019. From https://www.sciencedirect.com/ science/article/pii/S0168169917311742

Lawrence Zitnick.C & Doll´ar.P. (2014). Edge Boxes: Locating Object Proposals from Edges. *Microsoft Research*. Retrieved August 28, 2019. From https://pdollar.github.io/files /papers/ZitnickDollarECCV14edgeBoxes.pdf

Maarit.W.(2019). From Modeling to Scoring: Confusion Matrix and Class Statistics. Retrieved May 15, 2019. From https://www.knime.com/blog/from-modeling-to-scoring-confusion-matrix-and-class-statistics

Muhammad.R. (2018). VGG16 – Implementation Using Keras. Retrieved May 22, 2019. From https://engmrk.com/vgg16-implementation-using-keras/

Nessessence. (2018). Deep learning คืออะไร?. Retrieved May 28, 2019. From https://www. thaiprogrammer.org/2018/12/deep-learning-คืออะไร/

Popular.N. (2018). VGG16 – Convolutional Network for Classification and Detection. Retrieved July 28, 2019. From https://neurohive.io/en/popular-networks/vgg16/

Ren Jie Tan. (2013). Breaking Down Mean Average Precision (mAP). Retrieved July 5, 2019. From https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

Rohith.G. (2018). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Retrieved July 11, 2019. From https://towardsdatascience.com /r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Sumit.S. (2018). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, Retrieved June 11, 2019 From https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sladojevic.S, Arsenovic.M, Anderla.A, Culibrk.D & Stefanovic.D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Retrieved July 11, 2019. From https://www.semanticscholar.org/paper/Deep-Neural-Networks-Based-Recognition-of-Plant-by-Sladojevic- Arsenovic/18e3eec4e4c259cc0cbce91daff37e95e05d57d0

Thananop.K. (2018) Object Detection (Part 1). Retrieved July 28, 2019. From https://medium.com/@sapjunior/object-detection-part-1-37bf39dc557f [23]

Zhang.J, Zhan.J, Ma.Y. (2017). Mask R-CNN. Retrieved August 28, 2019. From https://cseweb.ucsd.edu/classes/sp18/cse252C-a/CSE252C_20180509.pdf

# ภาคผนวก

**ภาพผนวกที่ 1** ภาพจาก IoU Mask R-CNN

**ภาพผนวกที่ 2** ภาพจากการทำนาย Mask R-CNN

**ภาพผนวกที่ 3** ตัวอย่างไฟล์ที่ 2

**ภาพผนวกที่ 4** ตัวอย่างไฟล์ที่ 3 (มังคุด)

**ภาพผนวกที่ 5** ตัวอย่างไฟล์ที่ 4

**ภาพผนวกที่ 5** ภาพตัวอย่างที่มีสิ่งปลูกสร้าง

## ประวัติผู้เขียน

| | |
|---|---|
| ชื่อ-นามสกุล | พันจ่าอากาศโท รัตนโชติ พันธุ์วิไล |
| ประวัติการศึกษา | วิศวกรรมศาสตร์บัณฑิต<br>สาขาวิศวกรรมคอมพิวเตอร์<br>มหาวิทยาลัยนเรศวร<br>ปีการศึกษา 2552 |
| ตำแหน่งและสถานที่ทำงานปัจจุบัน | เจ้าหน้าที่สถิติกำลังพล แผนกสถิติกำลังพล กองข้อมูลกำลังพล สำนักการปกครองและพัฒนากำลังพล<br>กรมกำลังพลทหารอากาศ,<br>กองทัพอากาศ กระทรวงกลาโหม |

ผลงานทางวิชาการ

- ธนภัทร ฆังคะจิตร, **รัตนโชติ พันธุ์วิไล.** (2562). การตรวจหาโรคพืชอัตโนมัติด้วยโดรนและวิธีการเรียนรู้เชิงลึก NCCIT2019, 2562(15), 217-222.