



การพัฒนาวิธีการจำแนกโรคพืชโดยใช้การเรียนรู้เชิงลึก



โดย  
นายจิรศักดิ์ วงษ์บงกชไพศาล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต

ภาควิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2564

ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

การพัฒนาวิธีการจำแนกโรคพืชโดยใช้การเรียนรู้เชิงลึก



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2 ระดับปริญญามหาบัณฑิต

ภาควิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร

ปีการศึกษา 2564

ลิขสิทธิ์ของมหาวิทยาลัยศิลปากร

DEVELOPMENT OF PLANT DISEASES CLASSIFICATION METHOD USING DEEP  
LEARNING



By  
MR. Jirasak WONGBONGKOTPAISAN

A Thesis Submitted in Partial Fulfillment of the Requirements  
for Master of Engineering (ELECTRICAL AND COMPUTER ENGINEERING)

Department of ELECTRICAL ENGINEERING

Graduate School, Silpakorn University

Academic Year 2021

Copyright of Silpakorn University

หัวข้อ	การพัฒนาวิธีการจำแนกโรคพืชโดยใช้การเรียนรู้เชิงลึก
โดย	จิรศักดิ์ วงษ์บงกชไพศาล
สาขาวิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผนก ก แบบ ก 2 ระดับปริญญา มหาบัณฑิต
อาจารย์ที่ปรึกษาหลัก	อาจารย์ ดร. โสภณ ผู้มีจรรยา

---

บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร ได้รับพิจารณาอนุมัติให้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย  
(รองศาสตราจารย์ ดร.จุไรรัตน์ นันทานิช)

พิจารณาเห็นชอบโดย

.....ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.ระพีพันธ์ แก้วอ่อน)

.....อาจารย์ที่ปรึกษาหลัก  
(อาจารย์ ดร.โสภณ ผู้มีจรรยา)

.....ผู้ทรงคุณวุฒิภายใน  
(อาจารย์ ดร.ภมร ศิลาพันธ์)

.....ผู้ทรงคุณวุฒิภายนอก  
(ผู้ช่วยศาสตราจารย์ ดร.วีรพล จิรจรีต )



620920061 : วิศวกรรมไฟฟ้าและคอมพิวเตอร์ แผน ก แบบ ก 2 ระดับปริญญาโทมหาบัณฑิต

คำสำคัญ : การเพิ่มข้อมูลภาพแบบเชิงพื้นที่ (LoBIA), การเพิ่มข้อมูลภาพโดยรวม, โครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN), VGG19, MobileNet

นาย จิรศักดิ์ วงษ์บงกชไพศาล: การพัฒนาวิธีการจำแนกโรคพืชโดยใช้การเรียนรู้เชิงลึก  
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก : อาจารย์ ดร. โสภณ ผู้มีจรรยา

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการเพิ่มข้อมูลเชิงพื้นที่ของการเกิดโรคกับข้อมูลฝึกสอน เรียกว่า Local-based images augmentation หรือ LoBIA สำหรับฝึกสอนโมเดลจำแนกโรคพืช โดยใช้การเรียนรู้เชิงลึก ซึ่งเป็นวิธีการเพิ่มข้อมูลลงบนภาพด้วยการกระจายข้อมูลแบบสุ่มบนพื้นที่บริเวณภาพใบไม้ จากวิธีที่ได้กล่าวข้างต้นจะแตกต่างกับการเพิ่มจำนวนข้อมูลภาพแบบ Data augmentation ทั่ว ๆ ไปที่กระทำกับข้อมูลภาพทั้งภาพ เช่น การหมุนภาพ ปรับสีภาพ ปรับความสว่าง หรือสร้างสัญญาณรบกวน เป็นต้น ในงานวิจัยนี้เรียกว่า Global-based images augmentation ด้วยวิธีการทำ LoBIA นี้จะทำให้การเพิ่มจำนวนรูปแบบของข้อมูลมีความหลากหลายและใกล้เคียงกับความเป็นจริงของการเกิดโรคบนใบไม้ได้มากที่สุด ค่าความถูกต้องในการจำแนกชนิดของโรคพืชที่ใช้กลุ่มข้อมูล Original+LoBIA เป็นชุดข้อมูลฝึกสอน ให้ผลลัพธ์ที่ใกล้เคียงหรือมากกว่าการฝึกสอนจากกลุ่มข้อมูล Original หรือฝึกสอนด้วยกลุ่มข้อมูล Original+Global เนื่องจากการใช้ข้อมูลฝึกสอน LoBIA จะมีจำนวนของข้อมูลที่น้อยกว่าจึงทำให้ตัวโมเดลสามารถเรียนรู้ได้รวดเร็วและประหยัดทรัพยากรของเครื่องคอมพิวเตอร์ ในงานนำข้อมูลภาพจากฐานข้อมูลออนไลน์ Kaggle และ PlantVillage มาใช้สำหรับการทำวิจัยเป็นข้อมูลของโรคพืช 2 ชนิด ได้แก่ ข้อมูลของโรคใบองุ่น และข้อมูลของโรคใบมันฝรั่ง และโมเดลที่ใช้ฝึกสอนชุดข้อมูล ประกอบด้วย โมเดลโครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN) ที่ออกแบบขึ้น โมเดล VGG19 และ MobileNet

620920061 : Major (ELECTRICAL AND COMPUTER ENGINEERING)

Keyword : Local-based images augmentation (LoBIA), Global-based images augmentation, Convolutional Neural Network (CNN), VGG19 model, MobileNet model

MR. JIRASAK WONGBONGKOTPAISAN : DEVELOPMENT OF PLANT DISEASES CLASSIFICATION METHOD USING DEEP LEARNING THESIS ADVISOR : SOPON PHUMEECHANYA, Ph.D.

This thesis presents a method for spatially increased diseases area of training data. We were called Local-based images augmentation or LoBIA in order to train model for classified plant diseases using deep learning which is a way to increases information to the image with spreading random data on image of leaf area. According to above method, it is different from data augmentation of general form that it is adjust image overall such as: image rotation, image colour adjustment, brightness adjustment, or generated noise etc. This method is called Global-based images augmentation in this thesis. According to LoBIA method, it will provide disparate data and similarly realistic disease generation data on leaf. Accuracy value of plant disease type classification from using deep learning model that trained by Original+LoBIA dataset. It produces results that close to or greater than deep learning model that trained by Original dataset and Original+Global dataset. Using the LoBIA training dataset will have amount of training data is smaller than using the Global training dataset, allowing the model to learn faster and save on computer resources. In this research, image datasets from online databases Kaggle and PlantVillage were used research. It is data on two types of plant via grape leaves diseases dataset and potato leaves diseases dataset. Models are used to training datasets consists of our Convolutional Neural Network (CNN) model, VGG19 model, and MobileNet model.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์เป็นอย่างดีได้รับความช่วยเหลือ และการให้คำปรึกษาจาก อาจารย์ ดร.โสภณ ผู้มีจรรยา อาจารย์ที่ปรึกษาวิทยานิพนธ์ และคณะอาจารย์กรรมการสอบวิทยานิพนธ์ทุกท่าน ได้แก่ ผู้ช่วยศาสตราจารย์ ดร.ระพีพันธ์ แก้วอ่อน ประธานสอบวิทยานิพนธ์ อาจารย์ ดร.ภมร ศิลาพันธ์ กรรมการผู้ทรงคุณวุฒิภายใน และผู้ช่วยศาสตราจารย์ ดร.วีรพล จิรจรีด กรรมการผู้ทรงคุณวุฒิภายนอก ที่คอยให้คำแนะนำในทุกขั้นตอนที่ได้ทำการศึกษารายวิชาวิทยานิพนธ์ การวางแผนการศึกษา กระบวนการทำงานวิจัยทางด้านวิศวกรรม และการนำเสนอผลงานทางด้านวิศวกรรม ตลอดจนการเขียนรายงานและรูปเล่มวิทยานิพนธ์เพื่อนำเสนองานวิจัย แก่ไขข้อบกพร่องต่างๆ ของงานทุกอย่าง รวมทั้งวิทยานิพนธ์ฉบับนี้

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า และคณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร อันเป็นสถานศึกษาที่ประสิทธิ์ประสาทวิชาความรู้ ขอกราบขอบพระคุณคณาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากรทุกท่านที่ได้ให้การสั่งสอน ให้คำแนะนำ และข้อคิดเห็นอันเป็นประโยชน์ต่อการทำวิทยานิพนธ์ ขอขอบคุณเจ้าหน้าที่และบุคลากรภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ได้ให้ความช่วยเหลือและอำนวยความสะดวกในด้านต่างๆ

สุดท้ายผลอันจะเป็นประโยชน์ความดีความงามทั้งปวง ที่เกิดขึ้นจากการศึกษาวิทยานิพนธ์นี้ ขอมอบแต่บิดา มารดา และครอบครัวที่เคารพยิ่ง ตลอดจนถึง เพื่อนนักศึกษาผู้ที่ได้ให้การสนับสนุนช่วยเหลือในทุกๆ ด้าน และหากมีข้อบกพร่องด้วยประการใดๆ ผู้วิจัยขอน้อมรับไว้ด้วยความขอบคุณยิ่ง

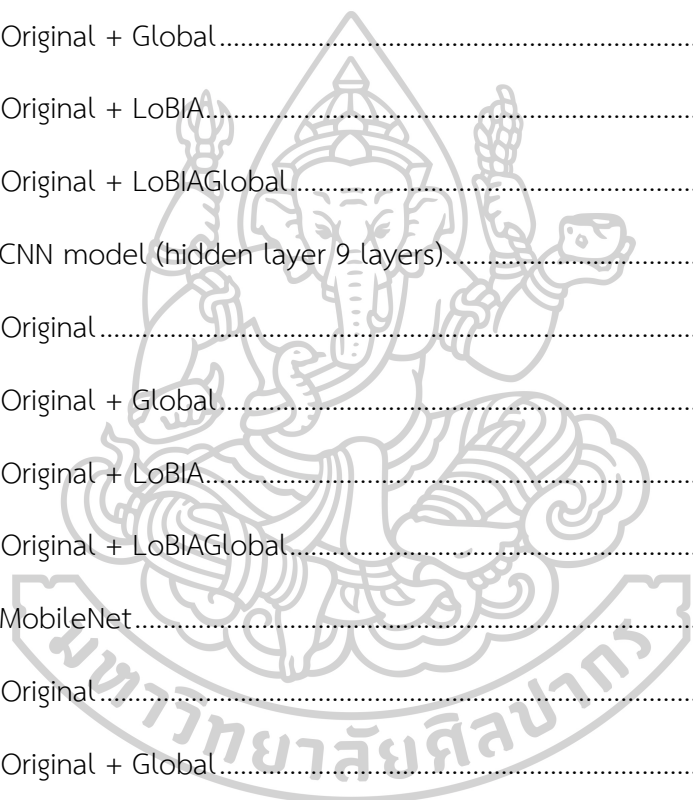
นาย จิรศักดิ์ วงษ์บงกชไพศาล

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญภาพ .....	ฉ
สารบัญตาราง.....	ต
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	4
1.3 เป้าหมายและขอบเขตแนวคิดของวิทยานิพนธ์ .....	4
1.4 ขั้นตอนและวิธีดำเนินงาน .....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	6
2.1 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 การมองเห็นสี .....	6
2.1.2 ระบบสี .....	8
ระบบของมันเซล (Munsell).....	8
ระบบสี CIE .....	10
2.1.3 การเปลี่ยนพิกัดสี.....	11
ภาพระดับสีเทา (Gray scale).....	11
ภาพขาว-ดำ (Binary).....	12

CIELAB 13	
2.1.4 ซุปเปอร์พิกเซล SLIC.....	18
2.1.5 โครงข่ายประสาทเทียม.....	20
โครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN) .....	23
โครงข่ายประสาทเทียมชนิดอื่น ๆ .....	32
VGG	32
MobileNet.....	34
2.1.6 การเพิ่มจำนวนข้อมูล.....	37
2.1 งานวิจัยที่เกี่ยวข้อง .....	39
บทที่ 3 วิธีการและขั้นตอนการดำเนินงาน.....	47
3.1 การแบ่งส่วนภาพไปไม้ด้วยวิธีซุปเปอร์พิกเซล SLIC สำหรับสร้างข้อมูลชุดฝึกสอนให้กับโมเดล CNN.....	48
3.2 การฝึกสอน CNN model สำหรับการแบ่งส่วนบริเวณที่เป็นโรคบนใบ .....	49
3.3 การแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ได้ฝึกสอนมาแล้ว .....	51
3.4 การเพิ่มจำนวนข้อมูลภาพด้วยการทำ Data Augmentation.....	52
3.5 ฝึกสอนโมเดลที่จะนำไปใช้สำหรับการจำแนกชนิดโรคพืช .....	55
3.6 ขั้นตอนของการทดสอบโมเดลที่ฝึกสอนมาแล้วกับชุดข้อมูลทดสอบ.....	56
บทที่ 4 ผลการทดลองของงานวิจัย.....	57
4.1 ชุดข้อมูลภาพที่ใช้ในการทดลอง.....	57
4.1.1 ชุดข้อมูลทดลองสำหรับการแบ่งส่วนบริเวณโรค.....	57
4.1.2 ชุดข้อมูลทดลองสำหรับการจำแนกโรค .....	58
Original.....	59
Original + Global.....	60
Original + LoBIA.....	61

Original + LoBIAGlobal.....	63
4.2 การวัดผลค่าความแม่นยำของการแบ่งส่วนบริเวณที่เป็นโรคด้วย F1-score.....	64
4.3 ผลการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ได้ฝึกสอนมาแล้ว.....	65
4.4 ผลการจำแนกโรคด้วยโมเดลการเรียนรู้เชิงลึก .....	70
4.4.1 CNN model (hidden layer 7 layers).....	72
Original.....	72
Original + Global.....	76
Original + LoBIA.....	81
Original + LoBIAGlobal.....	85
4.4.2 CNN model (hidden layer 9 layers).....	90
Original.....	90
Original + Global.....	94
Original + LoBIA.....	99
Original + LoBIAGlobal.....	103
4.4.3 MobileNet.....	108
Original.....	108
Original + Global.....	112
Original + LoBIA.....	117
Original + LoBIAGlobal.....	121
4.4.4 VGG19.....	126
Original.....	126
Original + Global.....	130
Original + LoBIA.....	134
Original + LoBIAGlobal.....	139



4.5 ผลที่เกิดจากการใช้ Local-based image augmentation ที่ทำให้โมเดลมีความสามารถในการจำแนกดีขึ้น .....	146
บทที่ 5 สรุปและข้อเสนอแนะ .....	149
5.1 สรุปผลการวิจัย.....	149
5.2 ข้อเสนอแนะ.....	150
5.3 แนวทางการพัฒนาต่อยอด .....	150
รายการอ้างอิง .....	152
ภาคผนวก.....	157
ภาคผนวก ก.....	158
ประวัติผู้เขียน .....	172



## สารบัญภาพ

	หน้า
รูปที่ 1.1 ภาพรวมการทำงานของวิทยานิพนธ์.....	3
รูปที่ 2.1 ช่วงความยาวคลื่นของสีที่มองเห็น .....	6
รูปที่ 2.2 การมองเห็นสีวัตถุ .....	6
รูปที่ 2.3 ส่วนขยายของเซลล์ภายในจอประสาทตา .....	7
รูปที่ 2.4 กราฟค่าความไวแสงของเซลล์รูปกรวย .....	8
รูปที่ 2.5 ระบบสีของมันเชล (Munsell).....	9
รูปที่ 2.6 เฉดสีหลักและเฉดสีรองของระบบสี Munsell .....	9
รูปที่ 2.7 ลักษณะการไล่ระดับสีในระบบ Munsell .....	10
รูปที่ 2.8 ระบบสี $L^* a^* b^*$ .....	11
รูปที่ 2.9 ระดับความเข้ม Gray scale 0-255 (8-bits).....	11
รูปที่ 2.10 ผลลัพธ์ของการแปลงภาพ RGB เป็น Gray scale.....	12
รูปที่ 2.11 ผลลัพธ์ของการแปลงภาพ RGB เป็น Binary .....	12
รูปที่ 2.12 CIELAB color space .....	13
รูปที่ 2.13 LAB .....	17
รูปที่ 2.14 ตัวอย่างลักษณะการแบ่งส่วนภาพด้วยซูเปอร์พิกเซล .....	19
รูปที่ 2.15 เซลล์ประสาท (Neuron).....	20
รูปที่ 2.16 ศักยะงาน (Action Potential).....	21
รูปที่ 2.17 แบบจำลองโครงข่ายประสาทเทียม .....	21
รูปที่ 2.18 ฟังก์ชันถ่ายโอนฮาร์ดลิมีต .....	22
รูปที่ 2.19 โครงสร้างโครงข่ายประสาทเทียมแบบสังวัตนาการ (Convolutional neural network) .....	24
รูปที่ 2.20 ตัวอย่างลักษณะของภาพต้นแบบ (Kernel) .....	25



รูปที่ 2.21 ข้อมูลภาพนำเข้า (Input image no padding).....	26
รูปที่ 2.22 cross-correlation (Stride = 1).....	28
รูปที่ 2.23 Feature map.....	29
รูปที่ 2.24 Cross-correlation (Stride = 1 and Zero padding) .....	29
รูปที่ 2.25 ขั้นตอนการแปลงข้อมูลผ่านฟังก์ชันการแปลง .....	30
รูปที่ 2.26 Max pooling.....	30
รูปที่ 2.27 Average pooling.....	31
รูปที่ 2.28 Flattening layer.....	31
รูปที่ 2.29 Fully connected.....	32
รูปที่ 2.30 โครงสร้างโมเดล VGG16.....	33
รูปที่ 2.31 โครงสร้างโมเดล VGG19.....	34
รูปที่ 2.32 โครงสร้างการทำ Depthwise Separable Convolution.....	36
รูปที่ 2.33 ลักษณะของการเพิ่มจำนวนข้อมูล “Images Augmentation” .....	37
รูปที่ 2.34 ภาพการตรวจจับโรคด้วยวิธีที่นำเสนอของงานวิจัยอ้างอิง .....	39
รูปที่ 2.35 โครงสร้างการทำ Transfer learning ของโมเดล Inception-V3 กับ DCNN.....	44
รูปที่ 2.36 การทำ Segmentation ด้วย Otsu thresholding .....	45
รูปที่ 2.37 โครงสร้างโมเดล D-CNN.....	45
รูปที่ 3.1 ภาพรวมของงานวิจัย.....	47
รูปที่ 3.2 ขั้นตอนของการเตรียมข้อมูลฝึกสอน .....	48
รูปที่ 3.3 บล็อกไดอะแกรมขั้นตอนการทำ Superpixel สำหรับสร้างชุดข้อมูลฝึกสอน.....	49
รูปที่ 3.4 CNN model ที่ใช้ในการจำแนกบริเวณที่เป็นโรคและบริเวณใบ .....	50
รูปที่ 3.5 ขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ผ่านการฝึกสอนมาแล้ว.....	51
รูปที่ 3.6 บล็อกไดอะแกรมขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรค.....	52
รูปที่ 3.7 ขั้นตอนการทำ Images augmentation .....	53

รูปที่ 3.8 บล็อกไดอะแกรมขั้นตอนการทำ Local-based image augmentation.....	54
รูปที่ 3.9 โมเดล CNN model 7 layers.....	55
รูปที่ 3.10 โมเดล CNN model 9 layers.....	56
รูปที่ 4.1 ตัวอย่างภาพบริเวณโรคสำหรับฝึกสอนแบ่งส่วน.....	57
รูปที่ 4.2 ตัวอย่างภาพบริเวณที่ไม่เป็นโรคสำหรับฝึกสอนแบ่งส่วน.....	58
รูปที่ 4.3 ตัวอย่างชุดภาพฝึกสอน (Original) ของใบองุ่น.....	59
รูปที่ 4.4 ตัวอย่างชุดภาพฝึกสอน (Original) ของใบมันฝรั่ง.....	59
รูปที่ 4.5 ตัวอย่างชุดภาพฝึกสอน (Original + Global) ของใบองุ่น.....	60
รูปที่ 4.6 ตัวอย่างชุดภาพฝึกสอน (Original + LoBIA) ของใบองุ่น.....	61
รูปที่ 4.7 ตัวอย่างชุดภาพฝึกสอน (Original + LoBIAGlobal) ของใบองุ่น.....	63
รูปที่ 4.8 ภาพตัวอย่างของภาพทดสอบการแบ่งส่วนโรค.....	65
รูปที่ 4.9 ภาพของการแบ่งส่วนโรค ที่ $K = 500$ .....	66
รูปที่ 4.10 ภาพของการแบ่งส่วนโรค ที่ $K = 600$ .....	67
รูปที่ 4.11 ภาพของการแบ่งส่วนโรค ที่ $K = 700$ .....	68
รูปที่ 4.12 ภาพของการแบ่งส่วนโรค ที่ $K = 800$ .....	68
รูปที่ 4.13 ผลลัพธ์ของการจัดกลุ่ม.....	69
รูปที่ 4.14 ตัวอย่างการทำ Data visualization แสดงข้อมูล Heat map.....	70
รูปที่ 4.15 ตัวอย่างภาพ Heat map ของข้อมูลที่ทำ Correlation matrix.....	71
รูปที่ 4.16 ตัวอย่างการทำ Data visualization จัดกลุ่มข้อมูลด้วย Swarm plot.....	71
รูปที่ 4.17 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original).....	72
รูปที่ 4.18 ผลการจัดกลุ่ม Swarm plot.....	74
รูปที่ 4.19 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)..	74
รูปที่ 4.20 ผลการจัดกลุ่ม Swarm plot.....	76

รูปที่ 4.21 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global) .....	76
รูปที่ 4.22 ผลการจัดกลุ่ม Swarm plot .....	78
รูปที่ 4.23 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global) .....	79
รูปที่ 4.24 ผลการจัดกลุ่ม Swarm plot .....	80
รูปที่ 4.25 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIA) .....	81
รูปที่ 4.26 ผลการจัดกลุ่ม Swarm plot .....	82
รูปที่ 4.27 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA) .....	83
รูปที่ 4.28 ผลการจัดกลุ่ม Swarm plot .....	84
รูปที่ 4.29 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIAGlobal) .....	85
รูปที่ 4.30 ผลการจัดกลุ่ม Swarm plot .....	87
รูปที่ 4.31 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal) .....	87
รูปที่ 4.32 ผลการจัดกลุ่ม Swarm plot .....	89
รูปที่ 4.33 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original) .....	90
รูปที่ 4.34 ผลการจัดกลุ่ม Swarm plot .....	92
รูปที่ 4.35 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original) ..	92
รูปที่ 4.36 ผลการจัดกลุ่ม Swarm plot .....	94
รูปที่ 4.37 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global) .....	94
รูปที่ 4.38 ผลการจัดกลุ่ม Swarm plot .....	96

รูปที่ 4.39 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global) .....	97
รูปที่ 4.40 ผลการจัดกลุ่ม Swarm plot .....	98
รูปที่ 4.41 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIA) .....	99
รูปที่ 4.42 ผลการจัดกลุ่ม Swarm plot .....	100
รูปที่ 4.43 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA) .....	101
รูปที่ 4.44 ผลการจัดกลุ่ม Swarm plot .....	102
รูปที่ 4.45 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIAGlobal) .....	103
รูปที่ 4.46 ผลการจัดกลุ่ม Swarm plot .....	105
รูปที่ 4.47 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal) .....	105
รูปที่ 4.48 ผลการจัดกลุ่ม Swarm plot .....	107
รูปที่ 4.49 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original) ....	108
รูปที่ 4.50 ผลการจัดกลุ่ม Swarm plot .....	110
รูปที่ 4.51 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)	110
รูปที่ 4.52 ผลการจัดกลุ่ม Swarm plot .....	112
รูปที่ 4.53 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global) .....	112
รูปที่ 4.54 ผลการจัดกลุ่ม Swarm plot .....	114
รูปที่ 4.55 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global) .....	115
รูปที่ 4.56 ผลการจัดกลุ่ม Swarm plot .....	116

รูปที่ 4.57 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIA) .....	117
รูปที่ 4.58 ผลการจัดกลุ่ม Swarm plot .....	118
รูปที่ 4.59 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA) .....	119
รูปที่ 4.60 ผลการจัดกลุ่ม Swarm plot .....	120
รูปที่ 4.61 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIAGlobal) .....	121
รูปที่ 4.62 ผลการจัดกลุ่ม Swarm plot .....	123
รูปที่ 4.63 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal) .....	123
รูปที่ 4.64 ผลการจัดกลุ่ม Swarm plot .....	125
รูปที่ 4.65 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original) ....	126
รูปที่ 4.66 ผลการจัดกลุ่ม Swarm plot .....	128
รูปที่ 4.67 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)	128
รูปที่ 4.68 ผลการจัดกลุ่ม Swarm plot .....	129
รูปที่ 4.69 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global) .....	130
รูปที่ 4.70 ผลการจัดกลุ่ม Swarm plot .....	132
รูปที่ 4.71 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global) .....	132
รูปที่ 4.72 ผลการจัดกลุ่ม Swarm plot .....	134
รูปที่ 4.73 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIA) .....	134
รูปที่ 4.74 ผลการจัดกลุ่ม Swarm plot .....	136

รูปที่ 4.75 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA) .....	137
รูปที่ 4.76 ผลการจัดกลุ่ม Swarm plot .....	138
รูปที่ 4.77 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIAGlobal) .....	139
รูปที่ 4.78 ผลการจัดกลุ่ม Swarm plot .....	140
รูปที่ 4.79 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal) .....	141
รูปที่ 4.80 ผลการจัดกลุ่ม Swarm plot .....	142
รูปที่ 4.81 ภาพตัวอย่างโรค Black measles ที่ทำนายผิด .....	146
รูปที่ 4.82 ภาพตัวอย่างโรค Black rot ที่ทำนายผิด .....	147
รูปที่ 4.83 ภาพตัวอย่างโรค Leaf blight ที่ทำนายผิด .....	147
รูปที่ 4.84 ภาพตัวอย่างโรค Early blight ที่ทำนายผิด .....	148
รูปที่ 4.85 ภาพตัวอย่างโรค Late blight ที่ทำนายผิด .....	148



## สารบัญตาราง

	หน้า
ตารางที่ 2.1 CIE 2 Degree Standard Observer .....	15
ตารางที่ 2.2 CIE 10 Degree Standard Observe .....	15
ตารางที่ 2.3 ตารางแสดงชนิดของฟังก์ชันถ่ายโอน (Types of transfer function).....	23
ตารางที่ 2.4 โครงสร้างโมเดล MobileNet.....	35
ตารางที่ 2.5 ตัวอย่างของการเกิด Underfitting และ Overfitting .....	38
ตารางที่ 2.6 ผลลัพธ์ค่าความผิดพลาดของงานวิจัยที่อ้างอิง .....	39
ตารางที่ 2.7 ผลลัพธ์จำนวนรอบและเวลาในการวิเคราะห์ของงานวิจัยที่อ้างอิง.....	40
ตารางที่ 2.8 ผลลัพธ์ค่าความแม่นยำ.....	40
ตารางที่ 2.9 ตารางเปรียบเทียบโมเดลกับการทำงานของชนิด Optimizers.....	41
ตารางที่ 2.10 ผลการทดลอง.....	42
ตารางที่ 2.11 ผลการทดลองเบื้องต้นของโมเดล MobileNet .....	43
ตารางที่ 2.12 ตารางผลของการเลือกใช้ค่า Optimizers.....	43
ตารางที่ 4.1 ตารางเปรียบเทียบผลค่า $K = 500, 600, 700, 800$ .....	69
ตารางที่ 4.2 ตารางผลทดสอบการจำแนกชนิดโรคของใบองุ่น.....	144
ตารางที่ 4.3 ตารางผลทดสอบการจำแนกชนิดโรคของใบมันฝรั่ง .....	145
ตารางที่ 4.4 ตารางผลทดสอบของแต่ละชนิดของชุดข้อมูลฝึกสอน .....	146



## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัญญาประดิษฐ์ (Artificial Intelligent) [1] หรือที่เรียกว่า AI เป็นระบบประมวลผลข้อมูลที่สร้างขึ้นให้กับเครื่องจักร (ในที่นี้กล่าวถึงคอมพิวเตอร์) ให้ได้เรียนรู้และสามารถวิเคราะห์ แก้ไขปัญหาได้ด้วยตัวเอง โดยโครงสร้างของ AI มีต้นแบบมาจากโครงข่ายประสาทของมนุษย์ที่สามารถเรียนรู้เพิ่มประสบการณ์ และเพิ่มประสิทธิภาพการประมวลผลได้ตามจำนวนข้อมูลที่ผ่านกระบวนการเรียนรู้ด้วยตัวเอง ดังนั้น AI จึงมีความสามารถในการเรียนรู้ วิเคราะห์ เชื่อมโยงความสัมพันธ์ และปรับปรุงเพิ่มประสบการณ์ให้กับตัวมันเองได้จากการเรียนรู้ของข้อมูลที่ซับซ้อนได้อย่างรวดเร็วเหมือนการทำงานของสมองมนุษย์

ปัญญาประดิษฐ์สามารถแบ่งออกได้เป็นหลายแขนงซึ่งในที่นี้จะพูดถึง การเรียนรู้ของเครื่อง (Machine learning) และ การเรียนรู้เชิงลึก (Deep learning) ซึ่งถือเป็นแขนงหนึ่งของปัญญาประดิษฐ์ที่ได้รับความนิยมเป็นอย่างมากเนื่องจากสามารถวิเคราะห์ข้อมูลที่มีความซับซ้อนได้เป็นอย่างดี โดยการทำงานของอัลกอริทึมนี้จะเป็นการทำงานโดยรวบรวมข้อมูลขนาดใหญ่ หรือข้อมูลที่มีการเปลี่ยนแปลงตลอดเวลา เรียกว่า Big data [2] มาทำการประมวลผลและเรียนรู้ซ้ำ ๆ กันหลาย ๆ รอบเพื่อให้เกิดการรู้จำและปรับปรุงประสิทธิภาพของตัวมันเองเพื่อที่จะนำไปใช้ในงานจริงได้ โดยการเรียนรู้ของข้อมูลนั้นดั้งเดิมจะเป็นการเรียนรู้จากโมเดลที่เรียกว่า โครงข่ายประสาทเทียม (Artificial neural network) หรือ ANN เป็นโครงข่ายจำลองการทำงานของโครงข่ายประสาทมนุษย์อย่างง่าย โดยสามารถเรียนรู้ที่จะทำงานที่ได้รับมอบหมายได้ จากการเรียนรู้ผ่านชุดข้อมูลตัวอย่าง หรือชุดข้อมูลทดสอบ (Training dataset)

ปัจจุบันเทคโนโลยีปัญญาประดิษฐ์เข้ามามีบทบาทในภาคอุตสาหกรรมอย่างแพร่หลายตั้งแต่ คมนาคม การแพทย์ การผลิต ไปตลอดจนถึงภาคการเกษตร ในที่นี้การทำเกษตรกรรมในยุคปัจจุบันก็ได้มีการนำเอาเทคโนโลยีนี้เข้ามาเป็นตัวช่วยการทำงานเพื่อปรับปรุงและเพิ่มประสิทธิภาพในการทำงาน กำลังการผลิตและผลผลิตทางการเกษตรให้มีคุณภาพและจำนวนผลผลิตให้ได้มากขึ้นเพื่อตอบสนองความต้องการ การเกษตรอัจฉริยะ หรือ Intelligent framing [3] เป็นชื่อเรียกการทำเกษตรสมัยใหม่ที่มีรูปแบบการทำเกษตรโดยนำเอาเทคโนโลยีและการจัดการจัดเก็บข้อมูลอย่างเป็นระบบมาใช้งาน เพื่อบริหารจัดการให้มีความถูกต้องและแม่นยำมากขึ้นในการจำแนกหรือทำนายสิ่งที

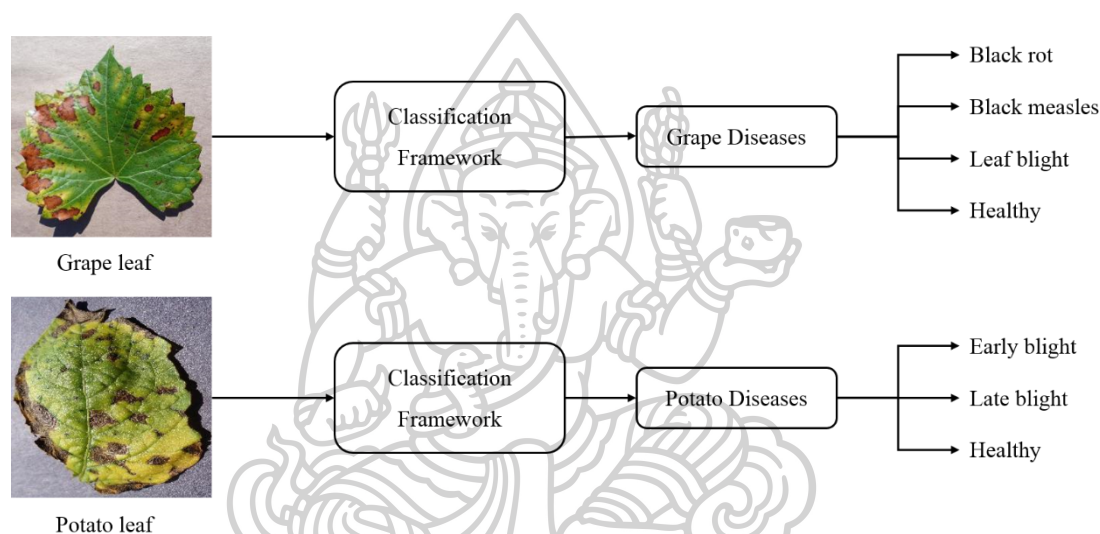


จะเกิดกับผลผลิตทางการเกษตรไปจนถึงการควบคุมประสิทธิภาพและคุณภาพการผลิต การนำเอา AI มาใช้อาจเริ่มตั้งแต่ขั้นตอนการเตรียมเมล็ดช่วยสำหรับคัดแยกเมล็ดพันธุ์ที่มีคุณภาพ ใช้ AI ช่วยในขั้นตอนการเพาะปลูกในการพยากรณ์สภาพอากาศหรือควบคุมสภาพแวดล้อมให้มีความเหมาะสมกับการเพาะปลูกพืชแต่ละชนิดเนื่อง เป็นต้น โดยปัจจัยหลักของการนำเอา AI มาใช้ในงานด้านการเกษตรกรรมเพื่อที่จะสามารถวิเคราะห์ค่าความแปรปรวนหรือตัวแปรที่มีผลต่อการเจริญเติบโตของพืชได้เพราะเนื่องจากตัวแปรส่วนใหญ่มีความแปรปรวนสูง เช่น อุณหภูมิหรือความชื้นอากาศของแต่ละวันที่ไม่เหมือนกัน เป็นต้น [4, 5]

สำหรับการนำเอา AI มาวิเคราะห์จำแนกชนิดของโรคที่เกิดกับพืชถือเป็นวิธีการหนึ่งที่เป็นที่นิยมของกลุ่มนักวิจัยด้วยการผนวกความรู้ทางด้านการประมวลผลภาพและรู้จำรูปแบบภาพ (Image processing and pattern recognition) มาประยุกต์ใช้งานกับปัญญาประดิษฐ์เพื่อให้ได้โปรแกรมที่สามารถจำแนกโรคที่เกิดขึ้นบนใบได้ และสามารถแยกชนิดของโรคได้

จากที่ได้กล่าวมาข้างต้นเห็นได้ชัดว่าชุดข้อมูลหรือข้อมูลสำหรับฝึกฝนถือเป็นส่วนที่สำคัญที่สุดในการทำงานทางด้านวิทยาศาสตร์ข้อมูล และถือได้ว่าเป็นส่วนที่ผู้ทำวิจัยหรือทำงานทางด้านวิทยาศาสตร์ข้อมูลจะใช้เวลาอันที่สุดของการทำงานอีกด้วย เพราะเป็นส่วนที่ทำให้การวิเคราะห์ข้อมูลของอัลกอริทึมสามารถวิเคราะห์ข้อมูลได้ถูกต้องและแม่นยำยิ่งขึ้นจากการปรับปรุงข้อมูลให้มีความเหมาะสมกับการนำไปฝึกสอน โดยในงานวิจัยที่ผ่านมาได้มีการใช้วิธีต่าง ๆ ในการจัดการกับข้อมูลเพื่อให้ความเหมาะสม เช่น การเปลี่ยนสี (Converting color) [6] การปรับค่าสี (Color adjustment) การเปลี่ยนรูปทรงของภาพ (Composed) การสร้างสิ่งรบกวน (Noise) การทำภาพเบลอ (Blur) หรือการทำทุกขั้นตอนที่กล่าวมาโดยเป็นการเพิ่มจำนวนข้อมูลเชิงรูปภาพในรูปแบบ (Image augmentation) [7-9] หรือเรียกว่า Global-based image augmentation [10-13] เป็นต้น ข้อเสียของการเพิ่มจำนวนข้อมูลรูปภาพแบบเชิงพื้นที่นั้นไม่อาจตอบสนองรูปแบบการกระจายตัวของโรคที่เกิดขึ้นบนใบได้เพราะเนื่องจากลักษณะการเกิดโรคที่แท้จริงบนใบไม้ไม่สามารถคาดเดาพื้นที่ของการเกิดโรคได้ ผู้วิจัยจึงได้เสนอวิธีการจำแนกบริเวณที่เป็นใบไม้กับพื้นหลังด้วยวิธีการทำซูเปอร์พิกเซล (Superpixel) [14, 15] และกระจายตัวของโรคบนใบไม้ในเชิงพื้นที่เพื่อให้เกิดความหลากหลายของการเกิดโรคแบบกระจายตัวบนใบไม้ซึ่งมีความใกล้เคียงที่สุดในการเกิดโรคจริง เรียกวิธีการนี้ว่าการเพิ่มจำนวนข้อมูลวัตถุเชิงพื้นที่ (Local-based image augmentation : LoBIA)

ผู้วิจัยเล็งเห็นถึงความสามารถของตัวโปรแกรมที่ผู้วิจัยได้ทำการพัฒนาขึ้นมา ที่จะสามารถนำไปใช้ให้เกิดประโยชน์ไม่มากนักน้อยในงานทางด้านการเกษตร โดยเน้นเนื้อหาใจความสำคัญของงานวิจัยไปที่การจำแนกโรคพืชบนใบโดยใช้โมเดลที่ฝึกสอนจากการฝึกฝนข้อมูลที่ถูกเพิ่มจำนวนด้วยวิธีการทำ Global-based image augmentation และ Local-based image augmentation ดังนั้นในวิทยานิพนธ์นี้จึงมีจุดมุ่งหมายเพื่อที่จะนำเสนอ การพัฒนาวิธีการจำแนกโรคพืชโดยใช้การเรียนรู้เชิงลึก ให้มีประสิทธิภาพมากขึ้นในการจำแนกโรคพืช โดยลำดับวิธีการของการทำงานในวิทยานิพนธ์นี้แสดงดังรูปที่ 1.1



รูปที่ 1.1 ภาพรวมการทำงานของวิทยานิพนธ์

โดยจะเป็นการนำภาพของข้อมูลโรคพืชบนใบของจำนวน 4 โรค ได้แก่ Black rot, Black measles, Leaf blight และ Healthy และใบมันฝรั่งจำนวน 3 โรค ได้แก่ Early blight, Late blight และ Healthy มาใช้ในการทดลองโดยเริ่มจากการนำภาพไปทำการแบ่งส่วนบริเวณที่เป็นโรคเพื่อนำไปสร้างชุดข้อมูลฝึกสอนให้กับโมเดลการเรียนรู้เชิงลึกที่ออกแบบ (CNN model) และนำโมเดลที่ได้ฝึกสอนนั้นมาใช้เป็นตัวแบ่งส่วนบริเวณที่เป็นโรคเพื่อนำผลลัพธ์บริเวณที่เป็นโรคที่ได้นำไปใช้ในการกระจายข้อมูลโรคลงบนชุดข้อมูลภาพที่นำไปสร้างเป็นชุดข้อมูลฝึกสอน โดยจะมีจำนวนของชุดข้อมูลฝึกสอนทั้งหมด 4 ชุดฝึกสอน ประกอบด้วย 1. ชุดฝึกสอนภาพต้นฉบับ (Original) 2. ชุดฝึกสอนภาพที่ถูกเพิ่มจำนวนข้อมูลด้วยวิธี Global-based image augmentation (Original + Global) 3. ชุดข้อมูลฝึกสอนที่ถูกเพิ่มจำนวนข้อมูลด้วยวิธี Local-based image augmentation (Original + LoBIA) และ 4. ชุดข้อมูลฝึกสอนที่นำข้อมูลจาก LoBIA มาทำการเพิ่มข้อมูลด้วย Global augmentation (Original + LoBIAGlobal) โดยชุดข้อมูลฝึกสอนทั้ง 4 จะถูกนำไปฝึกสอนกับโมเดล

การเรียนรู้เชิงลึกทั้ง 4 โมเดล ได้แก่ CNN model (Hidden layer 7 layers), CNN model (Hidden layer 9 layers), MobileNet และ VGG19 สำหรับนำไปใช้ในการจำแนกชนิดของโรคพืช โดยโมเดลทั้ง 4 ชนิดเปรียบเสมือนโมเดลที่มีจำนวนของ Hidden layer จำนวนชั้นที่น้อย มีจำนวนชั้นที่กลาง และมีจำนวนชั้นที่มาก เพื่อให้เห็นการเปรียบเทียบของผลลัพธ์ที่ได้จากการทดสอบค่าความถูกต้องการจำแนกขึ้นอยู่กับจำนวนชั้นของโมเดลหรือไม่ และวิธีการใช้ชุดข้อมูลฝึกสอนที่นำเสนอให้ผลลัพธ์อย่างไรเมื่อเทียบกับวิธีการใช้ชุดข้อมูลฝึกสอนแบบอื่น ๆ

## 1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 เพื่อศึกษาสถาปัตยกรรมของโมเดลการเรียนรู้เชิงลึกที่นำมาใช้ในการจำแนกโรคพืช
- 1.2.2 เพื่อศึกษาเทคนิคการประมวลผลภาพสำหรับนำมาใช้ในการเพิ่มจำนวนภาพฝึกสอนสำหรับโมเดลการเรียนรู้เชิงลึก
- 1.2.3 ออกแบบและพัฒนาวิธีการจำแนกโรคพืชที่เกิดบนใบโดยใช้โมเดลการเรียนรู้เชิงลึก

## 1.3 เป้าหมายและขอบเขตแนวคิดของวิทยานิพนธ์

เป้าหมายของวิทยานิพนธ์นี้ คือ การออกแบบโปรแกรมเพื่อใช้สำหรับการจำแนกโรคพืชบนใบ ด้วยวิธีการประมวลผลภาพและการเรียนรู้เชิงลึก เพื่อหาและจำแนกพื้นที่ส่วนที่เป็นโรคกับพื้นที่ส่วนที่ไม่เป็นโรคอีกทั้งยังจำแนกชนิดของโรคอีกด้วยเพื่อนำไปใช้ในการตัดสินใจของงานทางด้านการเกษตรต่อไป โดยได้มีการศึกษาวิธีการปรับปรุงรูปภาพด้วยวิธีการประมวลผลภาพสำหรับนำไปใช้เป็นข้อมูลฝึกสอน และยังได้ทำการศึกษาโครงสร้างและการทำงานของการทำงานของการฝึกสอนข้อมูลด้วยวิธีการเรียนรู้เชิงลึกแบบต่าง ๆ ซึ่งจะกล่าวไว้ในบทที่ 2 เพื่อออกแบบระเบียบวิธีการจำแนกโรคพืชเพื่อให้สามารถจำแนกโรคพืชออกจากบริเวณที่ไม่เป็นโรคและจำแนกชนิดของโรคได้อย่างมีประสิทธิภาพ เนื่องจากได้มีการออกแบบการแบ่งส่วนของใบไม้ออกจากส่วนที่เป็นพื้นหลังด้วยวิธีแอกทีฟคอนทัวร์และซูเปอร์พิกเซลเพื่อควบคุมบริเวณที่เป็นเงาเพื่อไม่ให้เกิดความสับสนระหว่างส่วนที่เป็นเงาและเป็นโรค ซึ่งได้กล่าวไว้ในบทที่ 3

ภาพใบไม้ที่นำมาใช้ในการทดลองเป็นภาพสี (RGB color) ที่นำมาจากเว็บไซต์ฐานข้อมูลภาพ Kaggle.com [16] และ PlantVillage [17]

## 1.4 ขั้นตอนและวิธีดำเนินงาน

- 1.4.1 ศึกษาวิธีการแยกองค์ประกอบของภาพด้วยแอกทีฟคอนทัวร์และซูเปอร์พิกเซล

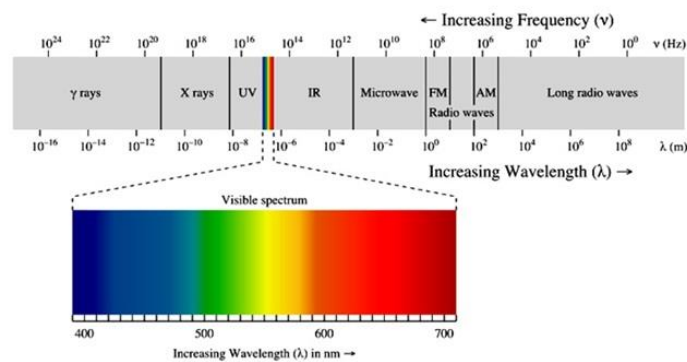
- 1.4.2 ศึกษาความสามารถและพฤติกรรมในการแบ่งองค์ประกอบของภาพด้วยซูเปอร์พิกเซลแบบต่าง ๆ
  - 1.4.3 ศึกษาความสำคัญของข้อมูลฝึกสอน
  - 1.4.4 ศึกษาโครงสร้างและค่าตัวแปรของอัลกอริทึม
  - 1.4.5 ออกแบบวิธีการแบ่งองค์ประกอบของใบไม้ออกจากพื้นหลัง
  - 1.4.6 ออกแบบวิธีการจำแนกส่วนที่เป็นโรคกับใบสำหรับเป็นข้อมูลฝึกสอน
  - 1.4.7 ออกแบบวิธีการเพิ่มจำนวนข้อมูลแบบเชิงรูปภาพและเชิงพื้นที่
  - 1.4.8 ออกแบบโครงสร้างทางสถาปัตยกรรมของโมเดลเรียนรู้เชิงลึก
  - 1.4.9 ทดลองวิธีการที่ออกแบบ เก็บผลการทดลอง และเปรียบเทียบ
  - 1.4.10 รวบรวมและสรุปผลงานวิจัย
- 1.5 ประโยชน์ที่คาดว่าจะได้รับ**
- 1.5.1 ได้ทราบถึงวิธีการแยกองค์ประกอบของภาพด้วยแอคทีฟคอนทัวร์และซูเปอร์พิกเซลแบบต่าง ๆ
  - 1.5.2 ได้ทราบถึงความสามารถของการแบ่งองค์ประกอบของภาพด้วยซูเปอร์พิกเซลแบบต่าง ๆ
  - 1.5.3 ได้ทราบถึงเกณฑ์และรูปแบบต่าง ๆ ของโมเดลการเรียนรู้เชิงลึก
  - 1.5.4 ได้ระเบียบวิธีการสร้างโปรแกรมในการแยกองค์ประกอบของใบไม้ออกจากพื้นหลัง
  - 1.5.5 ได้ระเบียบวิธีการแยกส่วนของพื้นที่บริเวณที่เป็นโรคกับบริเวณที่ไม่เป็นโรค
  - 1.5.6 ได้ระเบียบวิธีการสร้างโปรแกรมสำหรับสร้างข้อมูลใช้ในการฝึกฝน
  - 1.5.7 ได้โมเดลการเรียนรู้เชิงลึกที่ถูกปรับปรุงให้เหมาะสมกับการจำแนกโรคพืชบนใบ
  - 1.5.8 สามารถนำระเบียบวิธีการหรือโปรแกรมไปประยุกต์ใช้กับภาพอื่น ๆ ที่มีลักษณะโดยทั่วไปคล้ายกันได้

## บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

### 2.1 ทฤษฎีที่เกี่ยวข้อง

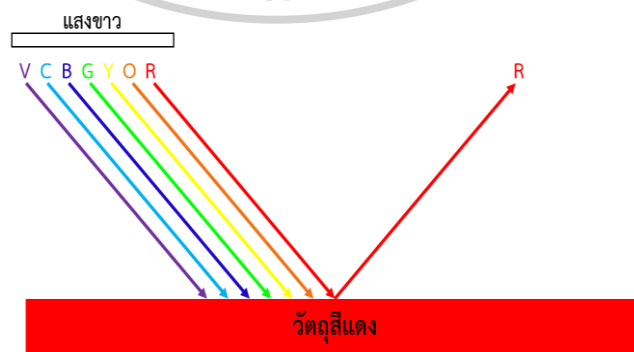
#### 2.1.1 การมองเห็นสี

การมองเห็นสีหรือการรับรู้สีของตามนุษย์ เกิดจากการที่แสงสีขาววิ่งมาตกกระทบบนวัตถุใด ๆ และเกิดการกระเจิงออกของแสงมาเข้าดวงตา โดยสีของวัตถุที่มองเห็นจะเป็นสี ณ ช่วงความยาวคลื่นช่วงหนึ่งที่วัตถุนั้น ๆ ไม่สามารถดูดซับสีนั้นได้และสะท้อนสีนั้นออกมากระทบยังดวงตามนุษย์ โดยแสงที่ตามนุษย์มองเห็น (Visible light) จะเป็นแสงที่อยู่ในช่วงความยาวคลื่น 380 นาโนเมตร ถึง 750 นาโนเมตร (380 nm – 750 nm) ประกอบด้วย 7 สี โดยเริ่มจากสีม่วง สีฟ้า สีน้ำเงิน สีเขียว สีเหลือง สีส้ม และสีแดง ตามลำดับ แสดงรูปที่ 2.1



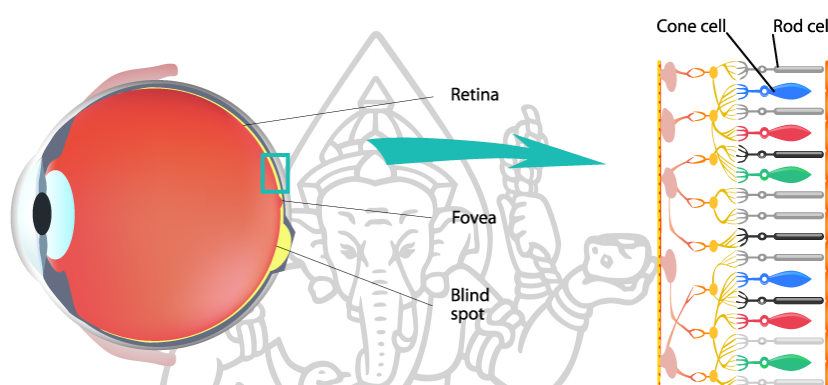
รูปที่ 2.1 ช่วงความยาวคลื่นของสีที่มองเห็น

ที่มา : [http://www.satriwit3.ac.th/external\\_newsblog.php?links=1241](http://www.satriwit3.ac.th/external_newsblog.php?links=1241)



รูปที่ 2.2 การมองเห็นสีวัตถุ

จากรูปที่ 2.2 เป็นตัวอย่างการมองเห็นสีของตามนุษย์ จะเห็นได้ว่าตามนุษย์นั้นสามารถมองเห็นวัตถุเป็นสีแดงเนื่องจากว่าวัตถุนั้นไม่สามารถหรือไม่ดูดซับสีแดงได้จึงเกิดการสะท้อนสีแดงออกจากตัววัตถุไปยังผู้สังเกตหรือตามนุษย์ทำให้ผู้สังเกตเห็นวัตถุนั้นนั้น ๆ เป็นสีแดง และเมื่อสีที่สะท้อนออกมาจากวัตถุเข้าสู่ตามนุษย์ โดยแสงจะผ่านเข้ามาภายในดวงตาเริ่มจากส่วนจอประสาทตา (Retina) ซึ่งเป็นบริเวณที่มีเซลล์รับแสงโดยภายในเรตินานั้นจะมีเซลล์สองประเภทที่ตอบสนองต่อการรับรู้ของสีที่มองเห็นคือ เซลล์รูปกรวย (Cone cell) และเซลล์รูปแท่ง (Rod cell) แสดงดังรูปที่ 2.3

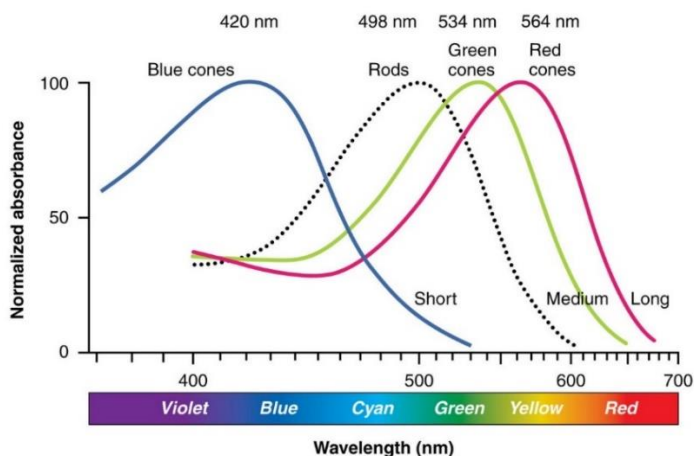


รูปที่ 2.3 ส่วนขยายของเซลล์ภายในจอประสาทตา

ที่มา : <https://www.benq.com/th-th/knowledge-center/knowledge/what-is-accurate-color.html>

เซลล์รูปกรวย หรือ Cone cell เป็นเซลล์ภายในจอประสาทตาที่มีความไวต่อแสงน้อยแต่ทำหน้าที่รับรู้สี และสามารถทำได้ดีที่สุดในสภาพแสงที่ค่อนข้างสว่าง ซึ่งจะตรงกันข้ามกับเซลล์รูปแท่งที่ทำงานได้ดีในสภาพแสงน้อยหรือแสงสลัว เซลล์รูปกรวยมีด้วยกัน 3 ชนิด แต่ละชนิดจะมีความไวต่อแสงสีปฐมภูมิ (RGB color) ต่างกัน ชนิดที่หนึ่ง S-type จะมีความไวสูงสุดต่อแสงสีน้ำเงินในช่วงความยาวคลื่นสั้น 420 นาโนเมตร ชนิดที่สอง M-type จะมีความไวสูงสุดต่อแสงสีเขียวในช่วงความยาวคลื่นกลาง 534 นาโนเมตร และชนิดที่สาม L-type จะมีความไวสูงสุดต่อแสงสีแดงในช่วงความยาวคลื่นยาว 564 นาโนเมตร แสดงดังรูปที่ 2.4





รูปที่ 2.4 กราฟค่าความไวแสงของเซลล์รูปกรวย  
ที่มา : <https://www.wikiwand.com/th/เซลล์รับแสง>

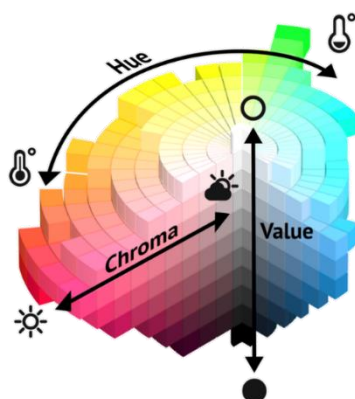
เซลล์รูปแท่ง หรือ Rod cell เป็นอีกหนึ่งในสองเซลล์ที่อยู่ภายในจอประสาทตาที่มีความไวต่อแสงมากและสามารถรับรู้แสงได้มากกว่าเซลล์รูปกรวยซึ่งจะทำหน้าที่รับแสงสลัวได้ดี แต่ไม่สามารถจำแนกสีได้ [18]

### 2.1.2 ระบบสี

การที่มนุษย์มองวัตถุที่มีสีนั้น จะเป็นการมองเห็นสีที่ได้ผลลัพธ์แตกต่างกันไปขึ้นอยู่กับมุมมองของผู้สังเกต แต่เพื่อให้เป็นมาตรฐานเดียวกันการวัดสีเพื่อให้ผู้สังเกตสามารถนำสีไปใช้ได้ง่ายและเป็นที่ยอมรับในระดับสากล ซึ่งมีหลายวิธี หลายระบบ ที่นิยมใช้กัน คือ ระบบของมันเชล (Munsell) และระบบ CIE

#### ระบบของมันเชล (Munsell)

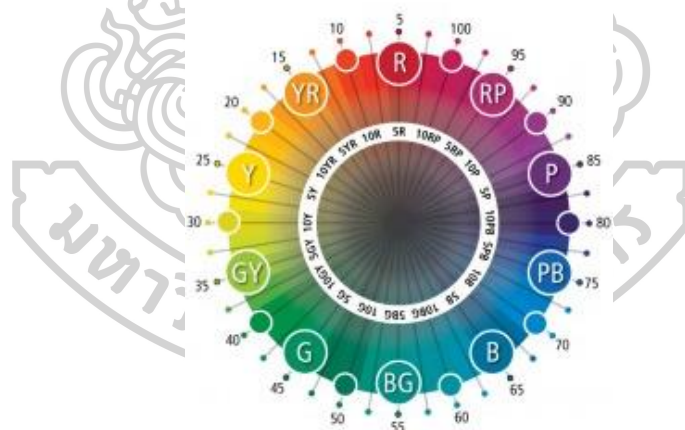
เป็นระบบการวัดสีที่จะบอกค่าลักษณะของสีเป็น 3 ลักษณะคือ ค่าเฉดสี (Hue) ค่าความสว่างของสี (Lightness/Value) และค่าความอิ่มตัวของสี (Saturation/Chroma) ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 ระบบสีของมันเซล (Munsell)

ที่มา : <https://www.subpng.com/png-b2601i/download.html>

ค่าเฉดสี (Hue) เป็นคุณสมบัติที่ระบุว่าเป็นสีใดสีหนึ่ง โดยในระบบ Munsell จะแบ่งเฉดสีออกเป็น 10 เฉดสี ประกอบด้วยเฉดสีหลัก 5 เฉดสีได้แก่ สีแดง สีเหลือง สีเขียว สีม่วง และ สีม่วงน้ำเงิน ที่เกิดจากการผสมกันของเฉดสีหลักอีก 5 เฉดสีได้แก่ สีส้ม สีเขียวเหลือง สีเขียวน้ำเงิน สีม่วงน้ำเงิน และสีม่วงแดง แสดงดังรูปที่ 2.6



รูปที่ 2.6 เฉดสีหลักและเฉดสีรองของระบบสี Munsell

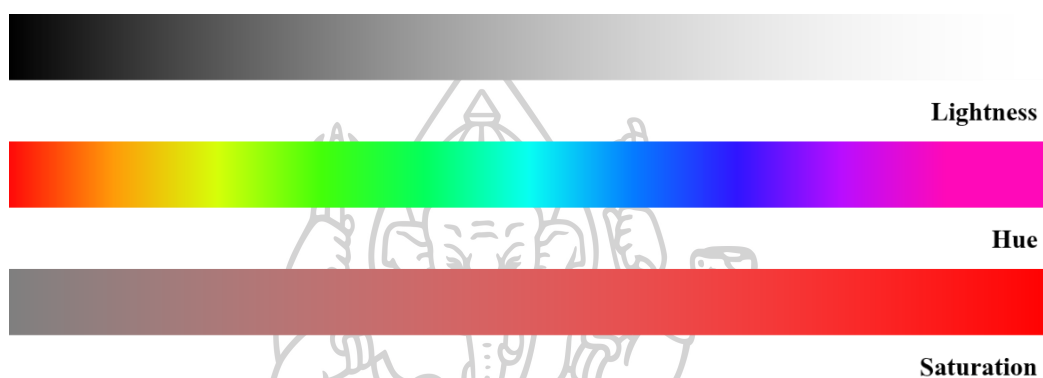
ที่มา : <https://munsell.com/about-munsell-color/how-color-notation-works/munsell-hue/>

ค่าความสว่างของสี (Lightness/Value) เป็นคุณสมบัติของค่าน้ำหนักอ่อนแก่ของกลุ่มสีดำ สีเทา และสีขาว โดยมีค่าน้ำหนักตั้งแต่ 0 (สีดำ) ไปจนถึง 10 (สีขาว) และค่าระหว่างสีดำกับสีขาว แบ่งเป็นค่าน้ำหนักระหว่างสีดำ สีเทา สีขาว โดยค่าน้ำหนักสีเทา เรียกว่า ค่าสีกลาง (Neutral) ค่า



ความสว่างของค่ากลางจะเป็นค่าสีที่ไม่มีคุณสมบัติของค่าเฉดสีอยู่ แต่จะมีแค่ค่าน้ำหนัก (Value) ซึ่งเมื่อนำไปผสมกับเฉดสี (Hue) จะทำให้ได้ค่าน้ำหนักอ่อนแก่ของสี

ค่าความอิ่มตัวของสี (Saturation/Chroma) คือค่าความสดใสของสีที่เกิดจากคุณสมบัติของค่าเฉดสี (Hue) ถูกผสมกับค่ากลาง (Neutral) ของความสว่าง (1-9) ทำให้ค่าความอิ่มตัวของสีนั้น ๆ เกิดการไล่ค่าน้ำหนักจากสีอ่อนสุด (Low saturation) ไปยังสีที่มีความจัดจ้านสุด (High saturation) แสดงดังรูปที่ 2.7

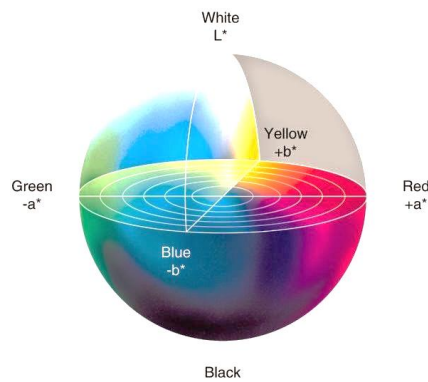


รูปที่ 2.7 ลักษณะการไล่ระดับสีในระบบ Munsell

### ระบบสี CIE

ระบบสี CIE เป็นระบบที่ CIE (Commission International de l' Eclairage) หรือ International commission on Illumination ซึ่งเป็นองค์กรที่มีบทบาทสำคัญในการกำหนดมาตรฐานด้านสี โดยองค์กรนี้ได้กำหนดมาตรฐานการวัดสีซึ่งเป็นที่ยอมรับอย่างกว้างขวางในวงการวิชาการและการวิจัยคือระบบสี CIE lab scale เป็นระบบสีที่ถูกพัฒนามาจากระบบสี CIEXYZ โดยใช้บรรยายสี แดง เขียว และน้ำเงิน แต่เนื่องจากระบบดังกล่าวไม่สามารถที่จะบรรยายถึงความมืดสว่างของสีได้

สำหรับระบบสี CIELAB เป็นระบบสีที่มีลักษณะการบรรยายแบบ 3 มิติ คือ  $L^* a^* b^*$  โดยในแกน  $L^*$  จะบรรยายถึงค่าความสว่าง (Lightness) ของสีจากค่า  $+L^*$  แสดงถึงสีขาว จนไปถึง  $-L^*$  แสดงถึงสีดำ แกน  $a^*$  เป็นแกนที่บรรยายแกนสีจาก  $-a^*$  แสดงสีเขียว ไปจนถึง  $+a^*$  แสดงถึงสีแดง และในส่วนของแกน  $b^*$  จะเป็นแกนที่ใช้บรรยายแกนสีจาก  $-b^*$  แสดงถึงสีน้ำเงิน ไปจนถึง  $+b^*$  แสดงถึงสีเหลือง แสดงได้ดังรูปที่ 2.8



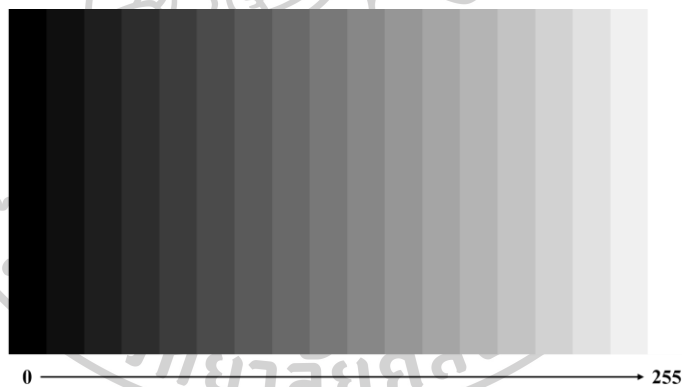
รูปที่ 2.8 ระบบสี  $L^*a^*b^*$

ที่มา : <https://xn--12cf0dj0aaufkr9l0ai2m6ab4p.blogspot.com/2014/06/lab.html>

### 2.1.3 การเปลี่ยนพิกัดสี

#### ภาพระดับสีเทา (Gray scale)

ภาพระดับสีเทา คือภาพ ขาว-เทา-ดำ ที่มีระดับความเข้มของสีเริ่มที่ 0 ถึง 255 (8-bits) แสดงดังรูปที่ 2.9

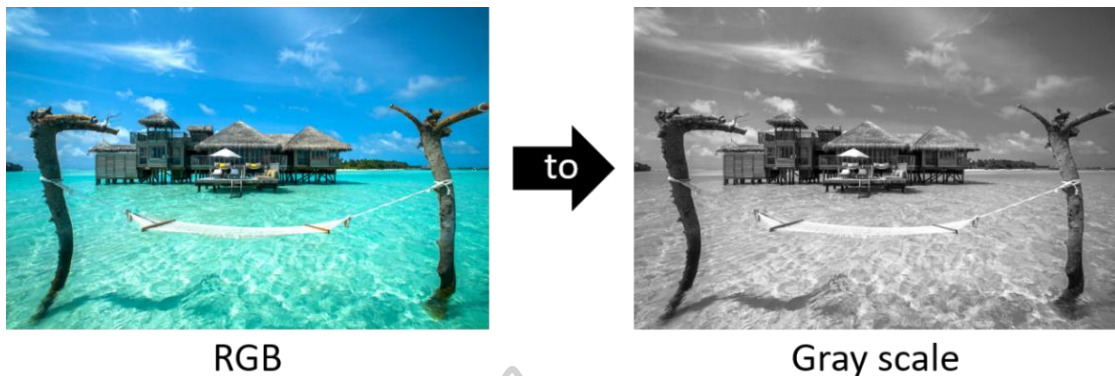


รูปที่ 2.9 ระดับความเข้ม gray scale 0-255 (8-bits)

โดยค่าความเข้มของระดับสีเทาที่เกิดจากการแปลงภาพสี RGB เป็นภาพ Gray scale หาได้ด้วย การหาค่าเฉลี่ยของค่าสีทั้งสามสีดังสมการที่ (2.1) และในสมการที่ (2.2) เป็นสมการการแปลงภาพ Gray scale ที่มีค่าน้ำหนักของแต่ละสีที่เหมาะสมและความใกล้เคียงกับการมองเห็นของตามนุษย์มากที่สุด ผลลัพธ์ที่ได้เป็นไปตามรูปที่ 2.10 [19]

$$gray\_scale = \frac{R + B + G}{3} \quad (2.1)$$

$$gray\_scale = 0.299R + 0.587G + 0.114B \quad (2.2)$$



รูปที่ 2.10 ผลลัพธ์ของการแปลงภาพ RGB เป็น Gray scale

#### ภาพขาว-ดำ (Binary)

ภาพขาว-ดำ เป็นการแปลงภาพจาก gray scale 0-255 (8-bits) ให้เป็นภาพที่แต่ละพิกเซลมีข้อมูลขนาด 1 บิต คือมีค่าแค่ 0 กับ 1 ในระบบดิจิทัล โดยที่ 0 จะมีค่าเป็นสีดำและ 1 มีค่าเป็นสีขาว แสดงดังรูปที่ 2.11 ผลลัพธ์ของการแปลงภาพ RGB เป็น Binary รูปที่ 2.11



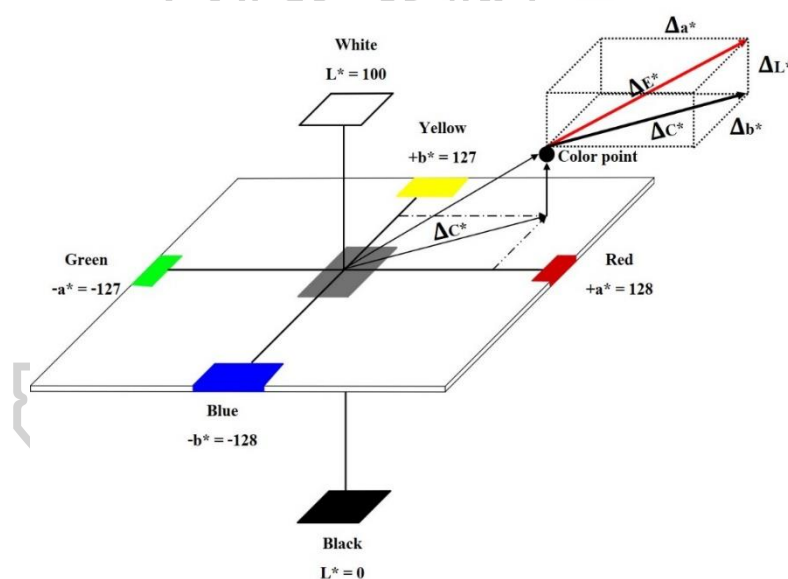
รูปที่ 2.11 ผลลัพธ์ของการแปลงภาพ RGB เป็น Binary

เป็นการแปลงภาพ Gray scale ให้เป็นภาพ ขาว-ดำ โดยกำหนดเงื่อนไขค่าเกณฑ์ (Thresholding) โดยที่ถ้าพิกเซลใดบนภาพ Gray scale ที่มีค่าความสว่างน้อยกว่าค่าเกณฑ์ค่าที่ได้จะเป็นค่า 0 แต่ถ้าพิกเซลใดที่มีค่าความสว่างมากกว่าค่าเกณฑ์ พิกเซลนั้นจะถูกแทนที่ค่าด้วยเลข 1 เป็นไปตามสมการที่ (2.3)

$$binary = \begin{cases} 1 & ; \quad gray\_scale \geq threshold \\ 0 & ; \quad gray\_scale < threshold \end{cases} \quad (2.3)$$

## CIELAB

CIELAB หรือ CIE  $L^*a^*b^*$  เป็นค่าสีที่ถูกพัฒนาโดย CIE ที่ถูกพัฒนามาจากค่าสี CIEXYZ ซึ่ง CIELAB ถูกนำมาใช้เป็นค่ามาตรฐานของระบบสีในปัจจุบัน เนื่องจากมีรูปแบบระดับสีที่มีความใกล้เคียงกับการมองเห็นของตามนุษย์มากที่สุดและง่ายต่อการนำไปใช้และนำไปเปรียบเทียบประกอบด้วยตัวแปร 3 ค่า ได้แก่  $L^*$ ,  $a^*$  และ  $b^*$  โดยที่  $L^*$  เป็นค่าตัวแปรที่แสดงถึงความมืดและสว่างของสี เริ่มจาก  $L^* = 0$  หมายถึงสีดำหรือด้านมืด และไล่ระดับไปจนถึง  $L^* = 100$  หมายถึงสีขาวหรือด้านสว่าง  $a^*$  คือค่าเฉพาะของเฉดสีโดยมีค่าระดับอยู่ที่แปดบิต คือ -128 ถึง 127 โดยที่แกนฝั่ง  $-a^*$  หมายถึงค่าสีเขียว และแกนทางฝั่ง  $+a^*$  หมายถึงค่าสีทางฝั่งสีแดง  $b^*$  คือค่าเฉพาะของเฉดสีเช่นเดียวกับ  $a^*$  แต่จะอยู่บนแกน  $b$  โดยที่  $-b^*$  หมายถึงค่าสีบนฝั่งสีน้ำเงิน และแกนฝั่ง  $+b^*$  หมายถึงค่าของสีเหลือง แสดงดังรูปที่ 2.12 [20, 21]



รูปที่ 2.12 CIELAB color space

ที่มา : “Innovative Tin Electrolyte combining high technical standards with significant cost saving potentials,” by Olaf Kurtz, Peter Köhlkamp, Jürgen Barthelmes, Robert Rüter, Din-Ghee Neoh, and Sia-Wing Kok, 2012, *International Electronics Manufacturing Technology (IEMT)* [20]

จากรูปที่ 2.12 มีค่าตัวแปรเดลต้าที่เกี่ยวข้องในระบบสีนี้ คือ  $\Delta L^*$ ,  $\Delta a^*$  และ  $\Delta b^*$  เป็นผลต่างของของแต่ละแกนบนแกน  $L^*$ ,  $a^*$  และ  $b^*$  โดยค่าเดลต้าเหล่านี้มักจะถูกนำไปใช้เป็นตัวปรับค่าความคลาดเคลื่อนของระบบสี CIELAB โดยที่รูปแบบข้อมูลที่อาจจะต้องทำการแก้ไขคือ

รูปแบบข้อมูลที่มีค่าเดลต้าเกินที่ต้องการหรือค่าเดลต้าเกินค่าความคลาดเคลื่อน ตัวอย่างเช่น ถ้าข้อมูลมีค่า  $\Delta a^*$  เกินกว่าค่าคลาดเคลื่อนจึงจำเป็นที่จะต้องทำการปรับค่าสีแดงหรือสีเขียวเพื่อให้ค่าคลาดเคลื่อนอยู่ในจุดที่ต้องการ

ระบบสี CIELAB นิยมถูกนำไปใช้งานในอุตสาหกรรมสำหรับวัดมาตรฐานสีให้กับวัตถุหรือทางด้านงานวิจัยก็เป็นที่นิยมนำมาใช้ในการเปรียบเทียบกับระบบของรูปภาพกับระบบสีอื่น ๆ ในงานวิจัยการเปลี่ยนระบบสี sRGB มาเป็นระบบสี CIELAB สามารถคำนวณได้จากสมการที่ (2.4) ถึง (2.6) ในกรณีที่ค่าของ  $X/X_n$ ,  $Y/Y_n$  และ  $Z/Z_n$  มีค่ามากกว่า  $\delta^3$

$$L^* = 116\sqrt[3]{Y/Y_n} - 16 \quad (2.4)$$

$$a^* = 500\left(\sqrt[3]{X/X_n} - \sqrt[3]{Y/Y_n}\right) \quad (2.5)$$

$$b^* = 200\left(\sqrt[3]{Y/Y_n} - \sqrt[3]{Z/Z_n}\right) \quad (2.6)$$

ในกรณีที่ค่าของ  $X/X_n$ ,  $Y/Y_n$  และ  $Z/Z_n$  มีค่าน้อยกว่าหรือเท่ากับ  $\delta^3$  จะสามารถคำนวณได้จากสมการที่ (2.7) ถึง (2.9)

$$L^* = 903.3(Y/Y_n) \quad (2.7)$$

$$a^* = 500\left[7.87\left(\frac{X}{X_n} - \frac{Y}{Y_n}\right)\right] \quad (2.8)$$

$$b^* = 200\left[7.87\left(\frac{Y}{Y_n} - \frac{Z}{Z_n}\right)\right] \quad (2.9)$$

โดยที่  $X$ ,  $Y$  และ  $Z$  เป็นค่าที่ได้จากกระนาบสี CIE tristimulus คำนวณหาได้จากสมการที่ (2.10)

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7151 & 0.0721 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.10)$$

$X_n$  และ  $Z_n$  เป็นค่าที่ได้จาก CIE tristimulus ของแสงสว่างซึ่งขึ้นอยู่กับแต่ละมาตรฐานระบบสีที่กำหนด แสดงดัง**ผิดพลาด! ไม่พบแหล่งอ้างอิง** และตารางที่ 2.2 CIE 10 Degree Standard Observe

$$Y_n = 100, \delta = \frac{6}{29}$$

ตารางที่ 2.1 CIE 2 Degree standard observer [21]

<b>CIE 2 Degree standard observer</b>		
<b>Illuminant</b>	$X_n$	$Z_n$
A	109.83	35.55
C	98.04	118.11
D <sub>65</sub>	95.02	108.82
F2	98.09	67.53
TL 4	101.40	65.90
UL 3000	107.99	33.91
D <sub>50</sub>	96.38	82.45
D <sub>60</sub>	95.23	100.86
D <sub>75</sub>	94.96	122.53

ตารางที่ 2.2 CIE 10 Degree Standard Observe [21]

<b>CIE 10 Degree standard observer</b>		
<b>Illuminant</b>	$X_n$	$Z_n$
A	111.16	35.19
C	97.30	116.14
D <sub>65</sub>	94.83	107.38
F2	102.13	69.37
TL 4	103.82	66.90
UL 3000	111.12	35.21
D <sub>50</sub>	96.72	81.45
D <sub>60</sub>	95.21	99.60
D <sub>75</sub>	94.45	120.70

จากสมการข้างต้นเมื่อทำการหาค่าของ  $L^*_{sample}$ ,  $a^*_{sample}$ ,  $b^*_{sample}$ ,  $L^*_{standard}$ ,  $a^*_{standard}$  และ  $b^*_{standard}$  ได้แล้วก็จะนำมาหาผลต่าง  $\Delta L^*$ ,  $\Delta a^*$  และ  $\Delta b^*$  ได้ตามสมการที่ (2.11) ถึง (2.13)

$$\Delta L^* = L^*_{sample} - L^*_{standard} \quad (2.11)$$

$$\Delta a^* = a^*_{sample} - a^*_{standard} \quad (2.12)$$

$$\Delta b^* = b^*_{sample} - b^*_{standard} \quad (2.13)$$

สมการที่ (2.14) เป็นผลรวมของผลต่างทั้งหมด

$$\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (2.14)$$

นอกจากนี้ยังสามารถหาค่าความอิ่มตัวของสี (Saturation/Chroma) ได้จากการหาผลต่างของค่าความอิ่มตัวของสี ( $\Delta C^*$ ) และหาค่าพิสัยของเฉดสี (Hue) ได้จากการหาผลต่างระหว่างมุมตัวอย่างกับมุมมาตรฐาน ( $\Delta H^*$ ) ซึ่งจะอยู่ในรูปแบบพิกัดเชิงขั้ว (Polar coordinate) ได้ตามสมการที่ (2.15) ถึง (2.17)

$$\Delta C^* = C^*_{sample} - C^*_{standard} \quad (2.15)$$

$$C^* = \sqrt{(a^*)^2 + (b^*)^2} \quad (2.16)$$

$$\Delta H^* = \sqrt{(\Delta E^*)^2 - (\Delta L^*)^2 - (\Delta C^*)^2} \quad (2.17)$$

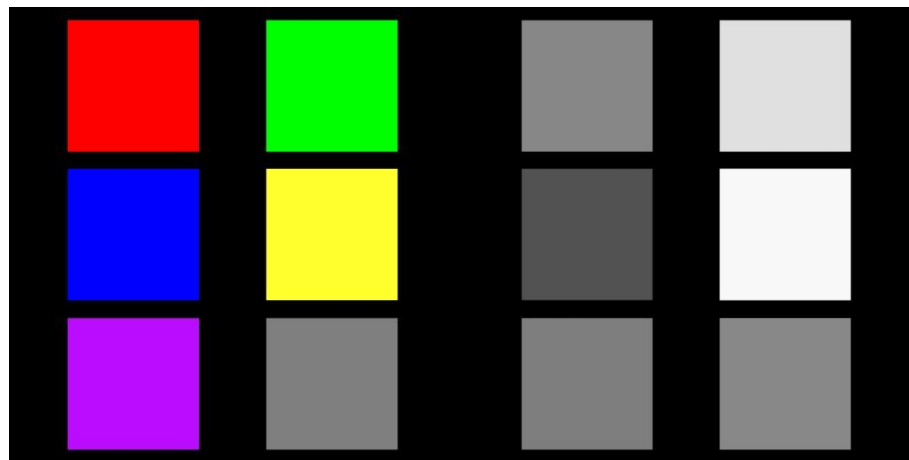
นำสมการที่ (2.14) แทนในสมการที่ (2.17) ได้สมการที่ (2.18)

$$\Delta H^* = \sqrt{\left[ \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \right]^2 - (\Delta L^*)^2 - (\Delta C^*)^2} \quad (2.18)$$

$$\Delta H^* = \sqrt{(\Delta a^*)^2 + (\Delta b^*)^2 - (\Delta C^*)^2} \quad (2.19)$$

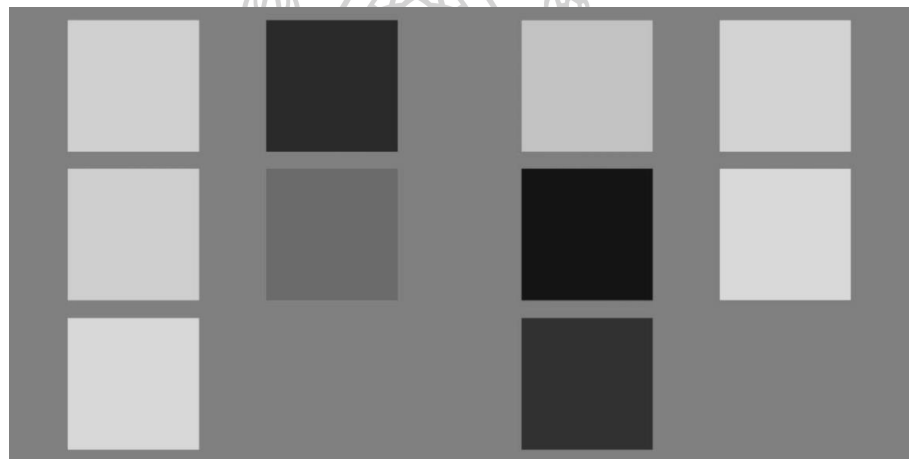
จากวิธีการข้างต้นเมื่อทำการแปลงหรือแยกช่องของสีในย่าน  $L^* a^* b^*$  ออกมาแล้วนั้น มีผลลัพธ์ด้วยกันทั้งหมด 3 รูปแบบ แสดงดังรูปที่ 2.13 LAB





(ก) sRGB channel

(ข) L channel



(ค) a channel

(ง) b channel

รูปที่ 2.13 LAB

จากรูปที่ 2.13 จะเป็นการแสดงถึงองค์ประกอบของย่านสี  $L^*a^*b^*$  ประกอบไปด้วย (ก) แสดงย่านสี sRGB (ข) แสดงย่านองค์ประกอบของความสว่างของสี คือ L channel (ค) แสดงถึงย่านขององค์ประกอบสีแดงกับสีเขียว คือ a channel และ (ง) ย่านองค์ประกอบของสีเหลืองกับสีน้ำเงิน คือ b channel

โดยเมื่อลองพิจารณาแต่ละย่าน L channel จะแสดงถึงองค์ประกอบของความสว่างของสี โดยเมื่อสังเกตจากรูปข้างต้นจะเห็นว่าสีที่ค่อนข้างมีค่าความสว่างอยู่ 2 สีโดยเมื่อนำไปเทียบกับภาพ RGB จะเป็นตำแหน่งของสีเหลืองกับสีเขียวซึ่งเป็นสีที่เมื่อสังเกตด้วยตาจะเห็นว่าค่อนข้างสว่างกว่าสีอื่น แต่จะเห็นได้ว่ามีอยู่หนึ่งสีที่ไม่เกิดการเปลี่ยนแปลงใด ๆ คือสีเทาเพราะเนื่องจากว่าสีเทานั้นเป็นสีที่มีค่าความสว่างอยู่ในช่วงระหว่างสีดำกับสีขาวหรือค่ากลางดังที่แสดงในรูปที่ 2.9



พิจารณาที่ a channel จะเป็นการพิจารณาขององค์ประกอบสีที่เกิดจากสีแดงกับสีเขียว เมื่อสังเกตรูปที่ 2.13(ค) จะเห็นว่าสีเหลืองสองอันข้างบนจะมีอันหนึ่งสว่างอีกอันหนึ่งมืด และเมื่อนำไปเปรียบเทียบกับรูปที่ 2.13(ก) ก็จะเห็นว่า เป็นสีเหลืองของตำแหน่งสีแดงกับสีเขียว โดยจะเห็นว่า สีแดงเมื่อสังเกตใน a channel จะเป็นสีเหลืองที่สว่างเนื่องจากว่าเมื่อดูในแกนของ  $a^*$  จากรูปที่ 2.12 สีแดงจะอยู่ในฝั่งของทาง  $+a^*$  และเมื่อนำค่าที่ได้ไปคำนวณก็จะมีค่า  $L^*$  ที่มากกว่าสีเขียวที่อยู่ทางฝั่ง  $-a^*$  จึงทำให้ที่ a channel สีแดงจะมีองค์ประกอบของค่าความสว่างมากกว่าสีเขียว ในทำนองเดียวกันของ b channel ซึ่งเป็นย่านที่จะพิจารณาขององค์ประกอบสีที่เกิดจากสีเหลืองกับสีน้ำเงิน โดยที่สีเหลืองจะอยู่ในทางฝั่ง  $+b^*$  และสีน้ำเงินจะอยู่ในทางฝั่ง  $-b^*$  บนแกน  $b^*$  และเมื่อสังเกตในรูปที่ 2.13(ง) เทียบกับรูปที่ 2.13(ก) ก็จะเห็นว่าองค์ประกอบของสีเหลืองจะสว่างกว่าสีน้ำเงิน แต่อย่างไรก็ตามจะเห็นว่าทั้งในรูปที่ 2.13(ค) และรูปที่ 2.13(ง) สีเหลืองขององค์ประกอบสีเทานั้นหายไปเพราะเนื่องจากเป็นสีที่มีค่ากลาง (Neutral) ซึ่งค่ากลางนั้นจะไปไม่มีเฉดสีใด เป็นองค์ประกอบอยู่เลย

#### 2.1.4 ซุปเปอร์พิกเซล SLIC

ซุปเปอร์พิกเซลถูกนิยามว่าเป็นลักษณะรูปแบบของการจัดกลุ่มข้อมูลบนพิกเซลที่มีลักษณะคล้าย รูปร่าง ขนาด หรือสี ที่คล้ายกัน โดยเป็นอัลกอริทึมที่ถูกนำมาใช้ในงานทางด้าน Computer vision และ Image processing อย่างแพร่หลาย เช่น งานด้านการแบ่งส่วนภาพ การหาความหมายตัวหนังสือ การตรวจจับและติดตามวัตถุ เป็นต้น

ในการแบ่งส่วนภาพของซุปเปอร์พิกเซลเป็นกระบวนการแบ่งชิ้นส่วนของภาพที่เกิดจากการรวมกันของหลายพิกเซลออกเป็นหลายชิ้นส่วนเพื่ออำนวยความสะดวกในการนำไปวิเคราะห์ กล่าวคือ การแบ่งส่วนภาพคือกระบวนการที่จะกำหนดป้ายให้กับทุกซุปเปอร์พิกเซลที่ถูกแบ่งไว้ เพื่อให้ซุปเปอร์พิกเซลพิกเซลที่มีป้ายกำกับเดียวกันจะมีลักษณะเฉพาะร่วมกันเพื่อการค้นหาพื้นที่และขอบเขตของวัตถุ แสดงดังรูปที่ 2.14



รูปที่ 2.14 ตัวอย่างลักษณะการแบ่งส่วนภาพด้วยซูเปอร์พิกเซล  
ที่มา : “SLIC Superpixels,” *École Polytechnique Fédérale de Lausanne (EPFL)* [22]

จากในรูปที่ 2.14 ความเป็นซูเปอร์พิกเซลนั้นจะพยายามจัดกลุ่มของพิกเซลหลาย ๆ พิกเซลมาอยู่รวมกันโดยพยายามจะทำการจัดกลุ่มให้พิกเซลที่อยู่ใกล้กัน และมีค่าความเข้มสีที่เหมือนกันมากที่สุด

ซูเปอร์พิกเซล SLIC (Simple linear iterative clustering) เป็นชนิดของซูเปอร์พิกเซลหนึ่งที่ทำกรจัดกลุ่มของพิกเซล โดยเปลี่ยนเวกเตอร์สี 3 มิติ ของรูปภาพจาก  $V_{rgb} = [r, g, b]$  เป็น  $V_s = [l, a, b, x, y]$  ขนาด 5 มิติ โดยที่  $[l, a, b]$  เป็นเวกเตอร์ของพิกเซลสีในปริภูมิสี CIELAB และ  $[x, y]$  เป็นเวกเตอร์ตำแหน่งของพิกเซลต่าง ๆ [22]

การทำงานของ SLIC เริ่มโดยการเลือกจำนวนซูเปอร์พิกเซลจำนวน  $K$  ขึ้น โดยสร้างขอบเขตตัวอย่างขึ้นมา หาจุดกึ่งกลางจากสมการ  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  ที่  $k = [1, K]$  หาระยะห่างของจุดกึ่งกลางระหว่างซูเปอร์พิกเซลได้  $S = \sqrt{N/K}$  ที่  $N$  คือจำนวนพิกเซลของรูปภาพ และ  $N/K$  คือขนาดพื้นที่ของซูเปอร์พิกเซล และทำการย้ายตำแหน่งจุดกึ่งกลางของซูเปอร์พิกเซลที่หาได้ใหม่ [14, 23, 24] จากสมการที่ (2.20) ถึง (2.22) โดยจะทำซ้ำไปเรื่อย ๆ จนกระทั่งถึงจุดเกณฑ์ที่ตั้งไว้

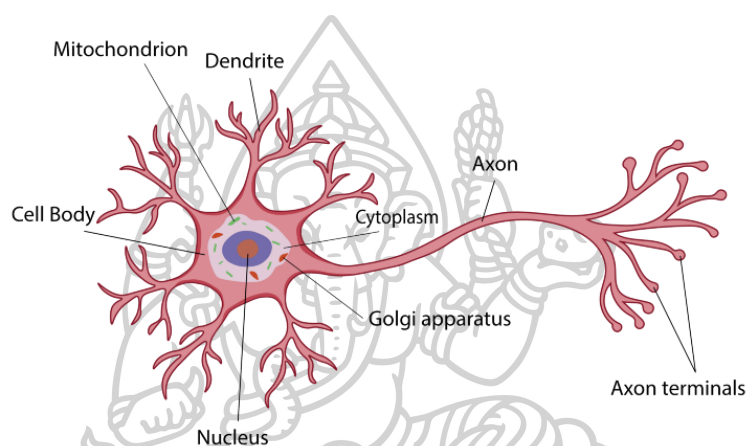
$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad (2.20)$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (2.21)$$

$$D_s = d_{lab} + \left(\frac{m}{S}\right) d_{xy} \quad (2.22)$$

- โดยที่  $d_{lab}$  คือ ระยะทางแบบยูคลิดของปริภูมิสี่ CIELAB
- $d_{xy}$  คือ ระยะทางแบบยูคลิดบนระนาบ  $xy$
- $D_s$  คือ ตำแหน่งจุดกึ่งกลางของซูปเปอร์พิกเซลใหม่
- $m$  คือ ค่าความหนาแน่นของซูปเปอร์พิกเซล เป็นค่าคงที่

### 2.1.5 โครงข่ายประสาทเทียม

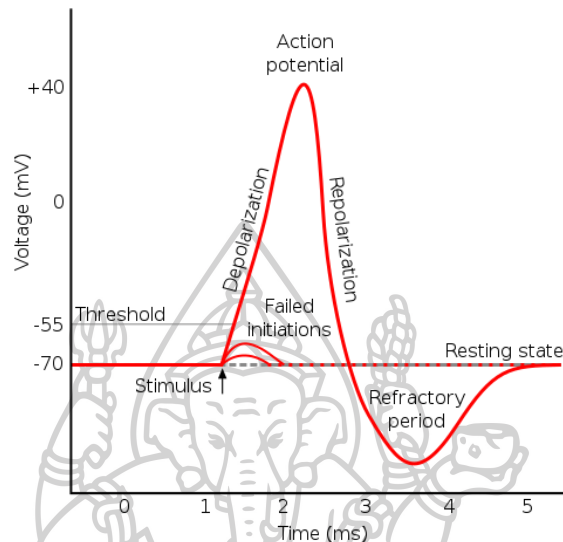


รูปที่ 2.15 เซลล์ประสาท (Neuron)

ที่มา : <https://ichi.pro/th/kar-reiyn-ru-serim-srang-smxng-laea-citwithya-phun-than-thang-prasath-withyasastr-1-141597428538670>

โครงข่ายประสาทเทียม หรือ “Neural network” คืออัลกอริทึมที่จำลองเรียนแบบการทำงานของเซลล์ประสาท (Neuron) ของมนุษย์ดังรูปที่ 2.15 โดยเซลล์ประสาทแต่ละเซลล์จะมีหลักการสื่อสารซึ่งกันและกันจากการถูกกระตุ้นของศักย์ไฟฟ้า เมื่อเซลล์ประสาทถูกกระตุ้นจะเกิดการตอบสนองด้วยการเปลี่ยนความต่างศักย์ ซึ่งจะมีผลลัพธ์เป็น 2 ลักษณะ ได้แก่ ลักษณะแรกจะเกิดกระแสประสาทเมื่อมีแรงกระตุ้นเกินขอบเขตการเร้า (Threshold of excitation) คือ จะมีการสื่อสารต่อกันไปยังเซลล์ประสาทอื่น หรือลักษณะที่สองจะไม่เกิดกระแสประสาทเมื่อแรงกระตุ้นนั้นไม่เกินขอบเขตการเร้า คือ จะไม่มีหรือไม่สามารถติดต่อสื่อสารไปยังเซลล์ประสาทอื่น แสดงดังรูปที่ 2.16 จากรูปข้างต้นจะต้องมีแรงกระตุ้นของศักย์ไฟฟ้าที่มากกว่า  $-50\text{mV}$  ถึงจะมีกระแสประสาทไหลไปยังเซลล์ประสาทอื่น กล่าวคือ เมื่อเซลล์ประสาทเริ่มต้นส่งสัญญาณจากตัวเซลล์ (Cell body) ผ่านแกนประสาท (Axon) ไปยังจุดประสานประสาท (Synapse/Axon terminals) ซึ่งจะเป็นส่วนที่ทำให้

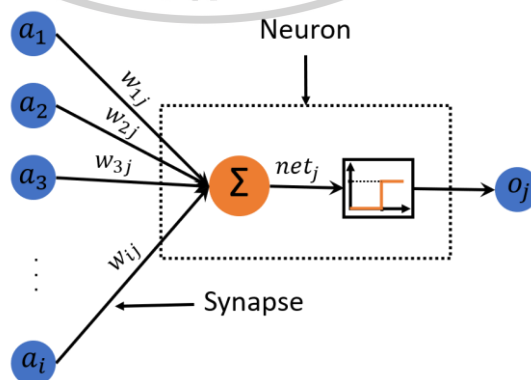
การติดต่อสื่อสารหรือส่งสัญญาณไปยังเซลล์ประสาทอื่นผ่านทางเส้นใยประสาท (Dendrite) ณ จุดที่เซลล์ประสาททำการสื่อสารกันนี้เองที่เป็นหลักการสำคัญที่นำไปประยุกต์ใช้กับโครงข่ายประสาทเทียม [1]



รูปที่ 2.16 ศักยะงาน (Action potential)

ที่มา : [https://th.wikipedia.org/wiki/ไฟล์:Action\\_potential.svg](https://th.wikipedia.org/wiki/ไฟล์:Action_potential.svg)

จะเห็นได้ว่า การเชื่อมต่อสื่อสารของเซลล์ประสาทระหว่างเซลล์เป็นสิ่งสำคัญในการทำงานของโครงข่ายประสาท ขนาดและความแข็งแรงของจุดประสานนั้นจะเปลี่ยนแปลงไปขึ้นอยู่กับประสบการณ์การเรียนรู้ที่ผ่านมาของเซลล์ประสาท ซึ่งโครงข่ายประสาทเทียมได้นำเอาแนวคิดนี้มาประยุกต์ใช้โดยจะเรียกว่า “Synaptic weight” หรือเรียกสั้น ๆ “Weight”



รูปที่ 2.17 แบบจำลองโครงข่ายประสาทเทียม

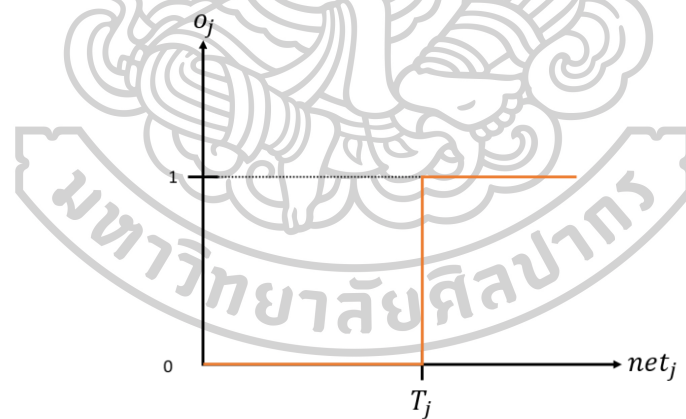
ในรูปที่ 2.17 เป็นรูปแบบโครงสร้างจำลองของโครงข่ายประสาทเทียมที่เซลล์ประสาท หรือนิวรอน (Neuron) แสดงในรูปของปม (Node) และส่วนของการเชื่อมต่อระหว่างเซลล์ประสาท หรือไซแนปส์ (Synapse) แทนด้วยเส้นเชื่อม (Edge) ที่มีค่าน้ำหนัก (Weight) กำกับอยู่ซึ่งก็คือค่าประสพการณ์ของปมนั้น ๆ โดยกำหนดให้  $w_{ij}$  แทนค่าน้ำหนักบนเส้นเชื่อมในทิศทางจากปม  $i$  ไป  $j$  ซึ่งสามารถคำนวณหาค่านำเข้าของข้อมูลรวมไปยังปม  $j$  หรือ  $net_j$  ได้ดังสมการที่ (2.23)

$$net_j = \sum_{i=1}^n w_{ij} a_i \quad (2.23)$$

โดยที่  $a_i$  หมายถึงค่าที่ส่งออกมาจากปมที่  $i$  และจากปมที่  $j$  เมื่อผ่านฟังก์ชันถ่ายโอนชนิดฮาร์ดลิมิตดังรูปที่ 2.18 แล้วผลลัพธ์ของ  $o_j$  จะมีได้สองสถานะตามสมการที่ (2.24) เป็นไปตามลักษณะของการกระตุ้นของเซลล์ประสาท

$$o_j = \begin{cases} 0 & ; \quad net_j < T_j \\ 1 & ; \quad net_j > T_j \end{cases} \quad (2.24)$$

โดยที่  $T_j$  คือ ค่าขีดขั้น (Threshold) หรือค่าขอบเขตการเร้า (Threshold of excitation)



รูปที่ 2.18 ฟังก์ชันถ่ายโอนฮาร์ดลิมิต

ฟังก์ชันถ่ายโอนนั้นถือว่าเป็นตัวกำหนดผลลัพธ์ที่จะออกไปจากโครงข่ายประสาทนั้น ๆ โดยในการเลือกใช้ชนิดของฟังก์ชันถ่ายโอนนั้นก็จะขึ้นอยู่กับประเภทหรือชนิดของข้อมูลที่นำมาใช้ โดยชนิดของฟังก์ชันถ่ายโอนนั้นมีด้วยกันหลายรูปแบบดังแสดงในตารางที่ 2.3

ตารางที่ 2.3 ตารางแสดงชนิดของฟังก์ชันถ่ายโอน (Types of transfer function)

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = \max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

Copyright © Sebastian Raschka 2016  
(<http://sebastianraschka.com>)

### โครงข่ายประสาทเทียมแบบสังวัตนาการ (CNN)

โครงข่ายประสาทเทียมแบบสังวัตนาการ (Convolutional neural network) หรือที่เรียกกันว่า “CNN” คือ ประเภทของโครงข่ายประสาทเทียมชนิดหนึ่งที่ถูกนำมาพัฒนาและใช้งานทางด้านการประมวลผลรูปภาพ (Image processing) คอมพิวเตอร์วิทัศน์ (Computer vision) เพราะสามารถให้ผลของการจำแนก (Classification) ประเภทได้ดีกับชนิดข้อมูลที่เป็นรูปภาพ เช่น การรู้จำรูปแบบตัวอักษร การรู้จำลายมือเขียน เป็นต้น

แนวคิดหลักของโครงข่ายประสาทเทียมประเภทนี้มาจากกระบวนการทำสังวัตนาการ หรือ Convolution ในด้านการประมวลผลภาพ กำหนดให้  $I$  เป็นภาพนำเข้า (Input image) ขนาด



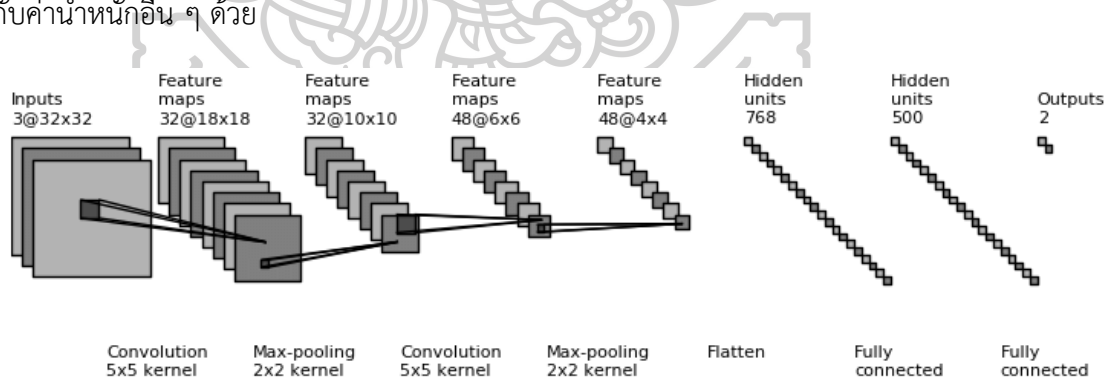
$w \times w$  พิกเซล และ  $K$  เป็นภาพต้นแบบ (Kernel) ขนาด  $(2m+1) \times (2m+1)$  พิกเซล โดยมีตัวปฏิบัติการทางรูปภาพที่จะทำหน้าที่แปลงภาพนำเข้า  $I$  เป็นภาพใหม่อยู่ 2 ประเภทคือ สหสัมพันธ์ (Cross-correlation :  $O_{cc}$ ) และ สัมพัทธ์ (Convolution :  $O_{cv}$ ) โดยมีขนาดภาพใหม่ที่  $(w-2m) \times (w-2m)$  พิกเซล [25] คำนวณได้จากสมการที่ (2.25) และ (2.26)

$$O_{cc}[x, y] = f \left( \sum_{i=-m}^m \sum_{j=-m}^m I[x+m+i, y+m+j] K[m+i, m+j] \right) \quad (2.25)$$

$$O_{cv}[x, y] = f \left( \sum_{i=-m}^m \sum_{j=-m}^m I[x+m+i, y+m+j] K[m-i, m-j] \right) \quad (2.26)$$

เมื่อ  $f(\cdot)$  คือฟังก์ชันของการแปลง เช่น Tanh, Sigmoid หรือ Relu

ในงานประมวลผลภาพนั้นวิธีการทำ Cross-correlation หรือ Convolution จะให้ค่าผลลัพธ์ที่สูงเมื่อบริเวณภาพ  $I$  นั้นมีลักษณะคล้ายนั้นมีลักษณะคล้ายกับภาพต้นแบบ  $K$  นั่นคือเราสามารถพิจารณาภาพต้นแบบเหล่านี้ว่ามีลักษณะเด่นได้อีกหลาย ๆ รูปแบบและใช้กระบวนการทั้งสองในการตรวจจับตำแหน่งในภาพ  $I$  บนบริเวณที่มีลักษณะเด่นแต่จะต้องกำหนดภาพต้นแบบขึ้นมาก่อนที่จะเริ่มสกัดคุณลักษณะเด่นต่าง ๆ ออกมา แต่ในโครงข่ายประสาทเทียมแบบสัมพัทธ์ค่าต่าง ๆ บนภาพต้นแบบเหล่านี้ถือเป็นค่าน้ำหนักของการเรียนรู้ของโครงข่ายที่จะถูกปรับไปพร้อม ๆ กับค่าน้ำหนักอื่น ๆ ด้วย



รูปที่ 2.19 โครงสร้างโครงข่ายประสาทเทียมแบบสัมพัทธ์ (Convolutional neural network)

ที่มา : “Nuclei Recognition Using Convolutional Neural Network and Hough Transform,” by M. Zejmo, M. Kowal, J. Korbicz and R. Monczak, 2018, International Conference on Diagnostics of Processes and Systems [26]



โครงสร้างของ CNN ที่ใช้งานประกอบด้วยชั้นสังวัตนาการ (Convolution layer) และชั้นการลดขนาด (Pooling) สลับกันตามด้วยชั้น Fully connected (คล้ายกับการทำ MLP) [27] แสดงดังรูปที่ 2.19 ชั้นการลดขนาดนั้นมีจุดประสงค์เพื่อลดอัตราความคลาดเคลื่อนที่เกี่ยวข้องกับตำแหน่งของลักษณะเด่นบนรูปภาพนำเข้า และภาพที่ได้จากการทำสังวัตนาการแล้วนั้นจะเรียกว่า ภาพลักษณะเด่น (Feature map) โดยในแต่ละชั้นนั้นจะประกอบไปด้วยภาพลักษณะเด่นได้หลายภาพนั้นคือ เมื่อชั้นสังวัตนาการอยู่ชั้น  $l$  ต่อกับภาพลักษณะเด่นทุกภาพในชั้น  $l-1$ ,  $S_k^{l-1}$ ,  $k = 1, 2, 3, \dots, n$  โดยในการต่อระหว่างภาพต้นแบบที่  $t$  ในชั้น  $l$  กับภาพลักษณะเด่นที่  $k$  ในชั้นที่  $l-1$  จะใช้ภาพต้นแบบที่แตกต่างกันออกไป ในการทำงานของชั้นสังวัตนาการที่  $l$  เพื่อคำนวณภาพที่  $t$  นั้นทำได้โดยสมการที่ (2.27) [25]

$$S_t^l[x, y] = f \left( \sum_{k=1}^n \sum_{i,j} S_k^{l-1}[x+m+i, y+m+j] K_{k,t}^l[m-i, m-j] \right) \quad (2.27)$$

จากที่ได้กล่าวมาข้างต้นนั้นการทำ Convolution หรือ Cross-correlation หลักการหรือตัวการสำคัญในการที่จะทำหน้าที่สกัดลักษณะเด่นออกมานั้นคือ ภาพต้นแบบ (Kernel) มีลักษณะตัวอย่างดังรูปที่ 2.20 และเมื่อนำไปสกัดลักษณะเด่นของภาพนำเข้าแล้วจะมี 2 วิธีที่นิยมทำในการสกัดคุณลักษณะ คือ การสกัดคุณลักษณะปกติแบบไม่เพิ่มขอบ กับการสกัดคุณลักษณะแบบเพิ่มขอบ (Padding)

0	1	0
1	1	1
0	1	0

Kernel size 3x3

รูปที่ 2.20 ตัวอย่างลักษณะของภาพต้นแบบ (Kernel)

ในการสกัดคุณลักษณะแบบไม่เพิ่มขอบจะเป็นการสกัดคุณลักษณะเด่นของภาพนำเข้าโดยการนำภาพต้นแบบมาวิ่งบนภาพนำเข้าในรูปที่ 2.21 และคำนวณหาค่า Convolution หรือ Cross-correlation โดยขยับทีละ 1 ครั้ง (Stride = 1) แสดงดังรูปที่ 2.22 และได้ผลลัพธ์ของภาพลักษณะเด่นใหม่เพื่อนำไปใช้กับภาพต้นแบบในการสกัดลักษณะในชั้นถัดไปดังรูปที่ 2.23

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Input image Size 7x7

รูปที่ 2.21 ข้อมูลภาพนำเข้า (Input image no padding)

0x1	1x0	0x0	1	1	0	0
1x1	1x1	1x1	1	0	0	1
0x0	1x1	0x0	1	1	0	0
0	0	0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 1

1	0x0	1x0	0x1	1	0	0
1	1x1	1x1	1x1	0	0	1
0	0x1	1x0	0x1	1	0	0
0	0	0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 2

1	0	0x0	1x1	0x1	0	0
1	1	1x1	1x1	1x0	0	1
0	1	0x0	1x1	0x1	0	0
0	0	0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 3

1	0	0	0x1	1x1	0x0	0
1	1	1	1x1	1x0	1x0	1
0	1	0	0x1	1x1	0x0	0
0	0	0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 4

1	0	0	1	0x1	1x0	0x0
1	1	1	1	1x0	1x0	1x1
0	1	0	1	0x1	1x0	0x0
0	0	0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 5

1	0	0	1	1	0	0
0x1	1x1	0x1	1	0	0	1
1x0	1x1	1x0	1	1	0	0
0x0	1x0	0x0	1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 6

1	0	0	1	1	0	0
1	0x1	1x1	0x1	0	0	1
0	1x1	1x0	1x1	1	0	0
0	0x0	1x0	0x1	0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 7

1	0	0	1	1	0	0
1	1	0x1	1x1	0x0	0	1
0	1	1x0	1x1	1x1	0	0
0	0	0x0	1x1	0x0	0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 8

1	0	0	1	1	0	0
1	1	1	0x1	1x0	0x0	1
0	1	0	1x1	1x1	1x0	0
0	0	0	0x1	1x0	0x0	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 9

1	0	0	1	1	0	0
1	1	1	1	0x0	1x0	0x1
0	1	0	1	1x1	1x0	1x0
0	0	0	1	0x0	1x0	0x1
1	1	0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 10

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0x0	1x1	0x0	1	1	0	0
1x0	1x0	1x0	1	0	0	1
0x1	1x1	0x0	1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 11

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	0x1	1x0	0x1	1	0	0
0	1x0	1x0	1x1	0	0	1
1	0x1	1x0	0x1	0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 12

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0x0	1x1	0x1	0	0
0	0	1x0	1x1	1x0	0	1
1	1	0x0	1x1	0x0	0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 13

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	0x1	1x1	0x0	0
0	0	0	1x1	1x0	1x0	1
1	1	0	0x1	1x0	0x0	0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 14

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	0x1	1x0	0x0
0	0	0	1	1x0	1x0	1x1
1	1	0	1	0x0	1x0	0x0
1	1	0	1	0	0	1
0	0	0	1	1	0	0

Step 15

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0x0	1x0	0x0	1	0	0	1
1x1	1x1	1x0	1	0	0	0
0x1	1x1	0x0	1	0	0	1
0	0	0	1	1	0	0

Step 16

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0x0	1x0	0x1	0	0	1
1	1x1	1x0	1x1	0	0	0
1	0x1	1x0	0x1	0	0	1
0	0	0	1	1	0	0

Step 17

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0x0	1x1	0x0	0	1
1	1	1x0	1x1	1x0	0	0
1	1	0x0	1x1	0x0	0	1
0	0	0	1	1	0	0

Step 18

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	0x1	1x0	0x0	1
1	1	0	1x1	1x0	1x0	0
1	1	0	0x1	1x0	0x0	1
0	0	0	1	1	0	0

Step 19

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0x0	1x0	0x1
1	1	0	1	1x0	1x0	1x0
1	1	0	1	0x0	1x0	0x1
0	0	0	1	1	0	0

Step 20

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0	0	1
0x1	1x1	0x0	1	0	0	0
1x1	1x1	1x0	1	0	0	1
0x0	1x0	0x0	1	1	0	0

Step 21

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0	0	1
1	0x1	1x0	0x1	0	0	0
1	1x1	1x0	1x1	0	0	1
0	0x0	1x0	0x1	1	0	0

Step 22

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0	0	1
1	1	0x0	1x1	0x0	0	0
1	1	1x0	1x1	1x0	0	1
0	0	0x0	1x1	0x1	0	0

Step 23

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0	0	1
1	1	0	0x1	1x0	0x0	0
1	1	0	1x1	1x0	1x0	1
0	0	0	0x1	1x1	0x0	0

Step 24

1	0	0	1	1	0	0
1	1	1	1	0	0	1
0	1	0	1	1	0	0
0	0	0	1	0	0	1
1	1	0	1	0x0	1x0	0x0
1	1	0	1	1x0	1x0	1x1
0	0	0	1	0x1	1x0	0x0

Step 25

รูปที่ 2.22 Cross-correlation (stride = 1)

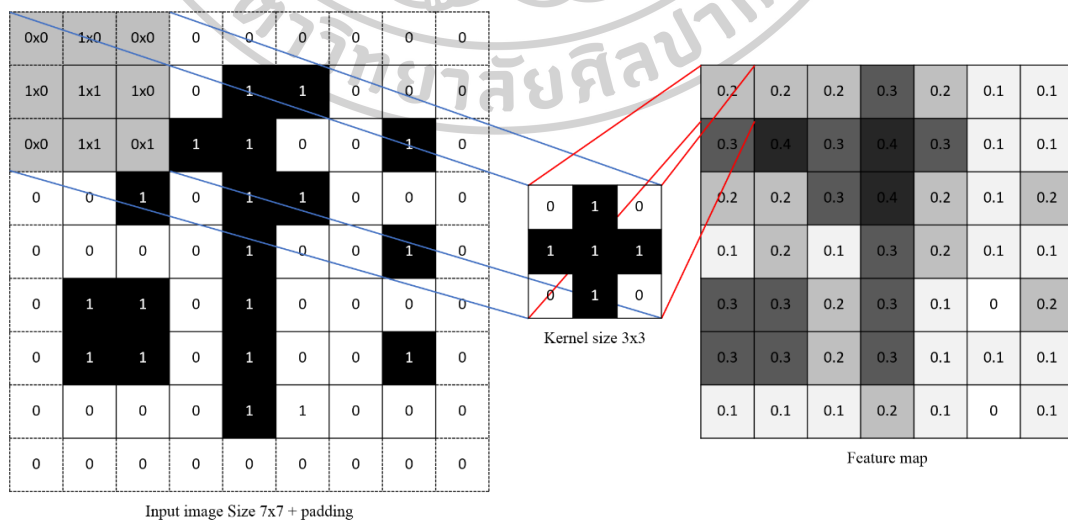
0.4	0.3	0.4	0.3	0.1
0.2	0.3	0.4	0.2	0.1
0.2	0.1	0.3	0.2	0.1
0.3	0.2	0.3	0.1	0
0.3	0.2	0.3	0.1	0.1

Feature map

รูปที่ 2.23 Feature map

จากการทำ Convolution (Cross-correlation) ในตัวอย่างข้างต้นกรณีที่ไม่มีการเพิ่มขอบหรือ Padding จะเห็นได้ว่าภาพลักษณะเด่น (Feature map) ที่ได้ออกมานั้นจะมีขนาดเล็กกว่าเท่ากับ  $(w - 2m) \times (w - 2m) = (7 - 2m) \times (7 - 2m) = 5$  ที่  $m = 1$  เท่ากับว่าภาพลักษณะเด่นที่ได้จะมีขนาดเท่ากับ  $5 \times 5$  พิกเซล ตามรูปที่ 2.23 จะเห็นได้ว่าภาพที่ได้จากการสกัดลักษณะนั้นจะมีลักษณะเด่นของจุดพิกเซลที่คล้ายกับตำแหน่งพิกเซลของภาพนำเข้าและพิกเซลของภาพต้นแบบมากที่สุด

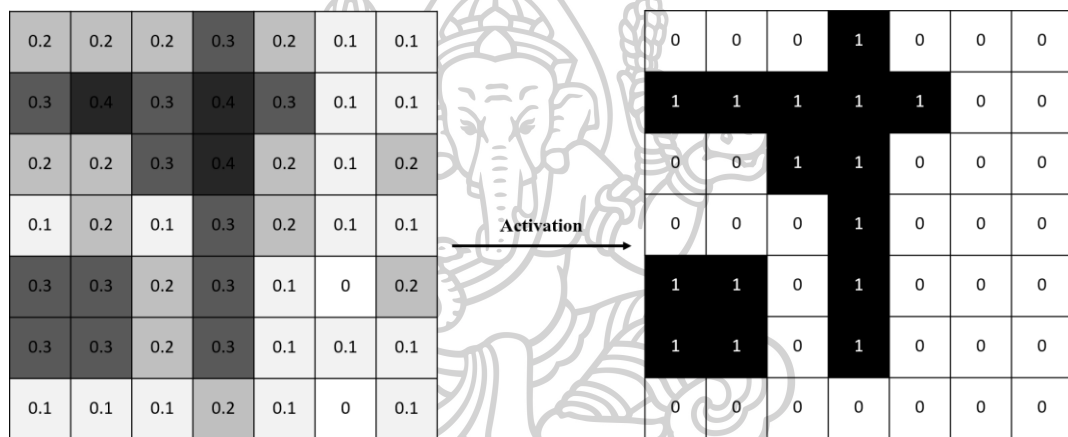
ในการสกัดคุณลักษณะแบบเพิ่มขอบ หรือ Padding นั้นคือการเพิ่มขอบให้กับภาพนำเข้าก่อนที่จะนำไปทำ Convolution โดยทั่วไปแล้วจะนิยมเพิ่มขอบด้วยค่า 0 หรือขอบดำเข้าไปโดยจะมีวิธีหรือขั้นตอนการสกัดลักษณะแบบเดียวกับไม่ทำ Padding ทุกประการ แสดงดังรูปที่ 2.24



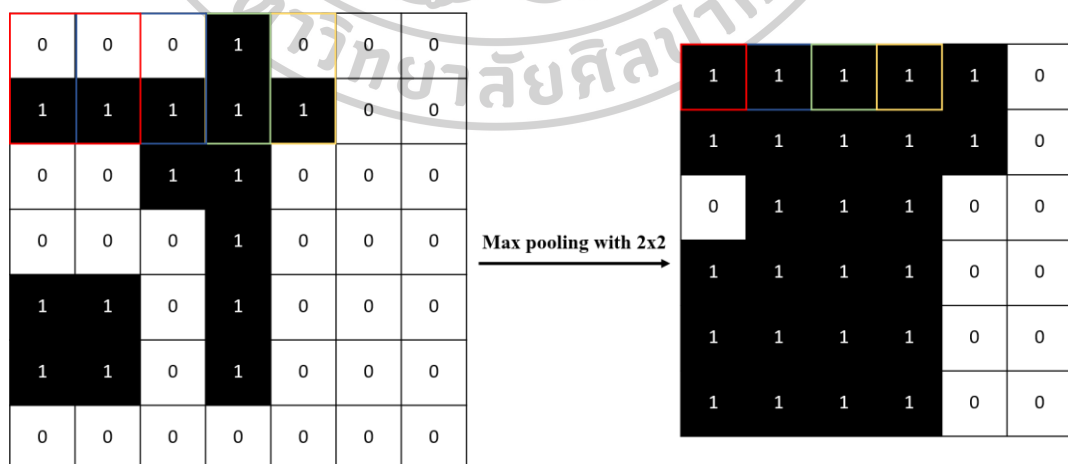
รูปที่ 2.24 Cross-correlation (Stride = 1 and zero padding)

ผลลัพธ์ของภาพลักษณะเด่นที่ได้จากการทำ Cross-correlation กับภาพนำเข้าที่มีการทำ Padding แล้วนั้นจะมีขนาดที่เท่ากับภาพเดิม คือ  $7 \times 7$  และยังให้ผลลัพธ์ของการสกัดลักษณะเด่นออกมาที่มีความละเอียดที่มากกว่าหรือคล้ายกับแบบไม่ทำการ Padding

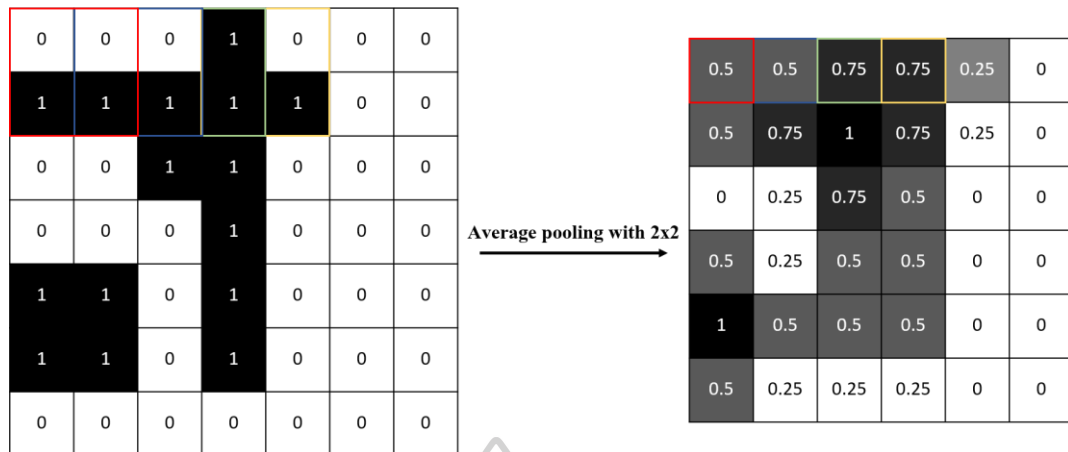
ในโครงสร้างของ CNN นั้นเมื่อภาพนำเข้ามากระทำการสังวัตนาการหรือ Convolution และผ่านการทำฟังก์ชันการแปลงมาแล้ว ดังรูปที่ 2.25 ขั้นต่อไปจะนำข้อมูลลักษณะเด่นที่ได้ไปลดขนาด (Pooling) เพื่อให้ข้อมูลนั้นมีความคลาดเคลื่อนของตำแหน่งลักษณะเด่นน้อยลง โดยจะมี 2 วิธีที่นำมาใช้ คือ การลดขนาดโดยเลือกค่าสูงสุดบนพื้นที่ที่ Kernel คลอบคลุมอยู่ (Max pooling) ดังรูปที่ 2.26 และการลดขนาดโดยใช้ค่าเฉลี่ย (Average pooling) ดังรูปที่ 2.27



รูปที่ 2.25 ขั้นตอนการแปลงข้อมูลผ่านฟังก์ชันการแปลง

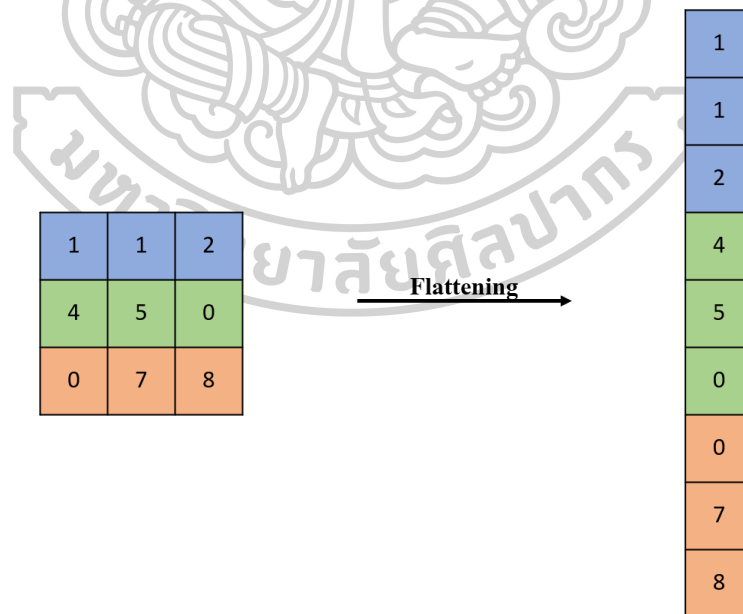


รูปที่ 2.26 Max pooling



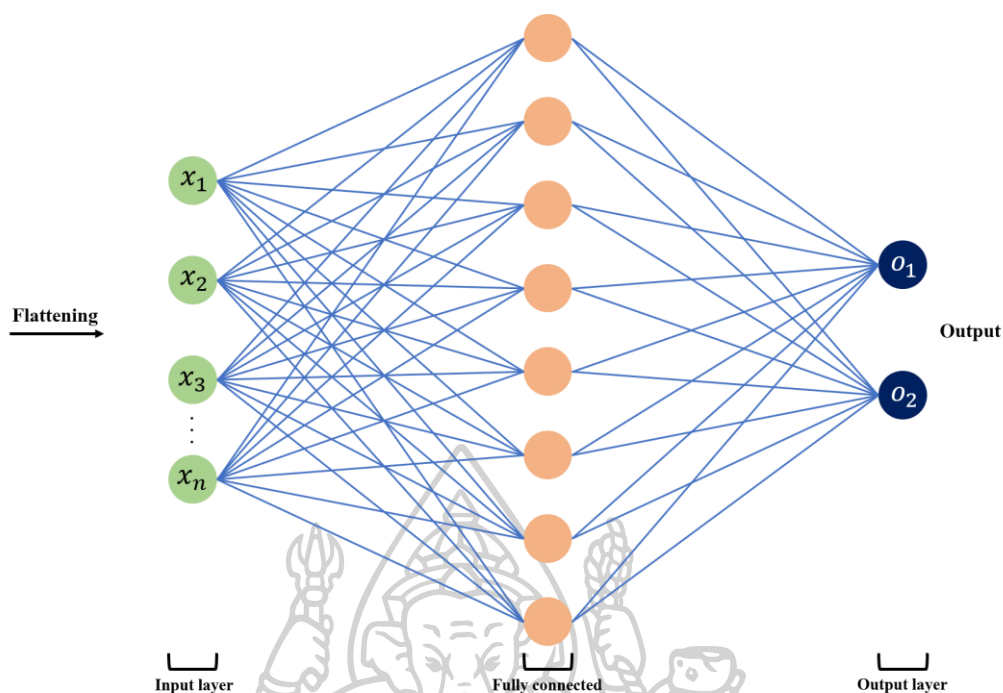
รูปที่ 2.27 Average pooling

ถัดจากขั้นของทำการ Cross-correlation ก็กับการลดขนาด Pooling มาแล้วจำเป็นที่จะต้องเปลี่ยนข้อมูลให้มีขนาด 1 มิติ ด้วยการทำให้ Flatten ดังรูปที่ 2.28 เพื่อให้สามารถนำข้อมูลมาใช้ในการทำ Fully connected หรือ Dense [28, 29] โดยจะมีการทำงานที่คล้ายกับเพอร์เซพตรอนหลายชั้นโดยถือว่าเป็นโครงข่ายประสาทขั้นสุดท้ายของการเรียนรู้ โดยที่ขั้นสุดท้ายของ Fully connected จะมีจำนวนโหนดเท่ากับจำนวนข้อมูลออกที่ต้องการ (Class) แสดงดังตัวอย่างรูปที่ 2.29



รูปที่ 2.28 Flattening layer





รูปที่ 2.29 Fully connected

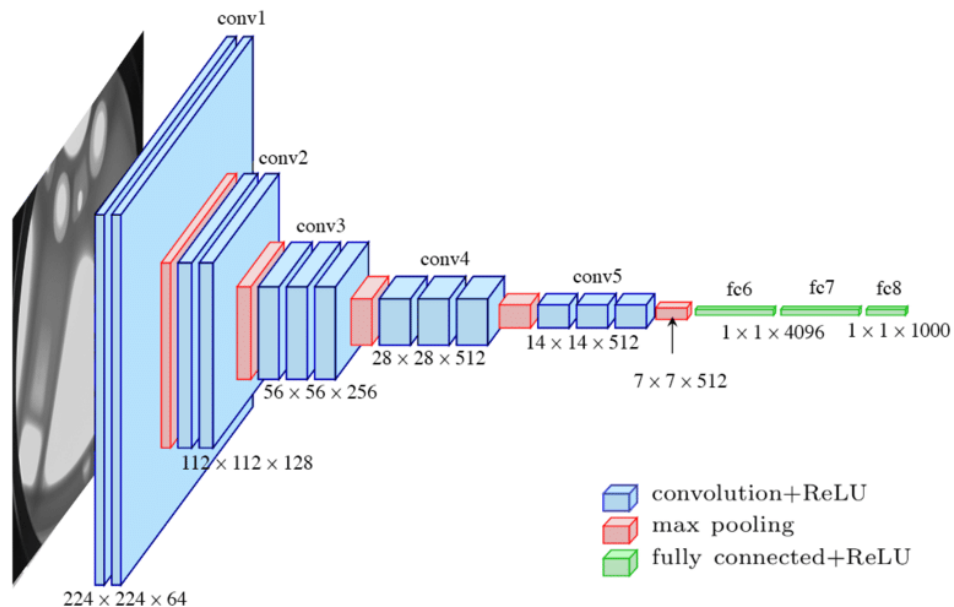
### โครงข่ายประสาทเทียบชนิดอื่น ๆ

#### VGG

VGG คือชื่อเรียกของโครงสร้างทางสถาปัตยกรรมการเรียนรู้เชิงลึกรูปแบบหนึ่งที่ได้ถูกพัฒนามาจากโครงสร้างของ CNN ให้มีจำนวนชั้นของ Hidden layer ที่ลึกขึ้นสำหรับนำไปประยุกต์ใช้กับการรู้จำรูปภาพที่มีขนาดใหญ่และให้ค่าความแม่นยำมากกว่าโมเดล CNN วิธีการนี้ได้ถูกนำเสนอโดย K. Simonyan และ A. Zisserman แห่งมหาวิทยาลัยออกซ์ฟอร์ด ในบทความวิจัยที่ชื่อว่า “Very Deep Convolutional Networks for Large-Scale Image Recognition” [30] โดยจุดมุ่งหมายของงานวิจัยนี้คือการสร้างโมเดลที่มีจำนวนชั้นของ Hidden layer ที่เพิ่มขึ้นและนำมาเปรียบเทียบกับโมเดลอื่น ๆ ที่มีลักษณะเดียวกัน จะเรียกโมเดลกลุ่มนี้ว่า ConvNets โดยชื่อ VGG เป็นชื่อที่มาจากชื่อกลุ่มของนักวิจัย Visual Geometry Group จึงตั้งชื่อว่า VGG โดยโมเดล VGG ที่กลุ่มนักวิจัยสร้างขึ้นนั้นมีอยู่ด้วยกัน 2 โมเดล คือ VGG16 และ VGG19 ซึ่งตัวเลข 16 และ 19 หลัง VGG จะบ่งบอกถึงจำนวนชั้นของ Hidden layer ของโครงสร้าง VGG นั้น ๆ [31-34]

VGG16 เป็นโมเดลที่ถูกพัฒนามาเป็นโมเดลแรกของกลุ่มโมเดล VGG ซึ่งได้พัฒนามาจากโมเดล CNN โดยที่ตัวโมเดล VGG16 นั้นเป็นโมเดลที่มีจำนวนชั้น Hidden layer ที่ 16 ชั้น ประกอบ

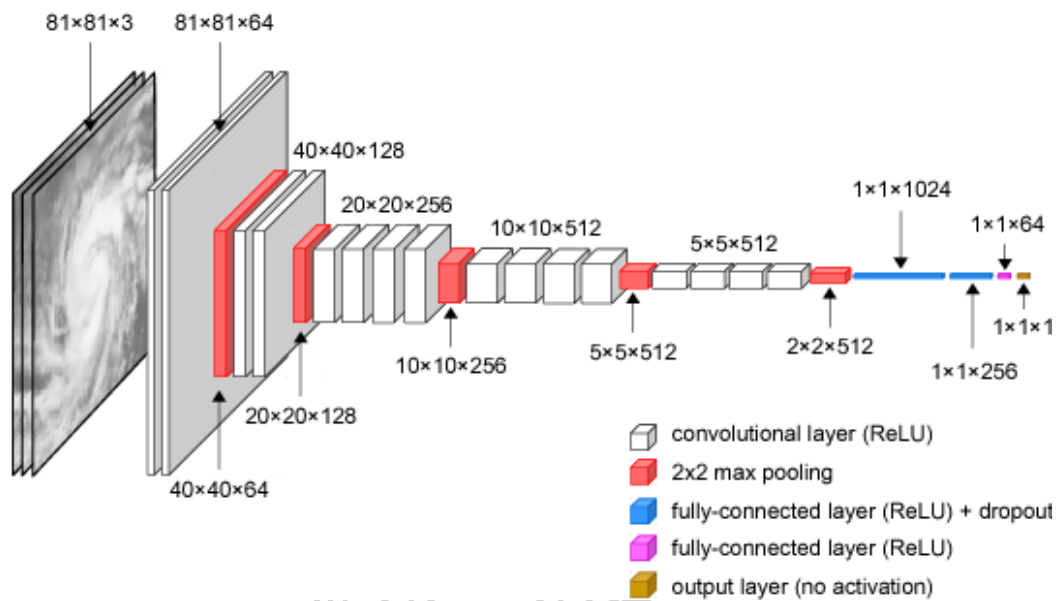
ไปด้วยชั้นของการทำ Convolution หรือ Cross-correlation 13 ชั้น และชั้นของการทำ Fully-connected 3 ชั้น รวมกันเป็น 16 ชั้น ของ Hidden layer โดยที่แต่ละชั้นก็จะมียังองค์ประกอบภายในที่แตกต่างกันแสดงดังรูปที่ 2.30



รูปที่ 2.30 โครงสร้างโมเดล VGG16

ที่มา : “Automatic localization of casting defects with convolutional neural networks,”  
by Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee and Kincho H. Law, 2017, *IEEE International Conference on Big Data (BIGDATA)*, 1735 [35]

VGG19 เป็นโมเดลที่ถูกพัฒนาขึ้นมาพร้อมกับโมเดล VGG16 โดยมีโครงสร้างและการทำงานที่คล้ายกันโดยจะมีความแตกต่างกันที่จำนวนชั้นของ Hidden layer ที่มากขึ้น และองค์ประกอบภายในที่แตกต่างออกไป แสดงดังรูปที่ 2.31 องค์ประกอบของโมเดล VGG19 นั้นประกอบด้วยชั้นของการทำ Convolution หรือ Cross-correlation จำนวน 16 ชั้น และชั้น Fully connected จำนวน 3 ชั้น โดยในชั้น Fully connected ของโมเดลนี้นั้นมีความแตกต่างจากโมเดล VGG16 อยู่ตรงที่มีการนำเอาการทำ dropout มาใช้ในส่วนของหลังการทำ Fully connected ในชั้นแรกซึ่งจะเป็นการลดจำนวนพารามิเตอร์ที่จะนำมาใช้ในการคำนวณของโครงข่ายประสาทในชั้น Fully connected ให้มีจำนวนไม่มากเกินไปจึงช่วยลดปัญหาการเกิด Overfitting ของข้อมูลได้



รูปที่ 2.31 โครงสร้างโมเดล VGG19

ที่มา : “A Convolutional Neural Network Approach for Estimating Tropical Cyclone Intensity Using Satellite-based Infrared Images,” by J. S. Combinido, John Robert Mendoza and Jeffrey A. Aborot, 2018, *International Conference on Pattern Recognition (ICPR)*, 24, 1476 [36]

### MobileNet

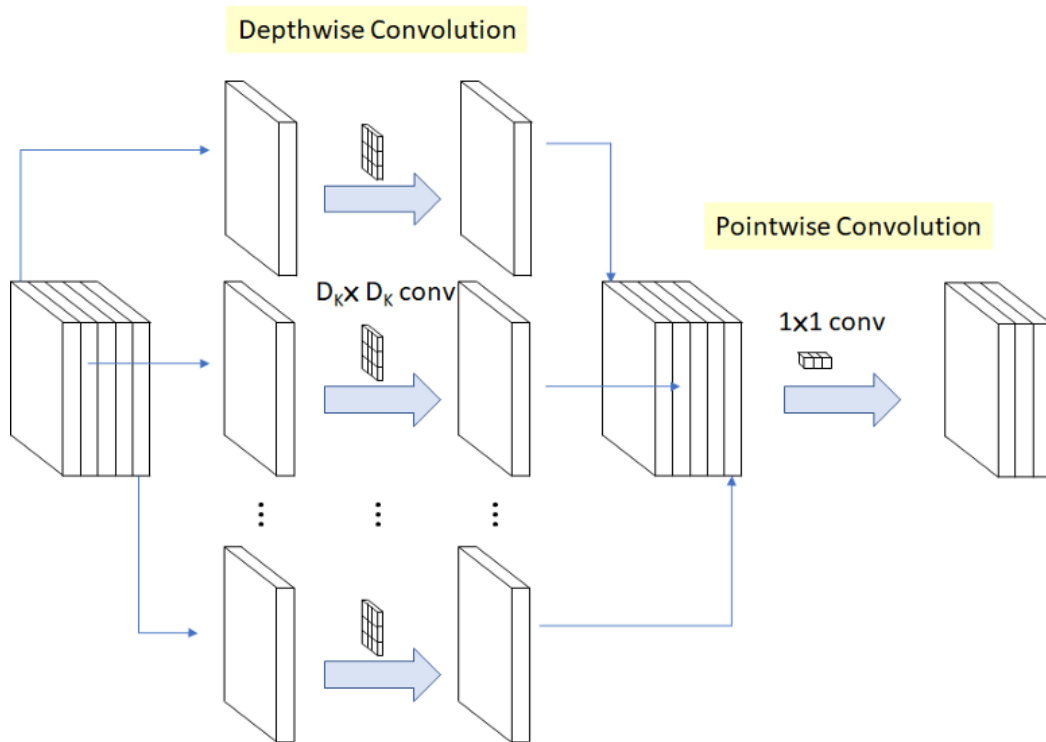
MobileNet เป็นโมเดลการเรียนรู้เชิงลึกที่ถูกสร้างขึ้นมาให้สามารถใช้ได้บนแอปพลิเคชัน โทรศัพท์มือถือหรือสมาร์ทโฟน และยังเป็นโมเดลทางด้าน Mobile Computer Vision โมเดลแรกของ TensorFlow โดยหลักการการทำงานของ MobileNet จะใช้การทำ Convolution หรือ Cross-correlation แบบแบ่งแยกขนาดออกไปเป็นเชิงลึก (Depthwise Separable Convolution) ซึ่งวิธีการนี้จะทำให้สามารถลดจำนวนของพารามิเตอร์ลงได้อย่างมากเมื่อนำไปเปรียบเทียบกับโครงข่ายหรือโมเดลอื่น ๆ ที่มีการทำ Convolution แบบปกติที่มีความลึกเท่ากัน ผลลัพธ์ที่ได้จึงทำให้โมเดล MobileNet นี้มีค่าน้ำหนักของพารามิเตอร์ที่ค่อนข้างน้อย โดยในการทำ Convolution แบบขนานนี้ เกิดจากการทำงานอยู่ 2 วิธีด้วยกัน คือ การ Convolution เชิงลึก (Depthwise convolution) และการ Convolution แบบจุด (Pointwise convolution) [27, 37-39]

MobileNet ถือเป็นชนิดของ CNN ชนิดหนึ่งที่ถูกพัฒนาขึ้นโดย Google ความสามารถของมันคือการฝึกฝนให้จำแนกวัตถุที่มีขนาดเล็กได้อย่างยอดเยี่ยมและรวดเร็ว โดยโครงสร้างโมเดลของ MobileNet แสดงดังตารางที่ 2.4

ตารางที่ 2.4 โครงสร้างโมเดล MobileNet

Type / Stride	Filter Shape	Input Size
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
5 x Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 1000	1 x 1 x 1024
SoftMax / s1	Classifier	1 x 1 x 1000

จากที่กล่าวมาข้างต้นว่าโมเดล MobileNet เป็นโมเดลที่มีวิธีการเรียนรู้แบบขนานเชิงลึกคือ เกิดจากการทำ Depthwise convolution และตามด้วยการทำ Pointwise convolution แสดงดังรูปที่ 2.32



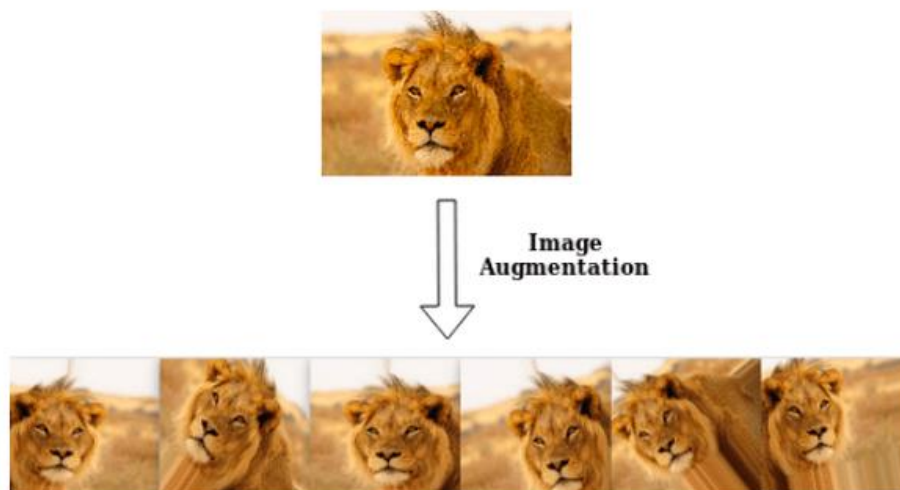
รูปที่ 2.32 โครงสร้างการทำ Depthwise Separable Convolution

ที่มา : <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>

จากตัวอย่างรูปที่ 2.32 ในขั้นตอนของการทำ Depthwise convolution จะเป็นการทำ Convolution เชิงพื้นที่ของแต่ละ Channel โดยจากภาพจะมี 5 Channels ซึ่งก็หมายความว่ามีการทำ Convolution ขนาด  $5 \times (D_k \times D_k)$  แล้วนำมาทำ Convolution ต่อในขั้น Pointwise convolution ซึ่งเป็นขั้นที่มีขนาดของตัว Convolution เป็น  $1 \times 1$  ซึ่งทำหน้าที่ในการเปลี่ยนรูปร่างมิติของข้อมูล โดยจากที่กล่าวมาเป็นการทำงานของ Convolution แค่หนึ่งตัวเท่านั้น

### 2.1.6 การเพิ่มจำนวนข้อมูล

การเพิ่มจำนวนข้อมูล หรือ “Images augmentation” แสดงตัวอย่างรูปที่ 2.33 คือการเพิ่มจำนวนหรือขยายรูปแบบของข้อมูลเพื่อให้มีจำนวนข้อมูลที่มากขึ้น หลากหลาย และเพียงพอต่อการนำไปใช้งาน



รูปที่ 2.33 ลักษณะของการเพิ่มจำนวนข้อมูล “Images augmentation”

ที่มา : <https://www.quora.com/What-is-data-augmentation-in-CNN>

ในงานทางด้าน Machine learning หรือ Deep learning การนำเอาภาพมาใช้ฝึกฝนเป็นวิธีการหนึ่งที่นิยมใช้ในงานด้านรู้จำตรวจจับวัตถุ ข้อมูลนั้นจึงถือว่ามีค่าอย่างมากในการนำมาใช้ และโดยปกติแล้วในการฝึกฝนโมเดลให้ Machine learning หรือ Deep learning นั้นยังมีจำนวนและความหลากหลายของข้อมูลมากเท่าใดก็จะยิ่งเพิ่มประสิทธิภาพของการเรียนรู้ให้มีความแม่นยำยิ่งขึ้นจากการฝึกฝนข้อมูลที่หลากหลายมากขึ้น และก็ยังช่วยลดอัตราการเกิด Underfitting และ Overfitting ให้กับการฝึกสอนและทดสอบของชุดข้อมูลได้ แต่อย่างไรก็ตามในการเพิ่มจำนวนของข้อมูลภาพที่มากเกินไปนั้นก็อาจจะไม่ได้ช่วยให้การฝึกสอนของโมเดลนั้นมีประสิทธิภาพที่มากขึ้นแล้วอาจจะส่งผลทำให้เกิดการเรียนรู้ที่มากหรือซับซ้อนเกินไปทำให้เกิดการตัดสินใจที่ผิดพลาดของข้อมูลและเกิด Overfitting ของการฝึกสอนและทดสอบของชุดข้อมูลนั้น ๆ ได้ ข้อควรระวังของการเพิ่มจำนวนข้อมูลฝึกสอนนั้นจึงควรเพิ่มจำนวนข้อมูลให้อยู่ในจำนวนที่เหมาะสม [10-13, 40]



ตารางที่ 2.5 ตัวอย่างของการเกิด Underfitting และ Overfitting

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> <li>• High training error</li> <li>• Training error close to test error</li> <li>• High bias</li> </ul>	<ul style="list-style-type: none"> <li>• Training error slightly lower than test error</li> </ul>	<ul style="list-style-type: none"> <li>• Very low training error</li> <li>• Training error much lower than test error</li> <li>• High variance</li> </ul>
Regression illustration			
Classification illustration			
Deep learning illustration			

ที่มา : <https://www.kaggle.com/getting-started/166897>

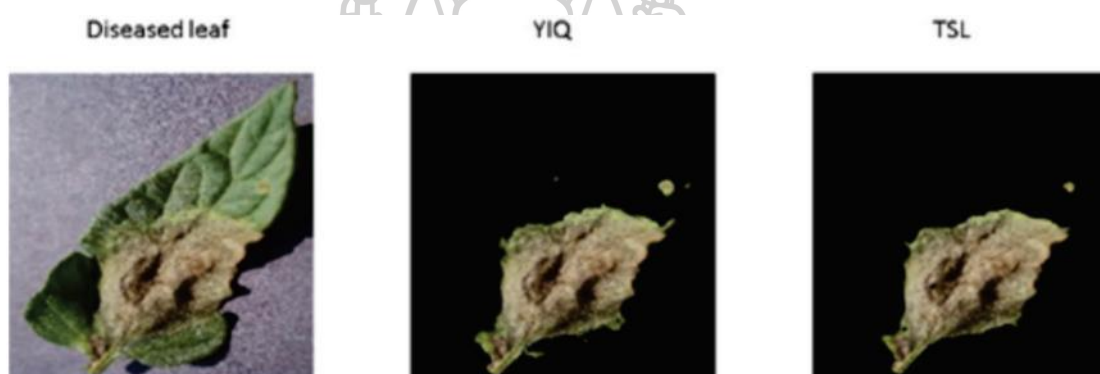
จากตารางที่ 2.5 จะเห็นได้ว่าลักษณะของการเกิด Underfitting นั้นจะเกิดจากการที่มีข้อมูลที่น้อยเกินไปสำหรับฝึกฝนโมเดลจึงทำให้ไม่สามารถตัดสินใจได้ดีในข้อมูลฝึกสอนที่มีความซับซ้อนมาก และในทำนองเดียวกันการเกิด Overfitting นั้นก็จะเกิดจากการที่มีข้อมูลมากเกินไปจากภาพด้านบนจะเห็นได้ว่า Overfitting นั้นพยายามหาค่าตัวแปรของสมการทอดยและค่าค่าความผิดพลาดในการฝึกสอนข้อมูลที่น้อยได้ก็ตามแต่ก็ไม่สามารถนำมาใช้งานจริงได้ จึงแสดงได้ว่าในการมีจำนวนที่มากหรือน้อยเกินไปนั้นส่งผลอย่างมากในการทำงานของ Machine learning และ Deep learning



## 2.1 งานวิจัยที่เกี่ยวข้อง

### 2.1.1 งานวิจัยของ Leidy Esperanza Pamplona Beron, Andrés Felipe Calvo Salcedo, และ Arley Bejarano Martínez เรื่อง “Detection of foliar diseases using image processing techniques” [6]

ในงานวิจัยนี้ได้มีการนำเสนอเกี่ยวกับการพัฒนาวิธีการตรวจหาเปอร์เซ็นต์ของพื้นที่ที่เกิดผลกระทบของโรค *Phytophthora infestans* บนใบมะเขือเทศโดยการใช้เทคนิคการประมวลผลภาพดิจิทัลมาใช้ในการสกัดคุณลักษณะสำหรับแยกพื้นที่ที่สนใจด้วยการวิเคราะห์ทางสี และใช้โมเดลสี YIQ และ TSL สำหรับตรวจจับโรค โดยวิธีการของงานวิจัยนี้ประกอบไปด้วยการแก้ปัญหาที่พบบ่อยของการแบ่งส่วนรูปภาพคือพื้นหลังและตัวโรคที่เกิดจากการถ่ายภาพที่ไม่สามารถควบคุมสภาพแสงได้



รูปที่ 2.34 ภาพการตรวจจับโรคด้วยวิธีที่นำเสนอของงานวิจัยอ้างอิง

ที่มา : “Detection of foliar diseases using image processing techniques,” by Leidy Esperanza Pamplona Beron, Andrés Felipe Calvo Salcedo and Arley Bejarano Martínez, 2020, *Revista Ceres*, 2, 109

ผลลัพธ์ดังรูปที่ 2.34 ด้วยวิธีการข้างต้นนี้ผู้วิจัยได้นำไปเปรียบเทียบกับวิธีการ K-means with LAB (Sharma et al., 2017) และวิธีการ K-means with HSV โดยให้ค่าผลลัพธ์ที่ดีกว่าโดยมีเปอร์เซ็นต์ผิดพลาดที่  $4.32 \pm 5.44$  % ดังแสดงในตารางที่ 2.6

ตารางที่ 2.6 ผลลัพธ์ค่าความผิดพลาดของงานวิจัยที่อ้างอิง

Methodologies	Referred error [%]
1. Proposed method	4.32 ± 5.44
2. K-means with LAB (Sharma et al., 2017)	7.72 ± 7.24
3. K-means with HSV	10.36 ± 7.60

### 2.1.2 งานวิจัยของ Shanwen Zhang, Zhuhong You และ Xiaowei Wu เรื่อง “Plant disease leaf image segmentation based on superpixel clustering and EM algorithm” [15]

งานวิจัยนี้นำเสนอวิธีการจำแนกโรคพืชด้วยวิธีการจัดกลุ่มแบบ Superpixel และ EM algorithm โดยในงานวิจัยนี้ได้เสนอวิธีการแบบใหม่ คือ วิธีการจัดกลุ่มแบบผสม Hybrid clustering โดยการนำเอาวิธีการซูเปอร์พิกเซล SLIC มาช่วยในการจัดกลุ่มภาพใบไม้เพื่อเป็นแนวทางในการนำไปแบ่งส่วนภาพเพื่อให้ EM algorithm สามารถทำงานได้รวดเร็วขึ้นและแม่นยำยิ่งขึ้น

ตารางที่ 2.7 ผลลัพธ์จำนวนรอบและเวลาในการวิเคราะห์ของงานวิจัยที่อ้างอิง

Method	K-means	Fuzzy C-means	EM algorithm	Superpixels + EM
Iterative number	378	475	362	659
Computation time (ms)	325	371	234	168

จากตารางที่ 2.7 จะเห็นได้ว่าวิธีที่ผู้วิจัยนำเสนอเป็นวิธีที่มีผลของการวิเคราะห์เร็วกว่าอัลกอริทึมอื่น ๆ ถึงแม้จะมีจำนวนรอบที่เยอะกว่าก็ตาม และเมื่อเปรียบเทียบกับ EM algorithm กับ superpixels + EM ก็สามารถสรุปได้ว่าการจัดกลุ่มด้วย superpixels สามารถช่วยให้ EM algorithm ทำงานได้เร็วยิ่งขึ้น

### 2.1.3 งานวิจัยของ Prakruti Bhatt, Sanat Sarangi และ Srinivasu Pappula เรื่อง “Comparison of CNN Models for Application in Crop Health Assessment with Participatory Sensing” [32]

ได้นำเสนอการเปรียบเทียบโมเดล CNN ในการนำไปใช้ประเมินความแข็งแรงของพืชโดยใช้ Participatory sensing และได้เปรียบเทียบกับโมเดล 4 โมเดล ได้แก่ VGG19, Inception-V3, Xception และ ResNet50 แสดงผลลัพธ์ดังตารางที่ 2.8

ตารางที่ 2.8 ผลลัพธ์ค่าความแม่นยำ

Architecture	Size (MB)	Memory usage (MB)	Operations count	Inference time (s)
VGG-19	597	658	19,632,166,912	1.073
Inception-v3	95	297.4	5,713,271,392	0.624
Xception	91	221.7	649,023,584	0.775
Resnet-50	102	258.8	3,834,386,432	0.737

	Color Format	VGG-19	Inception-v3	Xception	ResNet-50
80-10-10	RGB	98.2	98.4	98.6	99.7
	Gray	92.8	98.17	95.8	98.74
	CIElab	93.4	98.18	96.1	97.9
60-20-20	RGB	98.6	98.01	97.94	99.5
	Gray	93.10	97.6	95.3	98.45
	CIElab	92.9	97.73	97.9	97.3

2.1.4 งานวิจัยของ G.Suresh, V.Gnanaprakash และ R.Santhiya เรื่อง “Performance Analysis of Different CNN Architecture with Different Optimisers for Plant Disease Classification” [37]

นำเสนอการวิเคราะห์ประสิทธิภาพความแตกต่างของโครงสร้าง CNN กับความต่างของ Optimisers สำหรับจำแนกโรคพืช ได้มีการเปรียบเทียบโมเดล Inception-V3, MobileNetV2, ResNetV2 ในการทำ Optimisers Adagrad, Adadelta และ Adam โดยได้ผลลัพธ์ตารางที่ 2.9

ตารางที่ 2.9 ตารางเปรียบเทียบโมเดลกับการทำงานของชนิด Optimizers

S. No	Architecture	Validation Accuracy for different optimizers (%)		
		Adagrad	Adadelta	Adam
1	Inception v3	97.2	99.3	99.9
2	Mobilenet_v2_140_224	95.1	99.1	99.3
3	Resnet v2_50	88.8	95.8	95.8

ในงานวิจัยนี้สรุปได้ว่าการเลือกใช้ Optimizers กับโมเดลนั้นมีผลต่อค่าความถูกต้องในการวิเคราะห์ ซึ่งในการทำ Optimizer เป็นการทำให้ระบบพบเกิดการปรับปรุงค่า Error หรือ Loss เมื่อขณะทำการฝึกฝนอยู่โดยทำการปรับเปลี่ยนค่าน้ำหนัก (Weight) และค่าเอนเอียง (Bias)

2.1.5 งานวิจัยของ Endang Suryawati, Rika Sustika, R. Sandra Yuwana, Agus Subekti และ Hilman F. Pardede เรื่อง “*Deep Structured Convolutional Neural Network for Tomato Diseases Detection*” [33]

ในงานวิจัยนี้เป็นงานวิจัยที่ทำการทดสอบประสิทธิภาพของโมเดลทั้ง 4 โมเดล ได้แก่ Baseline, AlexNet, GooglrNet และ VGGNet (VGG16) โดยนำไปฝึกสอนกับชุดภาพฝึกสอนจากฐานข้อมูล PlantVillage สำหรับตรวจจับโรคที่เกิดกับมะเขือเทศ มีผลลัพธ์แสดงในตารางที่ 2.10

ตารางที่ 2.10 ผลการทดลอง


Architecture	Accuracy (%)	Loss
Baseline	84.58	0.47
AlexNet	91.52	0.51
GoogleNet	89.68	0.30
VGGNet	<b>95.24</b>	<b>0.17</b>

จากการทดลองทำการแบ่งภาพสำหรับ train 80% และสำหรับ validate 10% มีจำนวน class ทั้งหมด 10 class ที่เป็นโรค 9 class ได้แก่ Bacterial spot 2,127 ภาพ Early blight 1,000 ภาพ Late blight 1,909 ภาพ Leaf Mold 952 ภาพ Septoria Leaf spot 1,771 ภาพ Spider mites 1,676 ภาพ Target spot 1,404 ภาพ Mosaic virus 373 ภาพ Yellow Leaf Curl virus 5,357 ภาพ และ 1 Class เป็นมะเขือเทศไม่เป็นโรค 1,591 ภาพ ผลที่ได้คือ VGGNet สามารถให้ค่าความถูกต้องสูงกว่าโมเดลอื่น ๆ และสรุปว่าโมเดลยังมีความลึกจะช่วยในการแก้ไขให้มีความแม่นยำในการจำแนกขึ้นได้ และขนาดของตัวกรองมีผลอย่างมากในการลดกำลังในการคำนวณ

2.1.6 งานวิจัยของ Azeddine Elhassouny และ Florentin Smarandache เรื่อง “*Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks*” [27]

ในงานวิจัยนี้ได้นำเสนอการรู้จำโรคบนใบแอปเปิ้ลโดยใช้ CNN ด้วยระบบ Smart mobile application โดยใช้โมเดล MobileNet กับ Optimisers จากการทดลองเบื้องต้นของโมเดลในการจำแนกโรคที่ใบมะเขือเทศได้ผลดังตารางที่ 2.11 และตารางที่ 2.12

ตารางที่ 2.11 ผลการทดลองเบื้องต้นของโมเดล MobileNet

Image example	Obtained result	Expected results
	<ul style="list-style-type: none"> <li>- leaf mold passalora fulva 0.99999</li> <li>- septoria leaf spot septoria lycopersici 5.87832e-06</li> <li>- mosaic virus 2.34571e-06</li> <li>- late blight phytophthora infestans 1.02782e-06</li> <li>- early blight alternaria solani 2.06558e-07</li> </ul>	- leaf mold passalora fulva

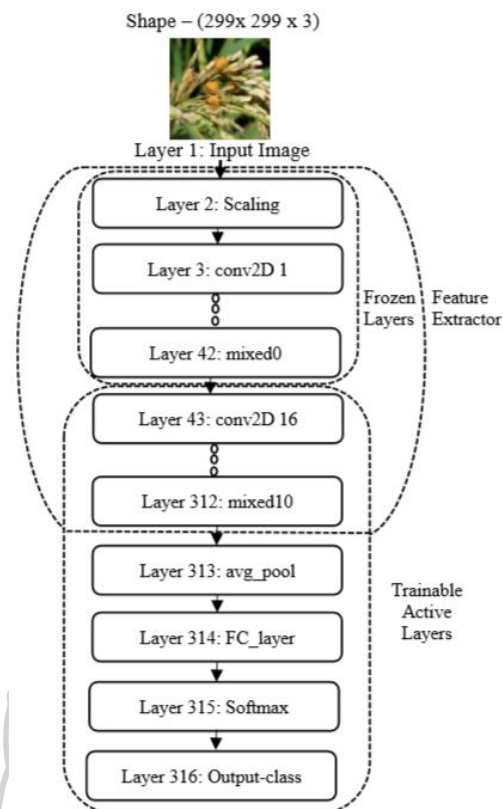
ตารางที่ 2.12 ตารางผลของการเลือกใช้ค่า Optimizers

Optimization method	Final test accuracy
Stochastic gradient descent	88.4%
Adagrad	88.3%
SGD with Momentum	87.6%
Adam	88.5%
Proximal Gradient Descent	89.2%
Proximal Adagrad	88.8%
RMSProp	85.9%

จากงานวิจัยได้ทำการเลือก MobileNet มาใช้บนระบบ Smart mobile เนื่องจากมีขนาดความจำค่อนข้างเล็กและจากผลในตารางที่ 2.12 เมื่อนำโมเดล MobileNet มาทำการเปรียบเทียบค่าที่ได้จากการทำ Optimize จะเห็นว่า Proximal Gradient Descent ทำงานได้ดีกับโมเดลโดยให้ค่าความถูกต้องที่ 89.2%

2.1.7 งานวิจัยของ Md. Jahid Hasan, Shamim Mahbub, Md. Shahin Alom และ Md. Abu Nasim เรื่อง “Rice Disease Identification and Classification by Integrating Support Vector Machine with Deep Convolutional Neural Network” [41]

ได้นำเสนอการจำแนกลักษณะโรคของข้าวโดยการพัฒนา SVM กับ DCNN โดยขั้นแรกทำ Transfer learning ของ Inception-V3 กับ DCNN ดังรูปที่ 2.35 และทำการฝึกฝนกับชุดข้อมูลภาพจำนวน 1,080 ภาพ เมื่อนำไปทดสอบได้ค่าความถูกต้องที่ 97.5%



รูปที่ 2.35 โครงสร้างการทำ Transfer learning ของโมเดล Inception-V3 กับ DCNN ที่มา : “Rice Disease Identification and Classification by Integrating Support Vector Machine with Deep Convolutional Neural Network,” by Md. Jahid Hasan, Shamim Mahbub, Md. Shahin Alom and Md. Abu Nasim, 2019, *International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 1

2.1.8 งานวิจัยของ Guoxiong Zhou, Wenzhuo Zhang, Aibin Chen, Mingfang He และ Xueshuo Ma เรื่อง “*Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion*” [7]

ได้นำเสนอวิธีการตรวจจับโรคบนข้าวอย่างรวดเร็วโดยการรวมกันของ FCM-KM และ Faster R-CNN โดยในขั้นตอนการทำ Preprocessing ได้มีการปรับปรุงรูปภาพเพื่อให้เกิดลักษณะเด่นของโรค ได้ผลลัพธ์ดังรูปที่ 2.36





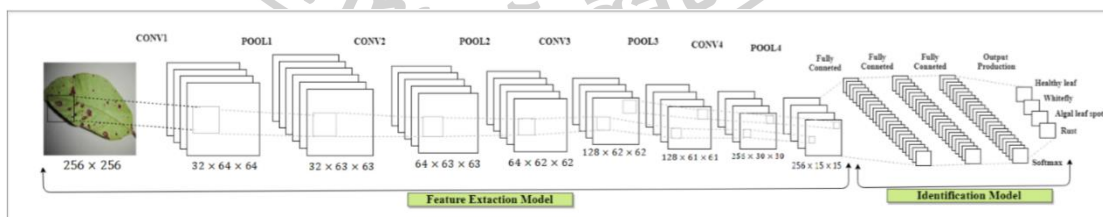
รูปที่ 2.36 การทำ Segmentation ด้วย Otsu thresholding

ที่มา : “Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion,”  
by Guoxiong Zhou, Wenzhuo Zhang, Aibin Chen, Mingfang He and Xueshuo Ma, 2019,  
*IEEE*, 143195

ในงานวิจัยการทำ Otsu threshold segmentation ก็เพื่อต้องการที่จะให้โมเดลสามารถทำงานตรวจจับโรคได้เร็วขึ้น

2.1.9 งานวิจัยของ Md. Rasel Howlader, Umme Habiba, Rahat Hossain Faisal และ Md. Mostafijur Rahman เรื่อง “Automatic Recognition of Guava Leaf Diseases using Deep Convolution Neural Network” [42]

ได้นำเสนอการรู้จำอัตโนมัติโรคที่เกิดบนใบฝรั่งโดยออกแบบ D-CNN ขึ้นมาโดยอ้างอิงจากโมเดล AlexNet มีจำนวนชั้นทั้งหมด 11 ชั้น ประกอบด้วย Convolution layers 4 ชั้น Max pooling layers 4 ชั้น และ Fully connection 3 ชั้น โดยมีขนาดและค่าตัวแปรดังรูปที่ 2.37



รูปที่ 2.37 โครงสร้างโมเดล D-CNN

ที่มา : “Automatic Recognition of Guava Leaf Diseases using Deep Convolution Neural Network,” by Md. Rasel Howlader, Umme Habiba, Rahat Hossain Faisal and Md. Mostafijur Rahman, 2019, *International Conference on Electrical, Computer and Communication Engineering (ECCE)*



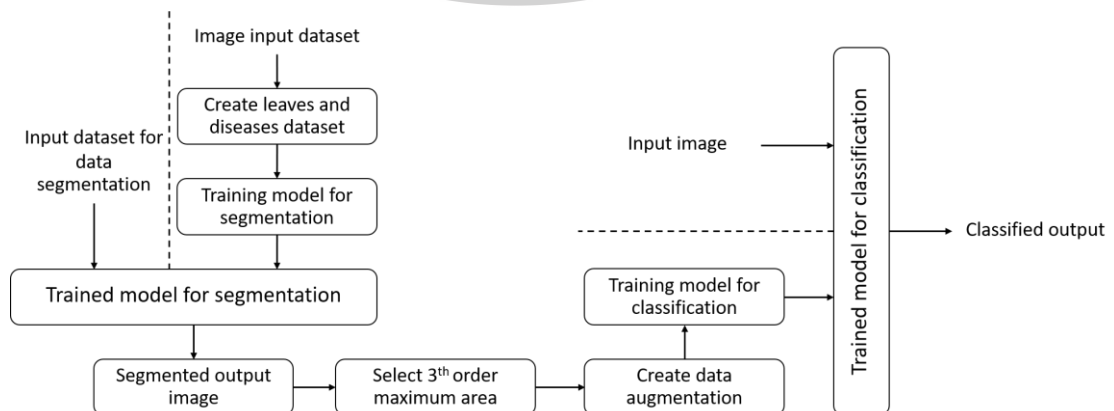
ในการทำงานของ D-CNN นี้จะต้องทำการปรับขนาดภาพให้เป็น  $256 \times 256$  พิกเซล และในชั้นของการคอนโวลูชันชั้นแรกมีจำนวนฟิลเตอร์ 32 ตัว ได้ Kernel ขนาด  $3 \times 3$  โดยชั้นต่อ ๆ ไปจะเป็น  $(64 \times 3 \times 3)$ ,  $(128 \times 3 \times 3)$  และ  $(256 \times 3 \times 3)$  ซึ่งช่วยให้มีประสิทธิภาพมากขึ้นโดยคอนโวลูชันของชั้นที่ 1 จะขยับ Kernel ทีละ 4 จังหวะ คอนโวลูชันและ Max pooling ของชั้นที่ 4 ขยับ Kernel ทีละ 2 จังหวะ ซึ่งผลลัพธ์ที่ได้มาค่าความแม่นยำในการจำแนก 98.74% เทียบกับ AlexNet ที่ 97.92%



### บทที่ 3 วิธีการและขั้นตอนการดำเนินงาน

การออกแบบพัฒนาระบบวิธีการจำแนกโรคพืชโดยใช้การเรียนรู้เชิงลึกนั้น ในวิทยานิพนธ์เล่มนี้ จึงได้มีการพัฒนาโปรแกรมสำหรับจำแนกและตรวจหาชนิดของโรคพืชด้วยการเรียนรู้เชิงลึก ซึ่งได้ ทำการศึกษาวิธีการจัดกลุ่มของข้อมูลแล้วนำมาใช้โดยเลือกวิธีการจัดกลุ่มด้วยวิธีการทำ SLIC superpixel เพื่อให้เกิดการจัดกลุ่มของข้อมูลบริเวณที่เป็นโรคและใบได้แม่นยำ และนำข้อมูลที่ได้ไป ใช้ฝึกสอนกับ CNN model ที่ออกแบบไว้ข้างต้น และนำโมเดลที่ได้นั้นมาใช้สำหรับการจัดเตรียม ข้อมูลครั้งต่อไปด้วยการทำ Local-based image augmentation จะเป็นการเพิ่มจำนวนของข้อมูล ในลักษณะของการเพิ่มข้อมูลเชิงพื้นที่และนำไปใช้ฝึกสอนกับโมเดลทั้ง 3 โมเดล ได้แก่ CNN model, VGG19 และ MobileNet สำหรับการวิเคราะห์จำแนกชนิดของโรคพืชต่อไป

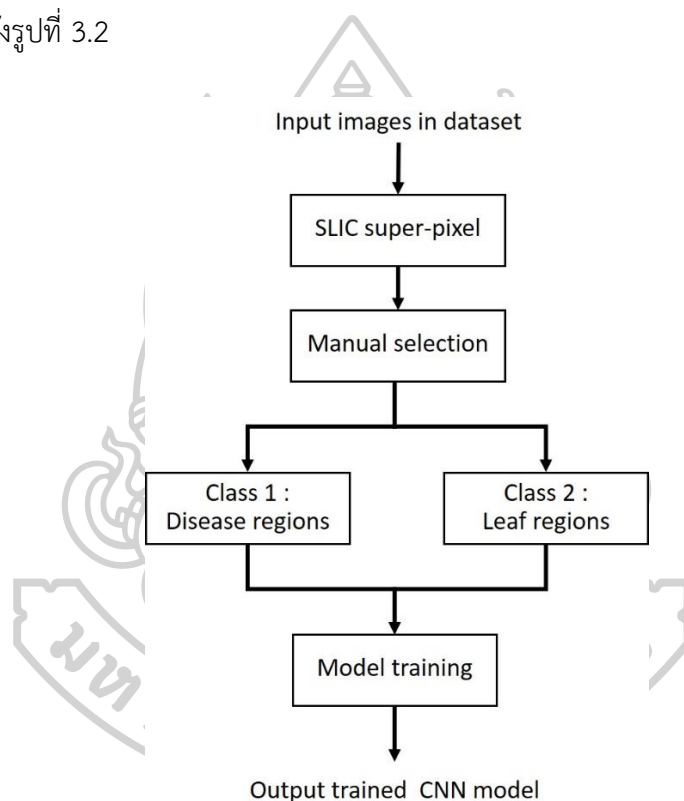
ในบทนี้จะกล่าวถึงลำดับขั้นตอนและวิธีการดำเนินงานในการทำโครงงานวิจัยนี้ เป็นการอธิบาย ถึงโครงสร้างและลำดับขั้นตอนการทำงานของโปรแกรม โดยในงานวิจัยนี้ได้แบ่งขั้นตอนและวิธีการ ดำเนินงานออกเป็น 6 ขั้นตอน ประกอบไปด้วย ขั้นตอนที่ 1. การแบ่งส่วนภาพใบไม้ด้วยวิธีซูเปอร์ พิกเซล SLIC สำหรับสร้างข้อมูลชุดฝึกสอนให้กับโมเดล CNN ขั้นตอนที่ 2. การฝึกสอน CNN model ขั้นตอนที่ 3. การแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ได้ฝึกสอนมาแล้ว ขั้นตอนที่ 4. การ เพิ่มจำนวนข้อมูลด้วยการทำ Images augmentation ขั้นตอนที่ 5. ฝึกสอนโมเดลที่จะนำไปใช้ สำหรับการจำแนกชนิดโรคพืช และขั้นตอนที่ 6. ขั้นตอนของการทดสอบโมเดลที่ฝึกสอนมาแล้วกับชุด ข้อมูลทดสอบ โดยจะเป็นการอธิบายลำดับขั้นตอนประกอบภาพ Framework และ Block diagram ตามรูปที่ 3.1



รูปที่ 3.1 ภาพรวมของงานวิจัย

### 3.1 การแบ่งส่วนภาพใบไม้ด้วยวิธีซูเปอร์พิกเซล SLIC สำหรับสร้างข้อมูลชุดฝึกสอนให้กับโมเดล CNN

ความสำคัญของการจำแนกหรือจัดกลุ่มให้กับข้อมูลลักษณะของโรคพืชที่นำมาใช้สำหรับเป็นชุดข้อมูลฝึกสอน เพื่อให้ได้ลักษณะเฉพาะที่บ่งบอกถึงความเป็นโรคและใบได้อย่างถูกต้อง ซึ่งได้กล่าวไว้ในบทที่ 1 เพื่อง่ายต่อการนำไปใช้งานและยังช่วยให้โมเดลที่ออกแบบมานั้นสามารถเรียนรู้และจำแนกลักษณะเด่นของประเภทของข้อมูลนั้น ๆ ได้แม่นยำยิ่งขึ้น ในส่วนของการจำแนกจัดกลุ่มของข้อมูลที่จะนำไปใช้สำหรับฝึกสอนโมเดลที่นำมาจำแนกบริเวณที่เป็นโรคนั้นมีลำดับขั้นตอนการทำงานตาม Flowchart ดังรูปที่ 3.2

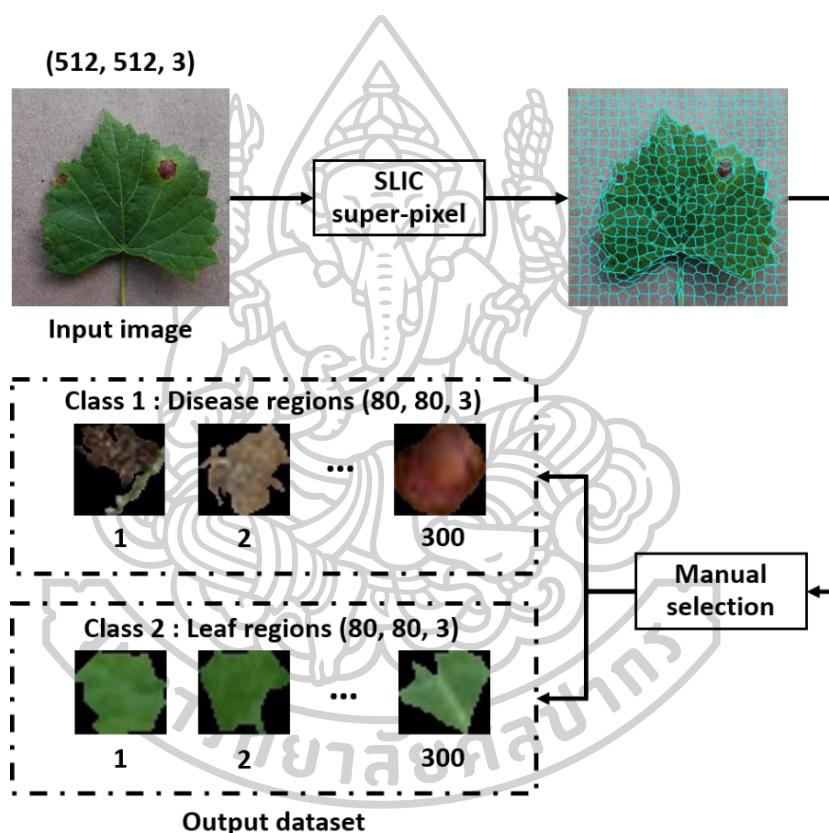


รูปที่ 3.2 ขั้นตอนของการเตรียมข้อมูลฝึกสอน

จากรูปที่ 3.2 แสดงขั้นตอนการออกแบบและการทำงานโปรแกรมในส่วนของกระบวนการแบ่งส่วนของภาพเพื่อนำมาใช้เป็นชุดฝึกสอนให้กับโมเดลของการแบ่งส่วนบริเวณของโรคและบริเวณของใบ โดยสามารถแบ่งออกเป็นขั้นตอนหลักได้ 2 ขั้นตอน คือ 1. ขั้นตอนของการแบ่งส่วนภาพ 2. ขั้นตอนการเลือกบริเวณข้อมูลภาพ

โดยในขั้นตอนที่ 1 ขั้นตอนการแบ่งส่วนภาพ เป็นขั้นตอนการแบ่งส่วนของพื้นที่บริเวณภายในภาพให้เป็นขอบเขตบริเวณพื้นที่ของโรคกับบริเวณพื้นที่ของใบโดยใช้วิธีการแบ่งขอบเขตด้วย SLIC

superpixel ซึ่งจะเป็นวิธีที่สามารถแบ่งขอบเขตพื้นที่ของบริเวณที่เป็นโรคกับใบได้ดี โดยวิธีของ SLIC superpixel จะมองเห็นว่าพิกเซลที่มีลักษณะของสีที่เหมือนกันหรือใกล้เคียงกันจะถือว่าเป็นกลุ่มของพิกเซลเดียวกันจึงสามารถแบ่งขอบเขตระหว่างพื้นที่ของโรคและใบได้ดีเนื่องจากมีสีที่ค่อนข้างแตกต่างกัน และในขั้นตอนที่ 2 จากขั้นตอนแรกจะได้ภาพที่ถูกแบ่งขอบเขตด้วย Superpixel ทางผู้วิจัยใช้วิธีการเลือกพื้นที่กลุ่มข้อมูลที่จะนำมาใช้ด้วยตนเองเพื่อให้ได้กลุ่มข้อมูลที่มีลักษณะที่หลากหลาย โดยแบ่งเป็นกลุ่มข้อมูลของบริเวณที่เป็นโรค 300 ภาพ และกลุ่มข้อมูลของบริเวณที่เป็นใบอีก 300 ภาพ วิธีการข้างต้นแสดงเป็น Block diagram ได้ดังรูปที่ 3.3

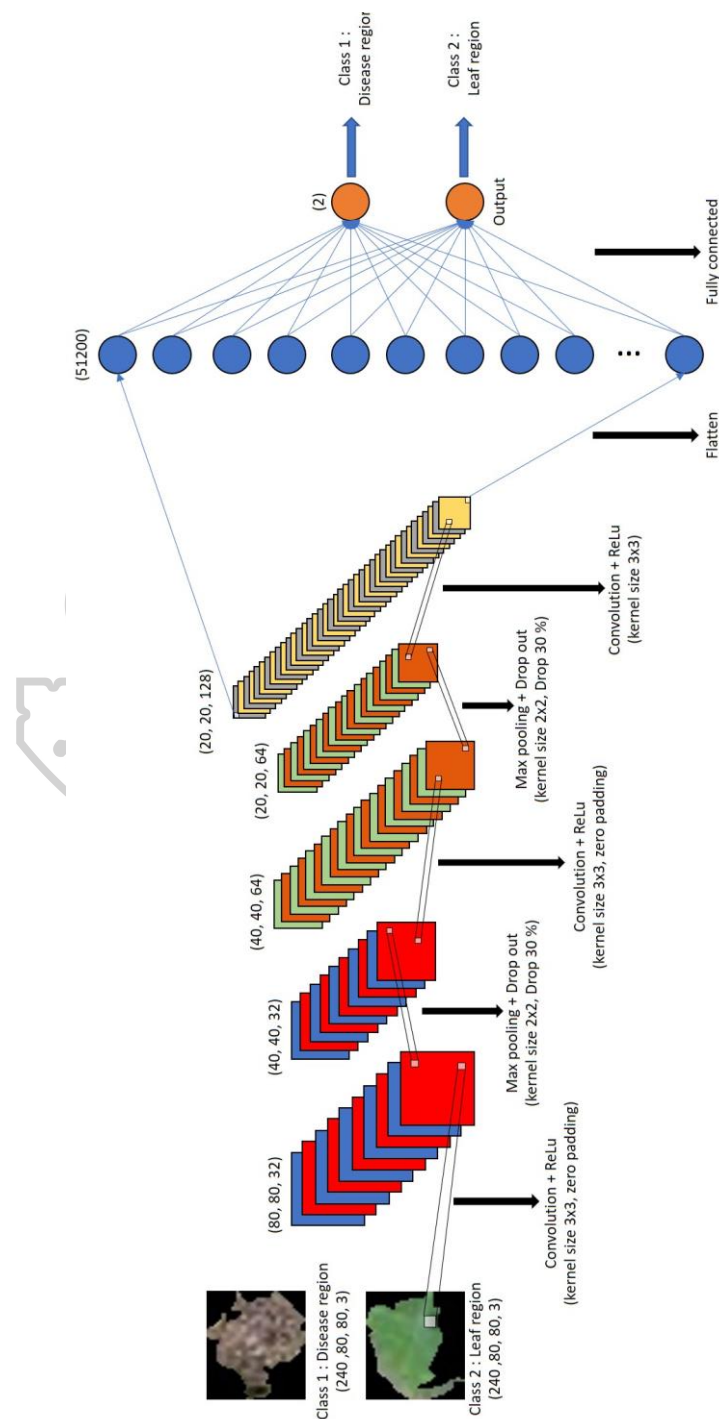


รูปที่ 3.3 บล็อกไดอะแกรมขั้นตอนการทำ Superpixel สำหรับสร้างชุดข้อมูลฝึกสอน

### 3.2 การฝึกสอน CNN model สำหรับการแบ่งส่วนบริเวณที่เป็นโรคบนใบ

ขั้นตอนการฝึกสอนโมเดลในขั้นตอนนี้เป็นการฝึกสอนโมเดลสำหรับโปรแกรมจำแนกบริเวณที่เป็นโรคกับใบ โดยโมเดลที่นำมาฝึกสอนนั้นเป็นโมเดล CNN ซึ่งเป็นโมเดลที่ทางผู้วิจัยได้ออกแบบขึ้นมา โดยที่เลือกใช้ CNN model เนื่องจากข้อมูลที่นำมาใช้ฝึกสอนนั้นเป็นข้อมูลที่มีความแตกต่างกันอย่างชัดเจนงานต่อการจำแนกจึงเลือกใช้โมเดลนี้มาใช้งานซึ่งเป็นโมเดลที่ไม่มีความซับซ้อนในการสร้าง แต่ให้ผลลัพธ์ของการจำแนกที่ค่อนข้างแม่นยำ โดยโมเดลที่ออกแบบนั้นมีจำนวนชั้นของ

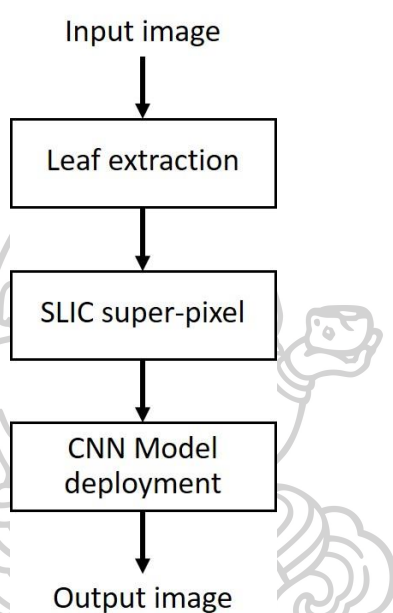
Hidden layer ทั้งหมด 7 ชั้น ประกอบด้วย ชั้น Convolution (Kernel size 3x3, Zero padding) + Relu จำนวน 3 ชั้น Max pooling (Kernel size 2x2) + Drop out (Drop 30%) 2 ชั้น Flattening 1 ชั้น และชั้นของ Fully connected 1 ชั้น โดยจำนวนเอาต์พุตหรือ Class ของโมเดลที่ออกมาจะเป็น 2 Classes คือ Class ที่ 1 เป็น Class ของบริเวณโรค และ Class ที่ 2 เป็น Class ของบริเวณใบ แสดงรายละเอียดของโมเดลในรูปที่ 3.4



รูปที่ 3.4 CNN model ที่ใช้ในการจำแนกบริเวณที่เป็นโรคและบริเวณใบ

### 3.3 การแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ได้ฝึกสอนมาแล้ว

ในส่วนขั้นตอนนี้กล่าวถึงการนำโมเดลจากขั้นตอนของการฝึกสอนโมเดลแล้วข้างต้นมาใช้สำหรับแบ่งส่วนบริเวณที่เป็นโรคและใบ โดยจะมีขั้นตอนการทำงานแบ่งออกเป็น 3 ขั้นตอนคือ 1. การสกัดคุณลักษณะ 2. สร้างขอบเขตของพื้นที่ใบไม้ด้วย SLIC super-pixel และ 3. ทำการวิเคราะห์แบ่งส่วนที่เป็นโรคโดยใช้โมเดลที่ผ่านการฝึกสอนมาแล้ว โดยในลำดับขั้นตอนนี้ดังกล่าวแสดงดังรูปที่ 3.5

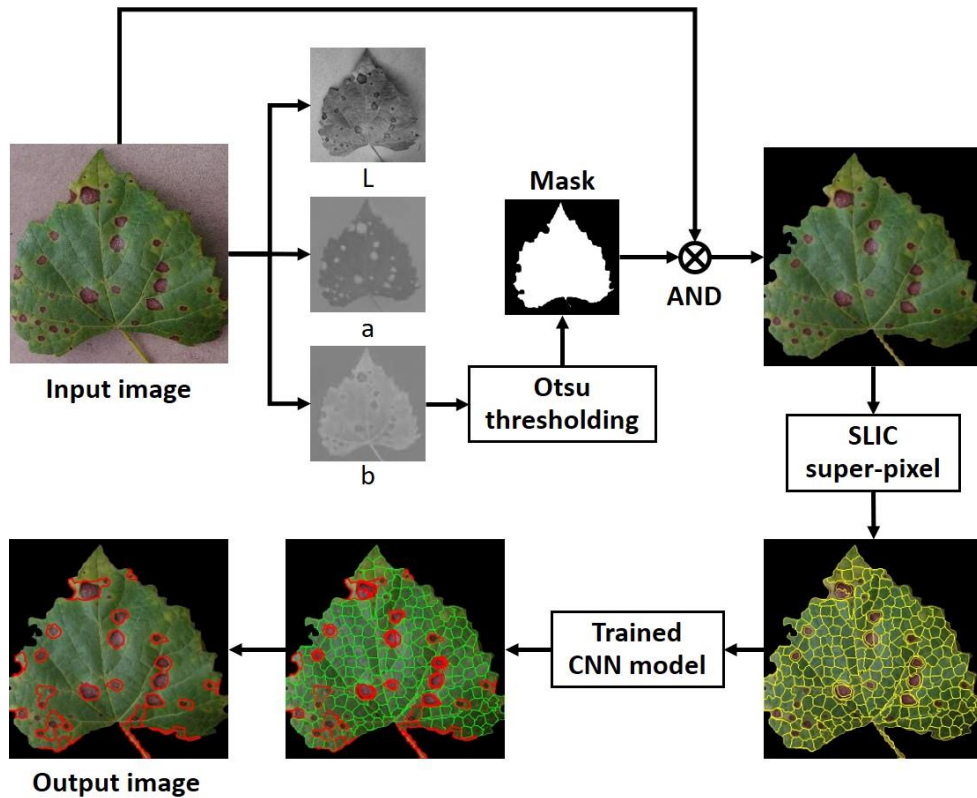


รูปที่ 3.5 ขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ผ่านการฝึกสอนมาแล้ว

จากที่กล่าวมาข้างต้นได้มีการแบ่งลำดับการทำงานออกเป็น 3 ส่วน โดยในขั้นตอนแรกเป็นขั้นตอนการสกัดคุณลักษณะของภาพทดสอบที่เตรียมไว้โดยเป็นการสกัดพื้นที่บริเวณที่ไม่เป็นใบไม้หรือก็คือพื้นที่ฉากหลังเพื่อให้ได้ภาพที่เป็นภาพใบไม้อย่างเดียวเพื่อให้มั่นใจได้ว่าพื้นหลังของภาพจะไม่ไปกระทบการจำแนกบริเวณที่เป็นโรค ในงานวิจัยได้ใช้วิธีการเปลี่ยนระบบสีจาก RGB ให้เป็น  $L^*a^*b^*$  และนำภาพของแชนเนล  $b^*$  มาทำ Otsu threshold ให้เหลือแต่พื้นหลังเพื่อสร้างเป็นหน้ากากสำหรับนำมา and operator กับภาพต้นจึงได้ภาพใบไม้อย่างเดียว เนื่องจากที่เลือกใช้แชนเนล  $b^*$  เพราะแชนเนล  $b^*$  ประกอบไปด้วยองค์ประกอบของสีน้ำเงินกับสีเหลืองซึ่งถือว่าเป็นองค์ประกอบส่วนหนึ่งของสีเขียวจึงทำให้พื้นที่บริเวณที่เป็นใบไม้นั้นเด่นชัดขึ้นมาทำให้ง่ายและแม่นยำในการสกัดคุณลักษณะ



หลังจากการสกัดคุณลักษณะของใบแล้วจึงนำมาแบ่งส่วนพื้นที่บนใบเพื่อเป็นการตรวจสอบว่ามีพื้นที่ของพื้นหลังเหลืออยู่แล้วจึงนำภาพที่จัดเตรียมไว้มาใช้กับโมเดลที่ฝึกสอน โดยจากที่กล่าวมาข้างต้นลำดับนั้นขั้นตอนของการทำงานในส่วนนี้แสดงดังรูปที่ 3.6



รูปที่ 3.6 บล็อกไดอะแกรมขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรค

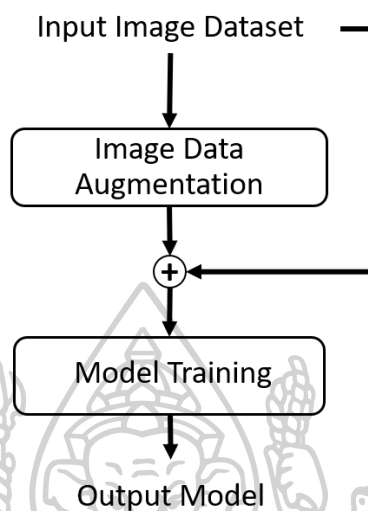
ในการที่แบ่งส่วนภาพเพื่อวิเคราะห์หาบริเวณที่เป็นโรคในขั้นตอนข้างต้นนั้น ข้อมูลหรือผลลัพธ์ที่ได้จะนำไปใช้ต่อในกระบวนการทำ Images augmentation เพื่อให้เกิดรูปแบบของข้อมูลที่หลากหลายยิ่งขึ้น

### 3.4 การเพิ่มจำนวนข้อมูลภาพด้วยการทำ Data Augmentation

ในขั้นตอนของการทำ Images augmentation ขั้นตอนนี้จะเป็นการนำเอาบริเวณพื้นที่ของโรคที่หาได้จากขั้นตอนของการแบ่งส่วนในข้อที่ 3.2 มาใช้ โดยจะเลือกบริเวณพื้นที่ที่มีขนาดความใหญ่ของบริเวณโรค 3 อันดับแรกมาใช้ โดยจะนำเอาบริเวณพื้นที่โรคที่ได้นั้นมากระจายตัวไปยังบนภาพที่จะนำมาใช้เป็นชุดภาพฝึกสอนเพื่อให้เกิดความหลากหลายของบริเวณที่เกิดโรค โดยทางผู้วิจัยเรียก

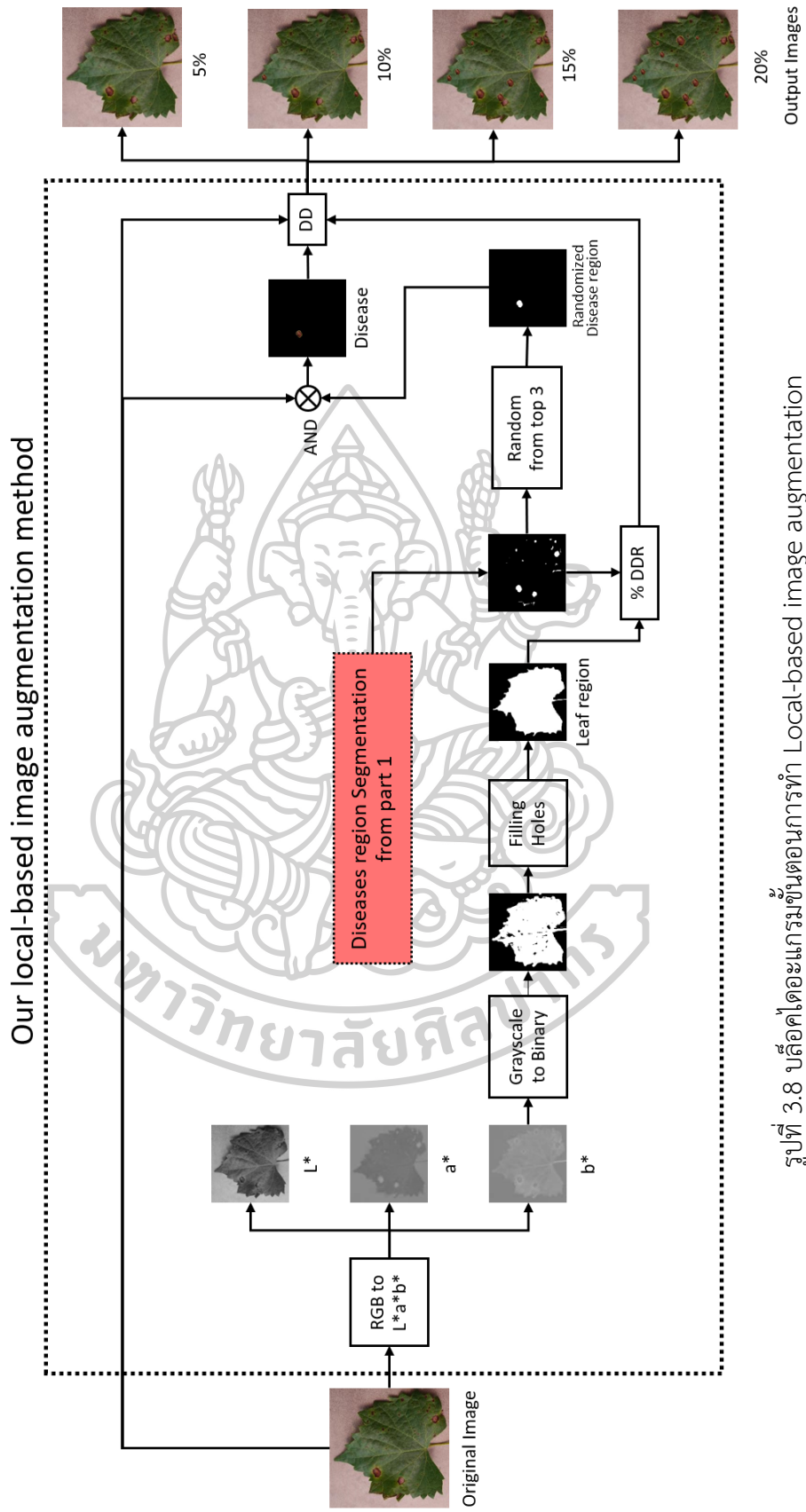


วิธีการนี้ว่า Local-based image augmentation ซึ่งใกล้เคียงกับความเป็นจริงที่ว่าโรคบนใบไม้ นั้นอาจเกิดได้ทุก ๆ บริเวณของใบไม้ โดยมีลำดับขั้นตอนแสดงดังรูปที่ 3.7



รูปที่ 3.7 ขั้นตอนการทำ Data augmentation

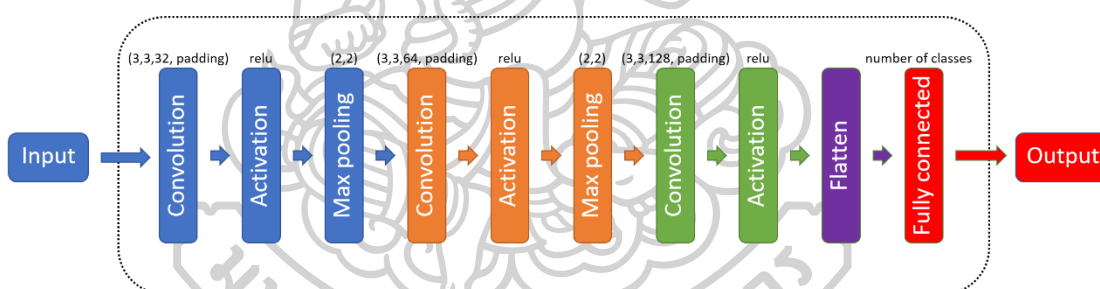
ในอันดับแรกนำภาพที่ใช้สำหรับเป็นภาพฝึกสอนมาเปลี่ยนแชนเนลสีจาก RGB เป็น  $L*a*b^*$  โดยจะนำภาพที่ได้จากแชนเนล  $b^*$  มาแปลงเป็นภาพขาวดำโดยให้พื้นที่บริเวณใบเป็นสีขาวและทำการอุดรอยรั้วที่อยู่ภายในสีขาวเพื่อให้ได้ภาพโครงร่างของใบไม้สำหรับนำมาใช้เปรียบเทียบหาเปอร์เซ็นต์ของพื้นที่ที่เป็นโรคจากภาพขาวดำที่ได้จากการแบ่งส่วนหาบริเวณโรคจากขั้นตอนก่อนหน้า หลังจากนั้นทำการหาบริเวณของโรคที่มีขนาดใหญ่ที่สุด 3 อันดับมาสุ่มเลือกและนำไป and operator กับภาพต้นฉบับเพื่อให้ได้ภาพบริเวณโรคเป็น RGB แล้วนำส่วนที่ได้ไปกระจายลงบนภาพต้นฉบับ โดยเงื่อนไขของการกระจายโรคลงบนใบไม้ นั้นจะคิดจากร้อยละของพื้นที่บริเวณที่เป็นโรคบนพื้นที่ทั้งหมด หากมีพื้นที่ที่เป็นโรคน้อยกว่า 5%, 10%, 15% และ 20% ของพื้นที่ทั้งหมดก็จะทำการกระจายตัวของโรคลงบนภาพต้นแบบให้มีพื้นที่ของโรคเท่ากับเงื่อนไขที่ตั้งไว้ เช่น ถ้ามีพื้นที่ของโรคคิดเป็นร้อยละ 3 ของใบก็จะกระจายตัวของโรคให้มีพื้นที่เท่ากับ 5%, 10%, 15% และ 20% แต่ถ้าใบไม้มีพื้นที่ของโรคคิดเป็นร้อยละ 13 ของใบก็จะกระจายตัวของโรคให้มีพื้นที่เท่ากับ 15% และ 20% เป็นต้น และขนาดของโรคที่นำมากระจายนั้นจะใช้วิธีการสุ่มขนาดเพื่อให้มีขนาดเล็กใหญ่ที่แตกต่างกันไป จากวิธีการข้างต้นนั้นสามารถแสดงเป็นบล็อกไดอะแกรมได้ดังรูปที่ 3.8



รูปที่ 3.8 บล็อกไดอะแกรมขั้นตอนการทำ Local-based image augmentation

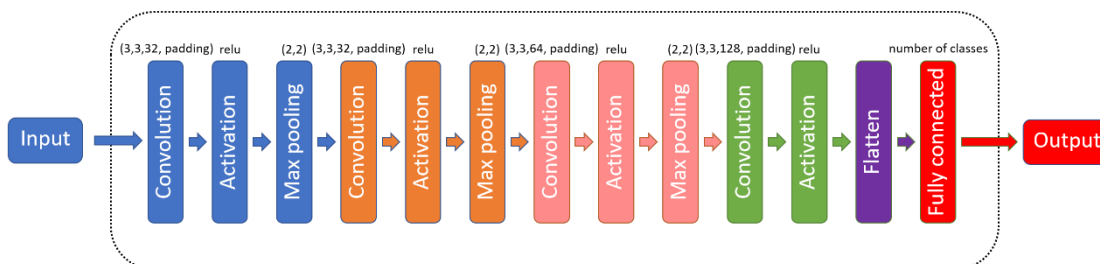
### 3.5 ฝึกสอนโมเดลที่จะนำไปใช้สำหรับการจำแนกชนิดโรคพืช

แผนขั้นตอนวิธีการฝึกสอนโมเดลสำหรับจำแนกโรคพืชนั้น ในงานวิจัยนี้ได้ใช้โมเดลสำหรับการจำแนกโรคด้วยกัน 4 ชนิด ซึ่งจะประกอบไปด้วยโมเดลที่สร้างขึ้น 2 โมเดล และโมเดลที่ผ่านการฝึกสอนโดย Keras TensorFlow อีก 2 โมเดล ได้แก่ VGG19 และ MobileNet โดยโครงสร้างของทั้งสองโมเดลนั้นได้กล่าวไว้แล้วข้างต้นในบทที่ 2 และอีก 2 โมเดลที่สร้างขึ้นมานั้นเป็นโมเดลชนิด CNN โมเดลทั้งสองโมเดล แต่มีความแตกต่างกันที่จำนวนชั้นของ Hidden layer เพื่อที่จะทดสอบว่าจำนวนชั้นของ hidden layer ที่แตกต่างกันนั้นจะสามารถให้ค่าความแม่นยำในการฝึกสอนและทดสอบที่ดีขึ้นหรือไม่ โดยโครงสร้างของ CNN model ที่ออกแบบนั้นโมเดลแรกประกอบด้วยชั้นของ Convolution (32 Conv, Kernel size 3x3, Zero padding) + Relu + Max pooling (Kernel size 2x2) ในชุดที่หนึ่ง Convolution (64 Conv, Kernel size 3x3, Zero padding) + Relu + Max pooling (Kernel size 2x2) ในชุดที่สอง Convolution (128 Conv, Kernel size 3x3, Zero padding) + Relu ในชุดที่สาม ชุดสุดท้ายจะเป็นการทำ Flattening เพื่อเข้าสู่ Fully connected และมีการ Bias ค่าการเรียนรู้กลับทุกตัวของการทำ Convolution แสดงดังรูปที่ 3.9



รูปที่ 3.9 โมเดล CNN model 7 layers

โมเดล CNN model อันที่สองที่ได้ออกแบบนั้นประกอบด้วยชั้นของ Convolution (32 Conv, Kernel size 3x3, Zero padding) + Relu + Max pooling (Kernel size 2x2) จำนวน 2 ชุด และตามด้วย Convolution (64 Conv, Kernel size 3x3, Zero padding) + Relu + Max pooling (Kernel size 2x2) ในชุดที่สาม Convolution (128 Conv, Kernel size 3x3, Zero padding) + Relu ในชุดที่สี่ ชุดสุดท้ายจะเป็นการทำ Flattening เพื่อเข้าสู่ Fully connected และมีการ Bias ค่าการเรียนรู้กลับทุกตัวของการทำ Convolution แสดงดังรูปที่ 3.10



รูปที่ 3.10 โมเดล CNN model 9 layers

โดยโมเดลทั้งหมดที่นำมาใช้ฝึกสอนนั้นจะใช้ตัวแปรอัตราการเรียนรู้ชนิด Adam โดยใช้ค่า Learning rate = 0.0001

### 3.6 ขั้นตอนของการทดสอบโมเดลที่ฝึกสอนมาแล้วกับชุดข้อมูลทดสอบ

ในขั้นตอนของการทดสอบโมเดลทางผู้วิจัยได้แบ่งข้อมูลของชนิดโรคพืชเป็น 2 ชนิด ได้แก่ โรคพืชใบอู่ และโรคพืชใบมันฝรั่ง วิธีทดสอบนั้นจะเป็นการแบ่งทดสอบของพืชแต่ละชนิดและหาเปอร์เซ็นต์ความถูกต้องในการตัดสินใจของโมเดลแต่ละโมเดล โดยภาพของชุดข้อมูลที่นำมาทดสอบนั้นจะเป็นภาพที่ผ่านการทำ Images augmentation มารวมกับภาพต้นฉบับที่ยังไม่ถูกนำไปทำการเปลี่ยนแปลงองค์ประกอบของข้อมูลใด ๆ โดยในการที่นำภาพที่ถูกทำ Images augmentation มารวมกับภาพต้นฉบับนั้นเนื่องจากเหตุผลที่ว่า การทำ Images augmentation นั้นคือการเพิ่มจำนวนข้อมูลให้กับชุดหรือฐานข้อมูลที่จะถูกนำมาใช้นั้นมีจำนวนจำกัดหรือไม่เพียงพอต่อการนำมาฝึกสอนเลยจำเป็นที่จะต้องมีการนำภาพที่ทำ Images augmentation นั้นมารวมกับภาพต้นฉบับด้วย

## บทที่ 4

### ผลการทดลองของงานวิจัย

ในการทดสอบประสิทธิภาพของงานวิจัยที่นำเสนอได้ทำการทดลองบนคอมพิวเตอร์โน้ตบุ๊ก Dell G3 15 3590 Gaming Notebook ระบบปฏิบัติการ Windows 10 CPU Intel(R) Core(TM) i5-9300H 2.40 GHz หน่วยความจำ Ram 32 GB หน่วยประมวลผลภาพการ์ดจอ GPU NVIDIA Geforce GTX 1650 20 GB

ชุดข้อมูลภาพที่นำมาใช้ในการทดลองนำภาพมาจากรฐานข้อมูลภาพออนไลน์ Kaggle.com [16] และ PlantVillage [17] ด้วยจะแบ่งเป็นกลุ่มโรคพืชใบของุ่นและกลุ่มโรคพืชใบมันฝรั่ง

#### 4.1 ชุดข้อมูลภาพที่ใช้ในการทดลอง

##### 4.1.1 ชุดข้อมูลทดลองสำหรับการแบ่งส่วนบริเวณโรค

ในการทดลองของขั้นตอนการแบ่งส่วนบริเวณโรคนั้นข้อมูลที่นำมาใช้ในการทดลองหรือใช้สำหรับฝึกฝนนั้นเป็นข้อมูลที่เกิดจากการนำภาพมาทำซูเปอร์พิกเซลตามขั้นตอนในรูปที่ 3.3 โดยภาพที่ได้จากขั้นตอนนั้นจะเป็นภาพของบริเวณส่วนที่เป็นโรคและบริเวณของส่วนใบไม้ที่เป็นเป็นโรคขนาด 80x80 พิกเซล แสดงตัวอย่างรูปที่ 4.1 และรูปที่ 4.2 อย่างละ 300 ภาพ



รูปที่ 4.1 ตัวอย่างภาพบริเวณโรคสำหรับฝึกสอนแบ่งส่วน



รูปที่ 4.2 ตัวอย่างภาพบริเวณที่ไม่เป็นโรคสำหรับฝึกสอนแบ่งส่วน

#### 4.1.2 ชุดข้อมูลทดลองสำหรับการจำแนกโรค

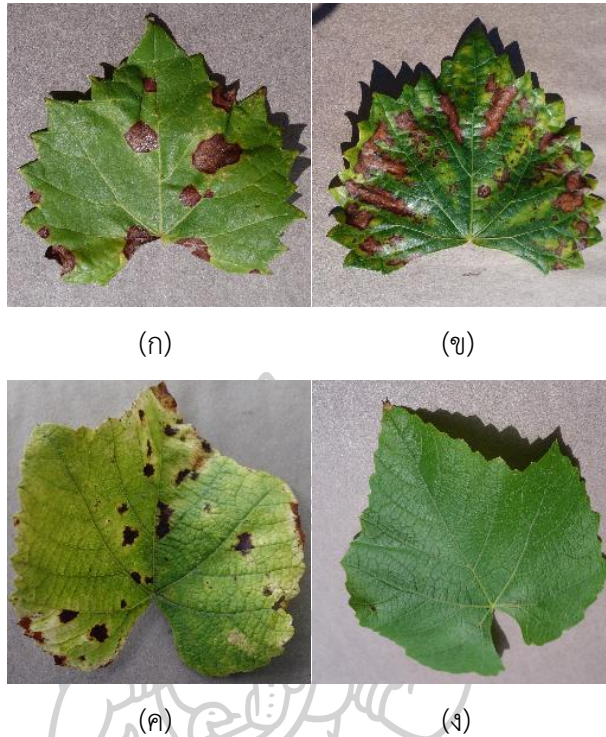
ข้อมูลภาพที่นำมาใช้สำหรับทดสอบประสิทธิภาพของโมเดลจะแบ่งออกเป็น 2 กลุ่มชนิดของใบพืช ได้แก่ ข้อมูลภาพกลุ่มที่หนึ่ง กลุ่มของใบอ่อนจะประกอบไปด้วยชุดภาพทดสอบ 4 ชุดของชนิดโรค ได้แก่ 1. Black rot จำนวน 40 ภาพ 2. Black measles จำนวน 40 ภาพ 3. Leaf blight จำนวน 40 ภาพ และ 4. Healthy จำนวน 40 ภาพ

กลุ่มข้อมูลภาพทดสอบกลุ่มที่สอง เป็นกลุ่มของใบมันฝรั่ง ประกอบไปด้วยชุดภาพทดสอบ 3 ชุดของชนิดโรค ได้แก่ 1. Early blight จำนวน 40 ภาพ 2. Late blight จำนวน 40 ภาพ 3. Healthy จำนวน 40 ภาพ

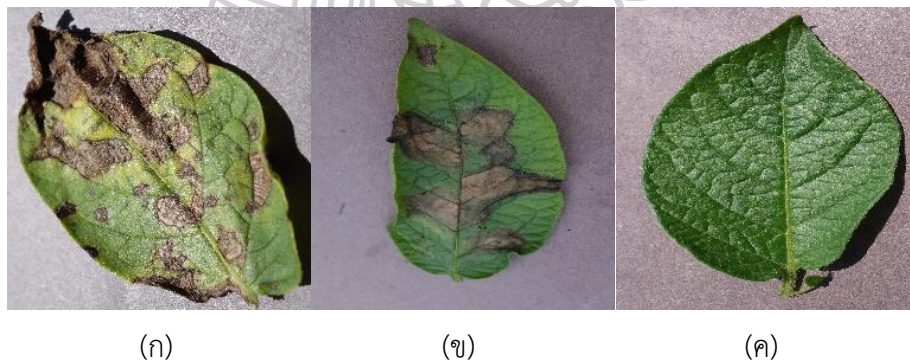
ในการทดลองนี้ได้แบ่งกลุ่มข้อมูลภาพที่ใช้สำหรับฝึกสอนโมเดลออกเป็น 4 กลุ่ม ได้แก่ กลุ่มข้อมูลภาพปกติ (Original) แสดงตัวอย่างดังรูปที่ 4.3 และรูปที่ 4.4 กลุ่มข้อมูลภาพปกติรวมกับภาพที่เพิ่มจำนวนด้วยวิธี Global augmentation (Original + Global) แสดงตัวอย่างดังรูปที่ 4.5 กลุ่มข้อมูลภาพปกติรวมกับกลุ่มข้อมูลภาพที่เพิ่มข้อมูลแบบเชิงพื้นที่ (Original + LoBIA) แสดงตัวอย่างดังรูปที่ 4.6 และกลุ่มสุดท้ายเป็นกลุ่มของข้อมูลภาพรวมกับกลุ่มข้อมูลภาพที่นำภาพจากการทำ LoBIA มาเพิ่มจำนวนด้วยวิธี Global augmentation (Original + LoBIAGlobal) แสดงตัวอย่างดังรูปที่ 4.7



Original



รูปที่ 4.3 ตัวอย่างชุดภาพฝักสอน (Original) ของใบองุ่น (ก) Black rot (ข) Black measles (ค) Leaf blight (ง) Healthy



รูปที่ 4.4 ตัวอย่างชุดภาพฝักสอน (Original) ของใบมันฝรั่ง (ก) Early blight (ข) Late blight (ค) Healthy

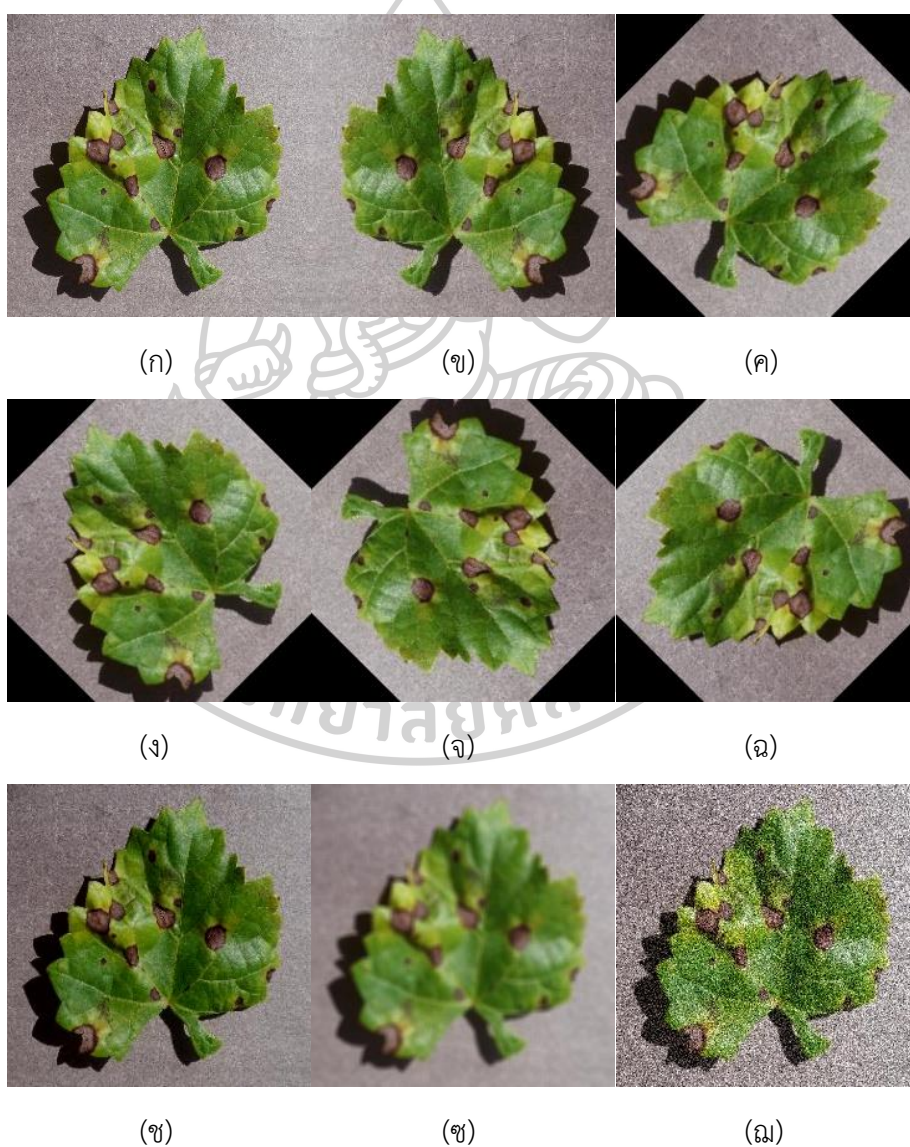
จากรูปที่ 4.3 เป็นรูปของชุดข้อมูลภาพฝักสอนของใบองุ่นที่เป็นภาพต้นฉบับ มีจำนวนภาพฝักสอนทั้งหมด 795 ภาพ แบ่งออกเป็น รูปที่ 4.3(ก) ภาพของโรค Black rot จำนวน 198 ภาพ รูปที่



4.3(ข) ภาพของโรค Black measles จำนวน 200 ภาพ รูปที่ 4.3(ค) ภาพของโรค Leaf blight จำนวน 197 ภาพ และรูปที่ 4.3(ง) ภาพของใบองุ่นปกติ Healthy จำนวน 200 ภาพ

จากรูปที่ 4.4 เป็นรูปของชุดข้อมูลภาพฝึกสอนของไวน์ฝรั่งเศสที่เป็นภาพต้นฉบับ มีจำนวนภาพฝึกสอนทั้งหมด 450 ภาพ แบ่งออกเป็น รูปที่ 4.4(ก) ภาพของโรค Early blight จำนวน 150 ภาพ รูปที่ 4.4(ข) ภาพของโรค Late blight จำนวน 150 ภาพ และรูปที่ 4.4(ค) ภาพของไวน์ฝรั่งเศสปกติ Healthy จำนวน 150 ภาพ

#### Original + Global



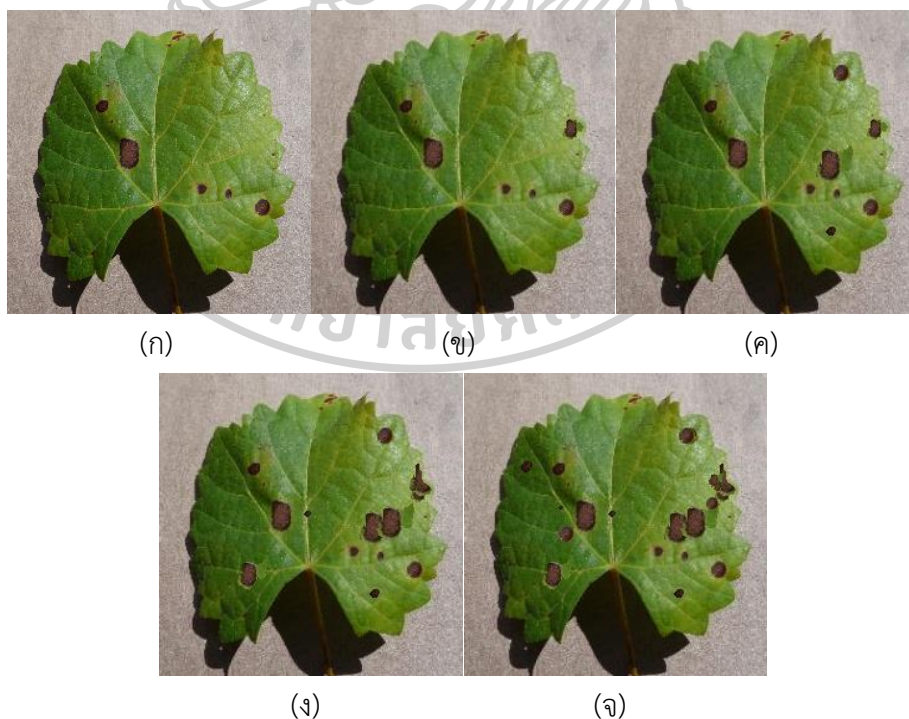
รูปที่ 4.5 ตัวอย่างชุดภาพฝึกสอน (Original + Global) ของใบองุ่น

จากรูปที่ 4.5 เป็นภาพตัวอย่างของชุดภาพฝึกสอนที่เกิดจากการรวมกันระหว่างภาพต้นฉบับในรูปที่ 4.5(ก) และภาพที่ถูกเพิ่มจำนวนขึ้นด้วยการทำ Global augmentation ในรูปที่ 4.5(ข) ถึงรูปที่ 4.5(ฉ) โดยการทดลองนี้ได้ทำการเพิ่มจำนวนภาพชุดฝึกสอนด้วย Global augmentation 8 รูปแบบ ได้แก่ (ข) Horizontal flip (ค) Rotate 45° (ง) Rotate -45° (จ) Rotate 135° (ฉ) Rotate -135° (ช) Low brightness (ซ) Blur และ (ฅ) Gaussian noise

ในกลุ่มของชุดข้อมูลฝึกสอนนี้ (Original + Global) ชุดข้อมูลภาพฝึกสอนของใบองุ่น มีจำนวนภาพฝึกสอนทั้งหมด 7,155 ภาพ แบ่งออกเป็น ภาพของโรค Black rot จำนวน 1,782 ภาพ ภาพของโรค Black measles จำนวน 1,800 ภาพ ภาพของโรค Leaf blight จำนวน 1,773 ภาพ และภาพของใบองุ่นปกติ Healthy จำนวน 1,800 ภาพ

กลุ่มของชุดข้อมูลภาพฝึกสอนของใบมันฝรั่ง (Original + Global) มีจำนวนภาพฝึกสอนทั้งหมด 4,050 ภาพ แบ่งออกเป็น ภาพของโรค Early blight จำนวน 1,350 ภาพ ภาพของโรค Late blight จำนวน 1,350 ภาพ และภาพของใบมันฝรั่งปกติ Healthy จำนวน 1,350 ภาพ

#### Original + LoBIA



รูปที่ 4.6 ตัวอย่างชุดภาพฝึกสอน (Original + LoBIA) ของใบองุ่น (ก) ภาพต้นฉบับ (ข) ภาพ LoBIA 5% (ค) ภาพ LoBIA 10% (ง) ภาพ LoBIA 15% (จ) ภาพ LoBIA 20%

จากรูปที่ 4.6 แสดงตัวอย่างของภาพจากชุดข้อมูลในกลุ่มของชุดข้อมูลฝึกสอนที่เกิดจากการรวมกันระหว่างภาพต้นฉบับในรูปที่ 4.6(ก) กับภาพที่ถูกนำไปเพิ่มจำนวนข้อมูลโรคเชิงพื้นที่ หรือ LoBIA ในรูปที่ 4.6(ข) ถึงรูปที่ 4.6(จ) โดยจะเห็นได้ว่าภาพที่ถูกนำไปทำ LoBIA นั้นจะมีลักษณะการกระจายตัวของบริเวณโรคเพิ่มมากขึ้น โดยในงานวิจัยนี้ได้มีการกำหนดค่าความหนาแน่นของการกระจายตัวของโรคไว้ที่ 5%, 10%, 15% และ 20% ของพื้นที่ไปไม่ทั้งหมด

การเพิ่มจำนวนภาพที่มีการกระจายตัวของบริเวณโรคที่เพิ่มขึ้นนั้นจะขึ้นอยู่กับเปอร์เซ็นต์ความหนาแน่นของภาพต้นฉบับ โดยจะแบ่งออกได้เป็น 5 เงื่อนไข

เงื่อนไขที่ 1 : กรณีที่เปอร์เซ็นต์ความหนาแน่นของโรคบนภาพต้นฉบับนั้นมีไม่ถึงหรือต่ำกว่า 5% โปรแกรมก็จะทำการสุ่มกระจายโรคลงบนบริเวณไปไม่ ให้ความหนาแน่นของโรคบนไปนั้นมีค่าใกล้เคียงแต่ไม่เกิน 5%, 10%, 15% และ 20% เท่ากับว่าจะได้จำนวนภาพที่เกิดการกระจายโรคเพิ่มขึ้นจำนวน 4 ภาพ ในรูปที่ 4.6(ข) ถึงรูปที่ 4.6(จ)

เงื่อนไขที่ 2 : กรณีที่เปอร์เซ็นต์ความหนาแน่นของโรคบนภาพต้นฉบับนั้นมีค่ามากกว่า 5% แต่น้อยกว่า 10% โปรแกรมก็จะทำการสุ่มกระจายโรคลงบนบริเวณไปไม่ ให้ความหนาแน่นของโรคบนไปนั้นมีค่าใกล้เคียงแต่ไม่เกิน 10%, 15% และ 20% เท่ากับว่าจะได้จำนวนภาพที่เกิดการกระจายโรคเพิ่มขึ้นจำนวน 3 ภาพ ในรูปที่ 4.6(ค) ถึงรูปที่ 4.6(จ)

เงื่อนไขที่ 3 : กรณีที่เปอร์เซ็นต์ความหนาแน่นของโรคบนภาพต้นฉบับนั้นมีค่ามากกว่า 10% แต่น้อยกว่า 15% โปรแกรมก็จะทำการสุ่มกระจายโรคลงบนบริเวณไปไม่ ให้ความหนาแน่นของโรคบนไปนั้นมีค่าใกล้เคียงแต่ไม่เกิน 15% และ 20% เท่ากับว่าจะได้จำนวนภาพที่เกิดการกระจายโรคเพิ่มขึ้นจำนวน 2 ภาพ ในรูปที่ 4.6(ง) ถึงรูปที่ 4.6(จ)

เงื่อนไขที่ 4 : กรณีที่เปอร์เซ็นต์ความหนาแน่นของโรคบนภาพต้นฉบับนั้นมีค่ามากกว่า 15% แต่น้อยกว่า 20% โปรแกรมก็จะทำการสุ่มกระจายโรคลงบนบริเวณไปไม่ ให้ความหนาแน่นของโรคบนไปนั้นมีค่าใกล้เคียงแต่ไม่เกิน 20% เท่ากับว่าจะได้จำนวนภาพที่เกิดการกระจายโรคเพิ่มขึ้นจำนวน 1 ภาพ ในรูปที่ 4.6(จ)

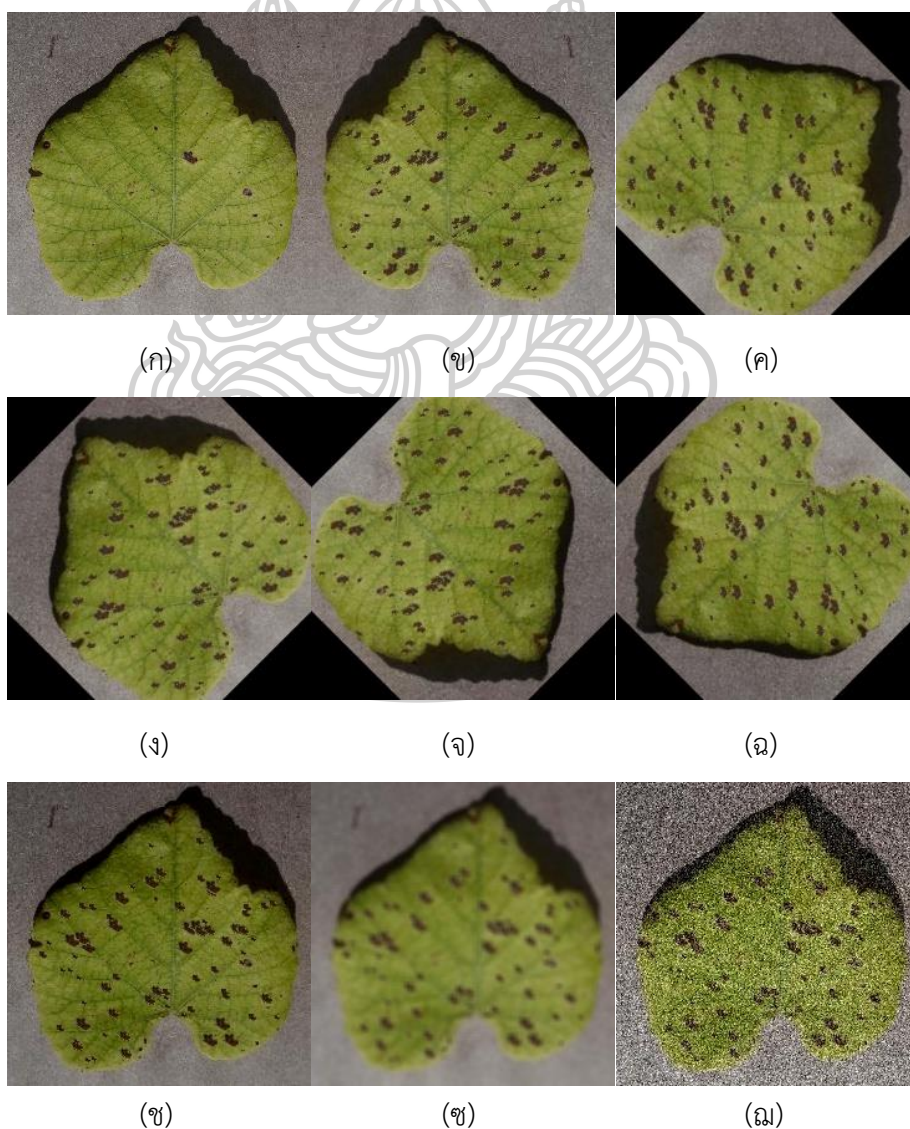
เงื่อนไขที่ 5 : กรณีที่เปอร์เซ็นต์ความหนาแน่นของโรคบนภาพต้นฉบับนั้นมีค่ามากกว่า 20% อยู่แล้ว โปรแกรมก็จะมองว่าภาพต้นฉบับนั้นไม่จำเป็นต้องทำการกระจายบริเวณโรคเพิ่มขึ้น ก็จะไม่มีการเพิ่มภาพที่ได้จากการทำ LoBIA ในส่วนนี้



ในกลุ่มของชุดข้อมูลฝึกสอนนี้ (Original + Local) ชุดข้อมูลภาพฝึกสอนของใบองุ่น มีจำนวนภาพฝึกสอนทั้งหมด 2,443 ภาพ แบ่งออกเป็น ภาพของโรค Black rot จำนวน 841 ภาพ ภาพของโรค Black measles จำนวน 570 ภาพ ภาพของโรค Leaf blight จำนวน 832 ภาพ และ ภาพของใบองุ่นปกติ Healthy จำนวน 200 ภาพ

กลุ่มของชุดข้อมูลภาพฝึกสอนของใบมันฝรั่ง (Original + Local) มีจำนวนภาพฝึกสอนทั้งหมด 995 ภาพ แบ่งออกเป็น ภาพของโรค Early blight จำนวน 405 ภาพ ภาพของโรค Late blight จำนวน 440 ภาพ และภาพของใบมันฝรั่งปกติ Healthy จำนวน 150 ภาพ

#### Original + LoBIAGlobal



รูปที่ 4.7 ตัวอย่างชุดภาพฝึกสอน (Original + LoBIAGlobal) ของใบองุ่น

จากรูปที่ 4.7 แสดงตัวอย่างของภาพจากชุดข้อมูลในกลุ่มของชุดข้อมูลฝึกสอนที่เกิดจากการรวมกันระหว่างภาพต้นฉบับในรูปที่ 4.7(ก) กับภาพ LoBIA ที่ถูกนำไปเพิ่มจำนวนข้อมูลภาพด้วยการทำ Global augmentation ในรูปที่ 4.7(ข) ถึงรูปที่ 4.7(ฉ) โดยจะเป็นการนำเอาภาพที่ถูกกระจายโรคด้วยการทำ LoBIA แล้วนั้นมาเพิ่มจำนวนภาพให้มากขึ้นด้วยวิธี Global augmentation เพื่อต้องการให้เกิดความหลากหลายของข้อมูลทั้งสองรูปแบบ และยังเป็นการเพิ่มจำนวนภาพสำหรับฝึกสอนโมเดลให้มากขึ้น

ในกลุ่มของชุดข้อมูลฝึกสอนนี้ (Original + LoBIAGlobal) ชุดข้อมูลภาพฝึกสอนของใบองุ่น มีจำนวนภาพฝึกสอนทั้งหมด 15,579 ภาพ แบ่งออกเป็น ภาพของโรค Black rot จำนวน 5,342 ภาพ ภาพของโรค Black measles จำนวน 3,160 ภาพ ภาพของโรค Leaf blight จำนวน 5,277 ภาพ และภาพของใบองุ่นปกติ Healthy จำนวน 1,800 ภาพ

กลุ่มของชุดข้อมูลภาพฝึกสอนของใบมันฝรั่ง (Original + Global) มีจำนวนภาพฝึกสอนทั้งหมด 6,010 ภาพ แบ่งออกเป็น ภาพของโรค Early blight จำนวน 2,190 ภาพ ภาพของโรค Late blight จำนวน 2,470 ภาพ และภาพของใบมันฝรั่งปกติ Healthy จำนวน 1,350 ภาพ

#### 4.2 การวัดผลค่าความแม่นยำของการแบ่งส่วนบริเวณที่เป็นโรคด้วย F1-score

ในการวัดผลหาค่าความแม่นยำของการแบ่งส่วนบริเวณที่เป็นโรคด้วย F1-score เพราะเนื่องจากคำตอบของการทำนายเป็นได้เพียงแค่ 2 คำตอบ คือ จริง (True) กับเท็จ (False) โดย F1-score เป็นการหาค่าเฉลี่ยฮาร์โมนิกของค่าความแม่นยำ (Precision) ในการทำนายและค่าการเรียกคืน (Recall) กล่าวคือเมื่อมีการทำนายบริเวณที่เป็นโรคแล้วควรจะมีค่าการเรียกคืนกลับมา 100% นั้นหมายความว่าพื้นที่ที่ทำนายนั้นเป็นพื้นที่ที่เป็นโรคและมีการเรียกคืนค่าทั้งหมด แต่กลับกันถ้าหากพื้นที่ที่ทำนายนั้นไม่ได้เป็นพื้นที่ของโรคค่าของการเรียกคืนก็จะลดลงไป โดยที่มาของสมการของ F1-score เกิดจากสมการที่ (4.1) ถึง (4.4)

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4.3)$$

$$\begin{aligned}
 F1 &= \frac{2 \times \left( \frac{TP}{TP + FP} \right) \left( \frac{TP}{TP + FN} \right)}{\left( \frac{TP}{TP + FP} \right) + \left( \frac{TP}{TP + FN} \right)} \\
 &= \frac{2TP}{2TP + FP + FN}
 \end{aligned}
 \tag{4.4}$$

โดยที่  $TP$  = True Positive คือ คลาสของโรคที่คาดการณ์ไว้เป็นจริง

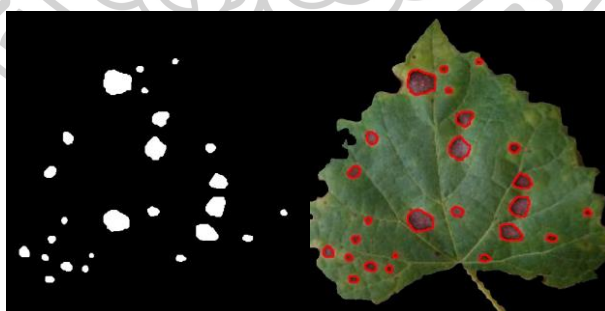
$FP$  = False Positive คือ คลาสของโรคที่คาดการณ์ไว้เป็นเท็จ

$FN$  = False Negative คือ คลาสไม่เป็นโรคที่คาดการณ์ไว้เป็นเท็จ

#### 4.3 ผลการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ได้ฝึกสอนมาแล้ว

ในการทดสอบวัดผลค่าความแม่นยำของการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model จากขั้นตอนในบล็อกไดอะแกรมรูปที่ 3.6 โดยทำการปรับเปลี่ยนค่าพารามิเตอร์ค่าความหนาแน่นกลุ่มข้อมูล  $K$  ของ SLIC superpixel โดยทดลองปรับค่าความหนาแน่นที่  $K = 500, 600, 700$  และ 800

ในการทดลองความแม่นยำของการแบ่งส่วนได้มีการทดลองกับภาพที่สร้างขึ้นเอง มีวัตถุประสงค์เพื่อทดลองให้เห็นอย่างชัดเจนถึงบริเวณที่เกิดการจำแนกด้วยโมเดลที่ออกแบบ โดยได้มีการสร้างภาพขาว-ดำที่เป็นบริเวณของการเกิดโรคกับชุดภาพใบไม้ที่นำมาใช้ทดสอบเช่นกัน แสดงดังรูปที่ 4.8



(ก)

(ข)

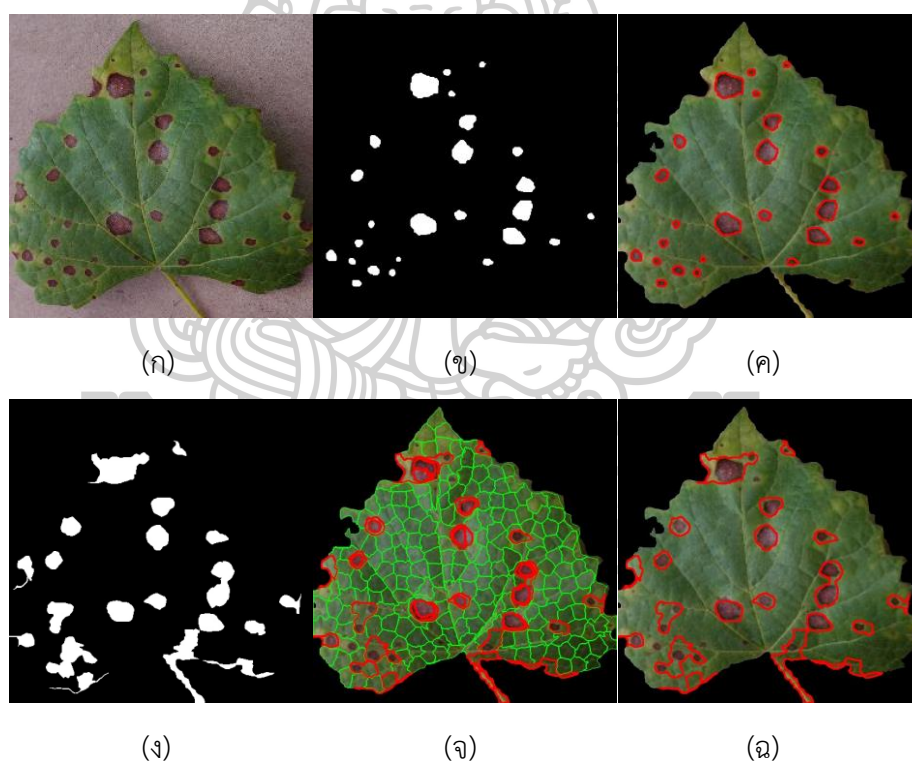
รูปที่ 4.8 ภาพตัวอย่างของภาพทดสอบการแบ่งส่วนโรค (ก) ภาพตัวอย่างขาว-ดำที่สร้างสำหรับทดสอบ (ข) ผลลัพธ์ที่ควรเป็น

จากรูปที่ 4.8 ทางด้านซ้ายจะเป็นภาพขาว-ดำที่นำมาใช้เป็นภาพเปรียบเทียบกับภาพขาว-ดำที่ได้ด้วยการใช้ CNN model ประกอบไปด้วย 2 บริเวณ ดังต่อไปนี้

บริเวณที่ 1 : สีดำ แทนบริเวณของพื้นที่ส่วนที่ไม่ใช่โรค และรวมไปถึงบริเวณของพื้นหลัง มีความเข้มสีเท่ากับ 0

บริเวณที่ 2 : สีขาว แทนที่บริเวณที่เกิดการแบ่งส่วนว่าเป็นบริเวณของการเกิดโรค ซึ่งจะเป็นบริเวณที่นำเอามาคำนวณเปรียบเทียบกัน มีความเข้มสีเท่ากับ 1

ในการทดลองของการแบ่งส่วนภาพ ภาพที่นำมาทดลองในส่วนนี้เป็นภาพของใบพืช 2 ชนิดผสมกัน ได้แก่ ใบองุ่นจำนวน 58 ภาพ และใบมันฝรั่งจำนวน 42 ภาพ โดยจะเป็นภาพที่ยังมีพื้นหลังอยู่ แล้วนำมาเข้ากระบวนการตามบล็อกไดอะแกรมรูปที่ 3.6 บล็อกไดอะแกรมขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรครูปที่ 3.6 โดยผลลัพธ์ที่ได้จากการแบ่งส่วนจำแนกหาบริเวณที่เป็นโรคนั้น แสดงผลลัพธ์ดังรูปที่ 4.9 ถึงรูปที่ 4.12



รูปที่ 4.9 ภาพของการแบ่งส่วนโรค ที่  $K = 500$  (ก) ภาพต้นฉบับ (ข) ภาพขาว-ดำสำหรับทดสอบ (ค) ผลลัพธ์ที่ควรเป็น (ง) ภาพขาว-ดำที่ได้จากการแบ่งส่วน (จ) SLIC superpixel ที่  $K = 500$  (ฉ) ผลลัพธ์ของการแบ่งส่วน

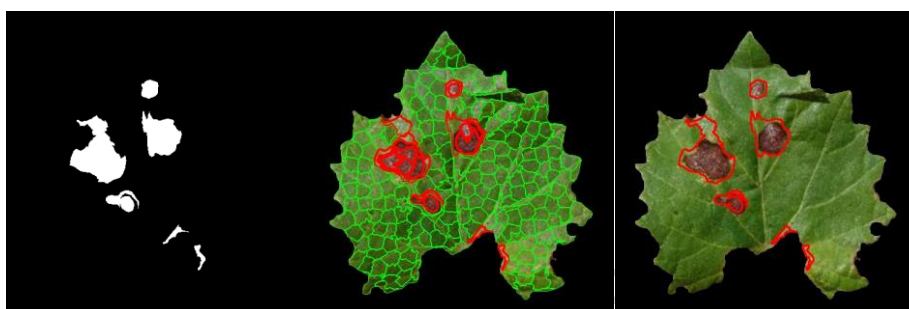




(ก)

(ข)

(ค)

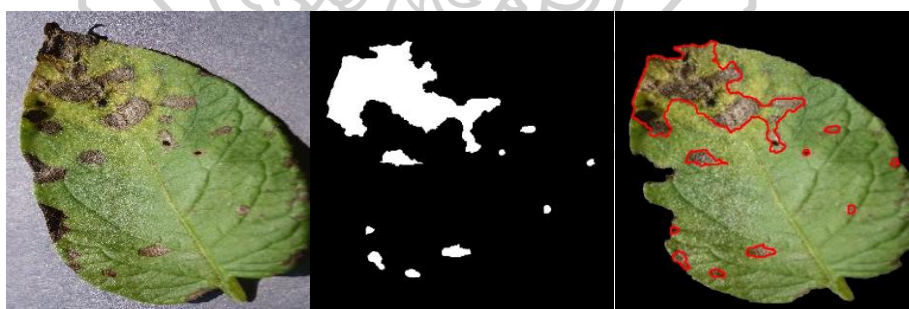


(ง)

(จ)

(ฉ)

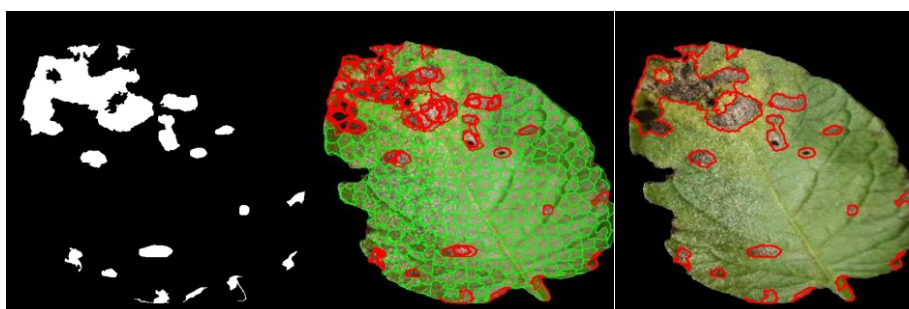
รูปที่ 4.10 ภาพของการแบ่งส่วนโรค ที่  $K = 600$  (ก) ภาพต้นฉบับ (ข) ภาพขาว-ดำสำหรับทดสอบ (ค) ผลลัพธ์ที่ควรเป็น (ง) ภาพขาว-ดำที่ได้จากการแบ่งส่วน (จ) SLIC superpixel ที่  $K = 600$  (ฉ) ผลลัพธ์ของการแบ่งส่วน



(ก)

(ข)

(ค)

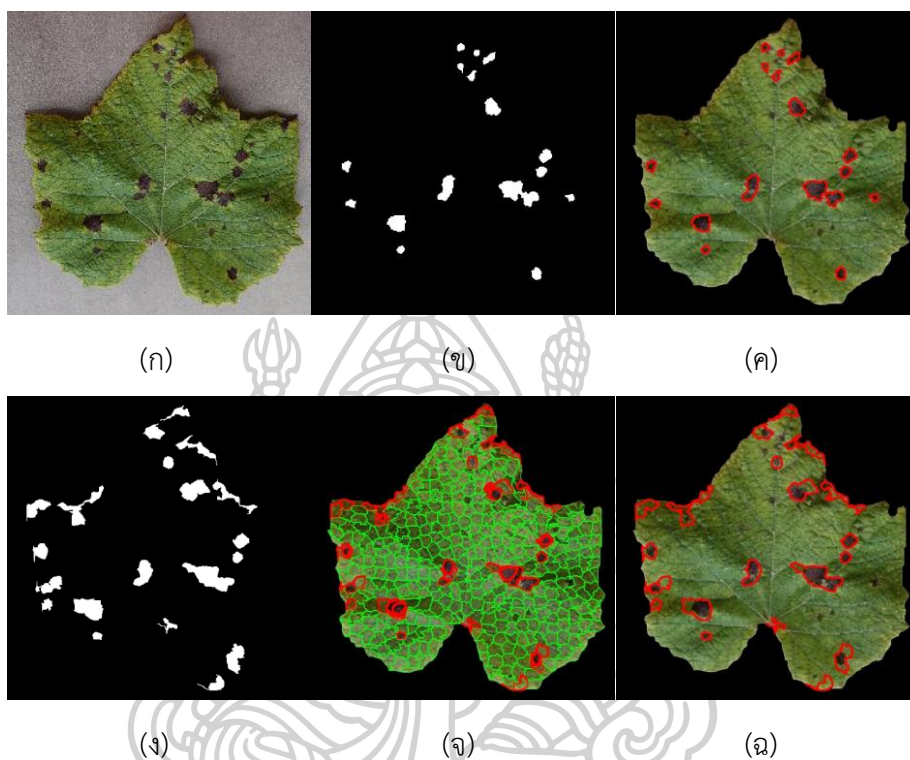


(ง)

(จ)

(ฉ)

รูปที่ 4.11 ภาพของการแบ่งส่วนโรค ที่  $K = 700$  (ก) ภาพต้นฉบับ (ข) ภาพขาว-ดำสำหรับทดสอบ  
 (ค) ผลลัพธ์ที่ควรเป็น (ง) ภาพขาว-ดำที่ได้จากการแบ่งส่วน (จ) SLIC superpixel ที่  $K = 700$   
 (ฉ) ผลลัพธ์ของการแบ่งส่วน



รูปที่ 4.12 ภาพของการแบ่งส่วนโรค ที่  $K = 800$  (ก) ภาพต้นฉบับ (ข) ภาพขาว-ดำสำหรับทดสอบ  
 (ค) ผลลัพธ์ที่ควรเป็น (ง) ภาพขาว-ดำที่ได้จากการแบ่งส่วน (จ) SLIC superpixel ที่  $K = 800$   
 (ฉ) ผลลัพธ์ของการแบ่งส่วน

จากรูปที่ 4.9 ถึงรูปที่ 4.12 ในรูป (จ) แสดงถึงความหนาแน่นของซูเปอร์พิกเซล เมื่อปรับค่าความหนาแน่นให้มากขึ้น จำนวนซูเปอร์พิกเซลก็มากขึ้นตามไปด้วย และทำให้สามารถพิจารณาในบริเวณของโรคที่มีขนาดเล็กได้ โดยจะเห็นได้ว่ารูปที่ 4.12(จ) ที่  $K = 800$  มีความหนาแน่นของซูเปอร์พิกเซลมากกว่าและแบ่งส่วนบริเวณของโรคที่เล็กได้ดีกว่ารูปที่ 4.9(จ)

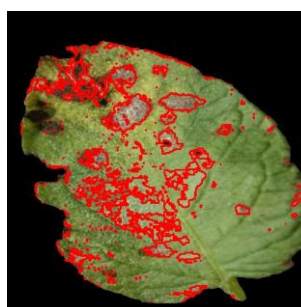
ตารางที่ 4.1 ตารางเปรียบเทียบผลค่า  $K = 500, 600, 700, 800$ 

Methods		F1-score			
		$\mu$	<i>S.D.</i>	min	max
Sachin's method [43]		0.386	0.208	0.013	0.819
Our's method	<i>K</i>				
	500	0.647	0.1603	0.0985	0.9351
	600	0.680	0.1395	0.1194	0.9407
	700	0.703	0.1320	0.1220	0.9390
	<b>800</b>	<b>0.705</b>	<b>0.1310</b>	<b>0.1498</b>	<b>0.9420</b>

ตารางที่ 4.1 เป็นตารางเปรียบเทียบค่าความแม่นยำของการแบ่งส่วนบริเวณโรคของค่าความหนาแน่นของซูปเปอร์พิกเซลแต่ละค่า โดยในการเปรียบเทียบจะเป็นการเปรียบเทียบค่าความเหมือนกันของพื้นที่สีขาวระหว่างภาพทดสอบกับภาพที่ได้จากโมเดลด้วยสมการ F1-score ตามสมการที่ (4.4) โดยจะเป็นการคำนวณเปรียบเทียบหาค่าเฉลี่ย (Mean :  $\mu$ ) ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation : *S.D.*) ค่าสูงสุด (Maximum : **max**) และค่าต่ำสุด (Minimum : **min**) ซึ่งผลที่ได้ระบุว่าเมื่อปรับเพิ่มจำนวนของซูปเปอร์พิกเซลที่  $K = 800$  สามารถแบ่งส่วนได้ดี โดยจากข้อมูลทดสอบ 100 ภาพ สามารถให้ข้อมูลใกล้เคียงกับภาพทดสอบได้ถูกต้องสูงสุดที่ 94.20% ถูกต่อน้อยสุดที่ 14.98% โดยคิดเฉลี่ยทั้ง 100 ภาพ สามารถให้ค่าความถูกต้องเฉลี่ยที่ 70.50% คิดเป็นส่วนเบี่ยงเบนมาตรฐาน 13.10% และจะสังเกตได้ว่าเมื่อเทียบกับผลทดลองในงานวิจัยอ้างอิง [43] โดยใช้วิธีการจัดกลุ่มแบบ K-mean ให้ผลที่ดีกว่าในทุก ๆ การปรับเพิ่มจำนวนของซูปเปอร์พิกเซล โดยได้แสดงผลลัพธ์เปรียบเทียบของการจัดกลุ่มระหว่างการจัดกลุ่มแบบซูปเปอร์พิกเซลและการจัดกลุ่มแบบ K-mean แสดงดังรูปที่ 4.13



(ก)



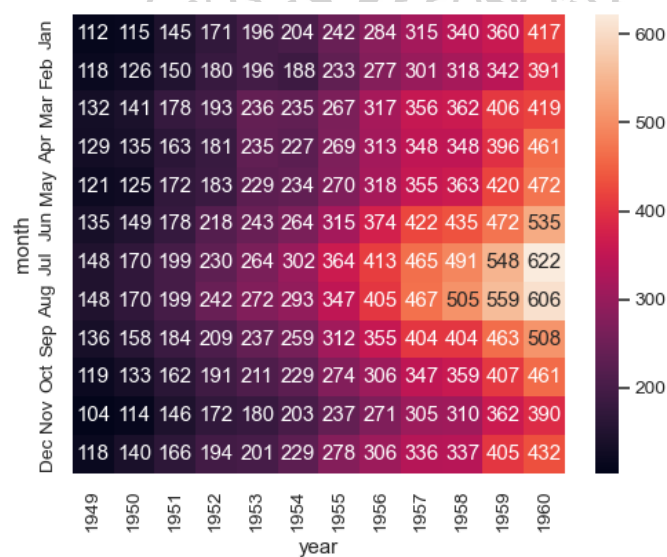
(ข)

รูปที่ 4.13 ผลลัพธ์ของการจัดกลุ่ม (ก) วิธีซูปเปอร์พิกเซล (ข) วิธี K-mean ของ Sachin [43]

#### 4.4 ผลการจำแนกโรคด้วยโมเดลการเรียนรู้เชิงลึก

ในการทดสอบประสิทธิภาพของโมเดลในการจำแนกชนิดของโรคในงานวิจัยได้ใช้โมเดลการเรียนรู้เชิงลึกทั้งหมด 4 โมเดลมาใช้เรียนรู้กับชุดข้อมูลภาพฝึกสอนที่เตรียมไว้ โดยโมเดลที่นำมาใช้ประกอบไปด้วย CNN model (Hidden layer 7 layers) ในรูปที่ 3.9, CNN model (Hidden layer 9 layers) ในรูปที่ 3.10, VGG19 ในรูปที่ 2.31 และ MobileNet ในรูปที่ 2.32 โดยในการทดลองจะเป็นการนำโมเดลทั้งสี่มาประมวลผลกับชุดข้อมูลภาพชุดเดียวกันของแต่ละชนิดพืช และเปรียบเทียบผลลัพธ์ของเปอร์เซ็นต์ค่าความถูกต้องในการจำแนกชนิดโรคพืช

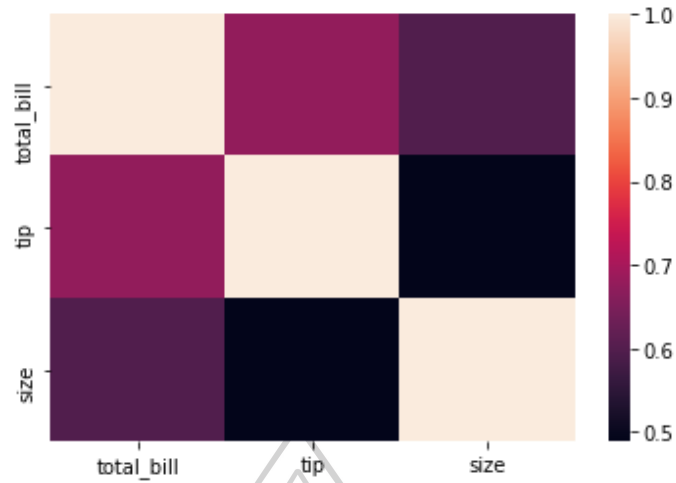
ในการวัดประสิทธิภาพของการจำแนกชนิดโรคในขั้นตอนนี้จะแสดงในรูปแบบของการทำ Data visualization Heatmap แสดงในรูปที่ 4.14 และการจัดกลุ่มแบบจุด (Swarm plot) แสดงในรูปที่ 4.16



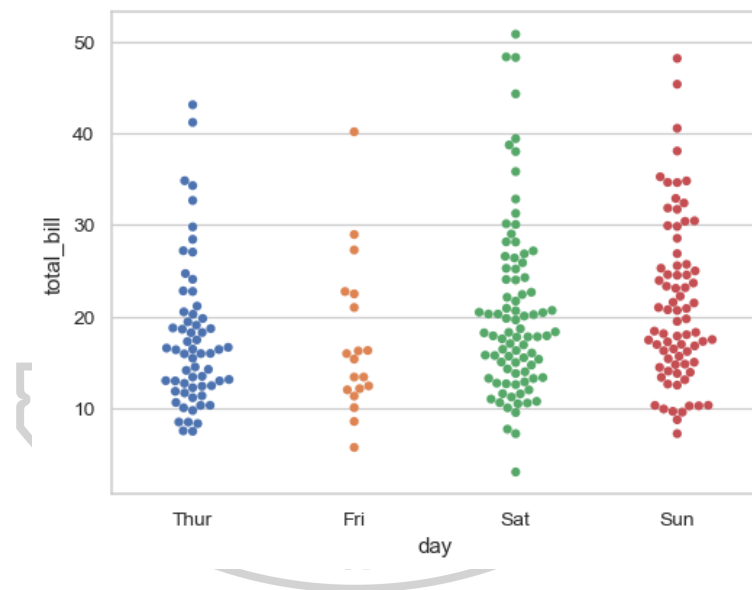
รูปที่ 4.14 ตัวอย่างการทำ Data visualization แสดงข้อมูล Heat map

ที่มา : [https://seaborn.pydata.org/\\_images/seaborn-heatmap-5.png](https://seaborn.pydata.org/_images/seaborn-heatmap-5.png)

จากรูปที่ 4.14 การแสดงข้อมูลของ Heat map นั้นจะเป็นการแสดงข้อมูลของค่าความเข้มที่อยู่ภายในแต่ละช่อง โดยจะเกิดจากการดูข้อมูลในแกน  $y$  เทียบกับแกน  $x$  แต่ถ้าหากข้อมูลนั้นมีการนำไปทำ Correlation matrix ก่อนมาแสดง Heat map ขนาดของ Heat map จะมีรูปร่างขนาด  $m \times m$  ซึ่งผลลัพธ์ที่ได้ควรจะมีค่าสีที่เป็นสีเดียวกันในแนวทแยง แสดงในรูปที่ 4.15 ซึ่งจะหมายความว่าข้อมูลในแนวแกน  $y$  มีความสัมพันธ์กับข้อมูลในแนวแกน  $x$  มากน้อยเพียงใด



รูปที่ 4.15 ตัวอย่างภาพ Heat map ของข้อมูลที่ทำ Correlation matrix

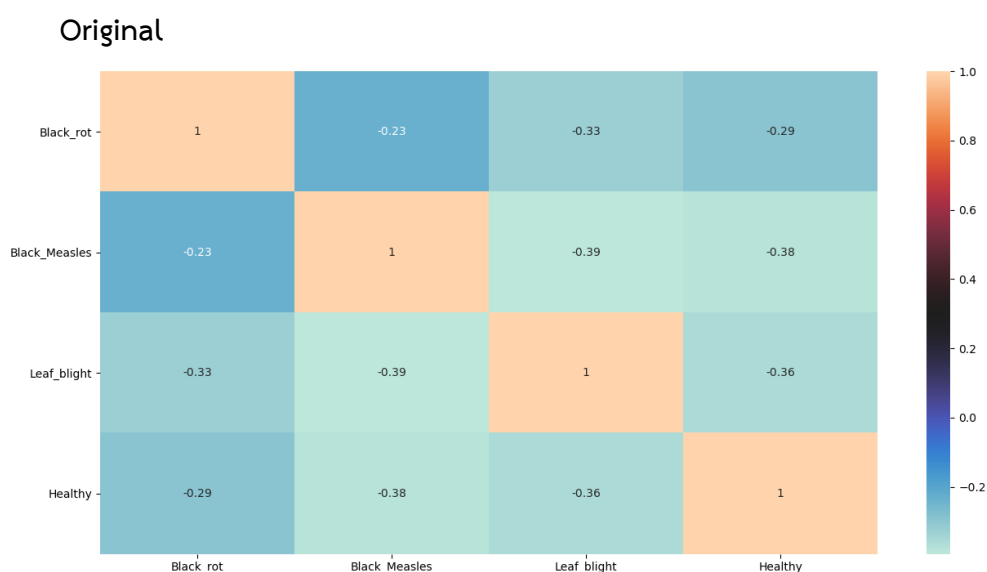


รูปที่ 4.16 ตัวอย่างการทำ Data visualization จัดกลุ่มข้อมูลด้วย Swarm plot  
ที่มา : [https://seaborn.pydata.org/\\_images/seaborn-swarmplot-2.png](https://seaborn.pydata.org/_images/seaborn-swarmplot-2.png)

สำหรับรูปที่ 4.16 เป็นการแสดงภาพการทำ Data visualization แบบ Swarm plot ซึ่งเป็นการจัดกลุ่มประเภทของข้อมูลที่เป็นประเภทเดียวกัน หรือแสดงถึงผลการจำแนกข้อมูล โดยผลของการแสดงที่ถูกต้อง ข้อมูลที่เป็นชุดเดียวกันนั้นควรให้ผลของการจำแนกที่เหมือนกันและอยู่รวมเป็นกลุ่มเดียวกัน

#### 4.4.1 CNN model (hidden layer 7 layers)

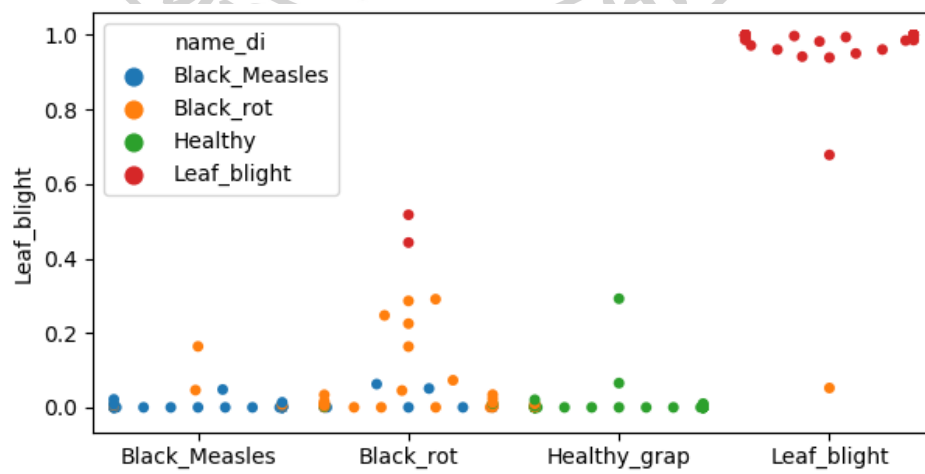
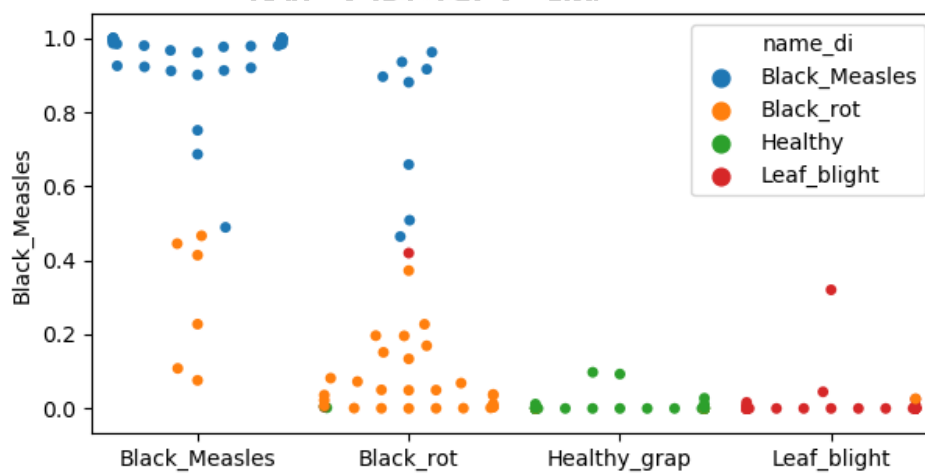
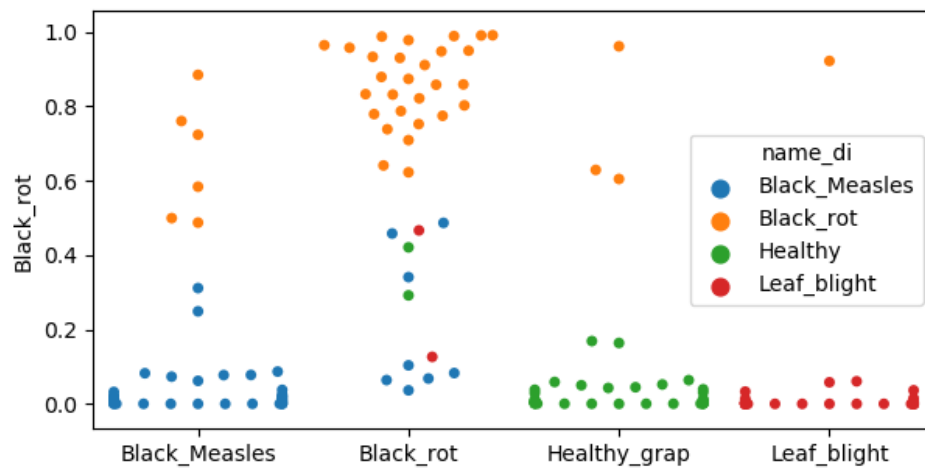
ในการทดสอบนี้เป็นการทดสอบวัดประสิทธิภาพในการจำแนกชนิดของโรคด้วยโมเดล CNN model มีชั้นของ Hidden layer จำนวน 7 ชั้น โดยจะใช้จำนวนรอบ (Epoch) ในการประมวลผล 50 รอบ หรือ 50 Epoch ผลลัพธ์ประสิทธิภาพของการจำแนกจะแสดงในรูปที่ 4.17 ถึงรูปที่ 4.32



รูปที่ 4.17 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original)

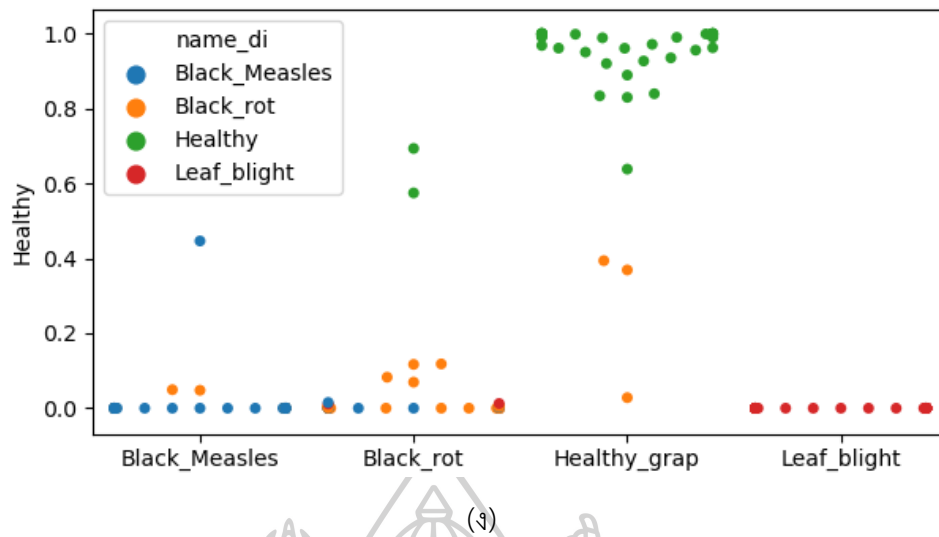
จากผลลัพธ์ในรูปที่ 4.17 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง





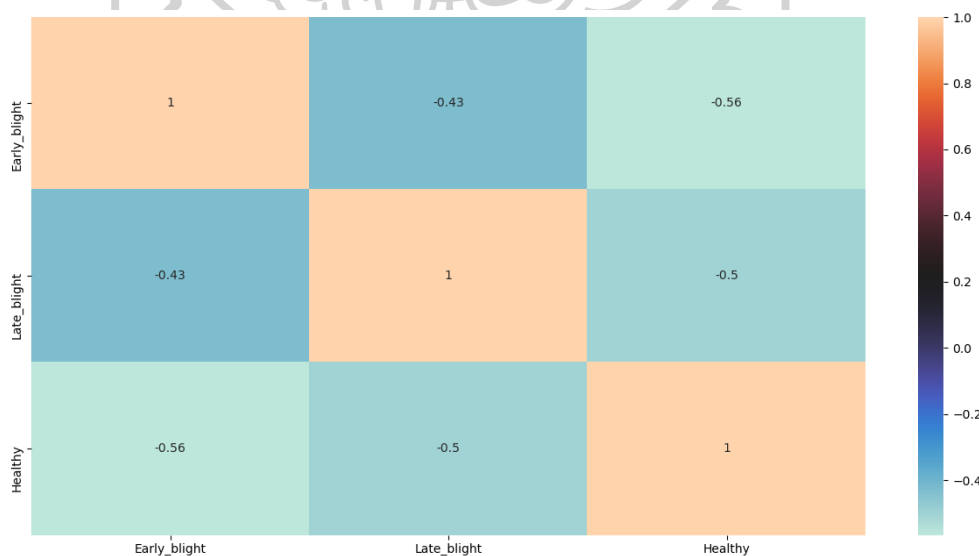
(a)





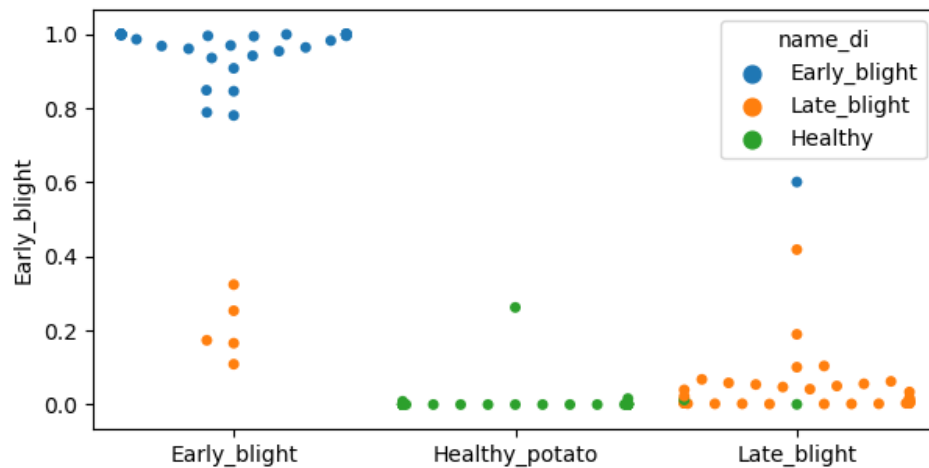
รูปที่ 4.18 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight (ง) Healthy

ในรูปที่ 4.18 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นค่าตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าค่าตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.18(ก) ผลของการจัดกลุ่มโรค Black rot แสดงเป็นสีส้มนั้นส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้

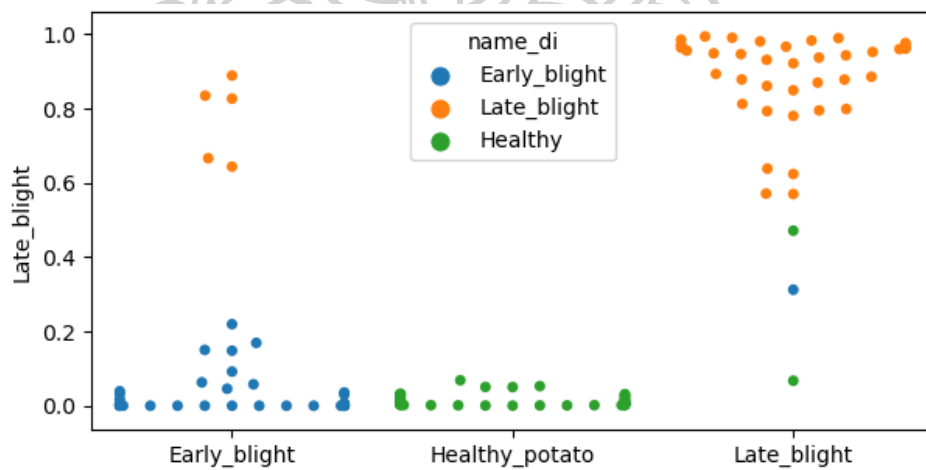


รูปที่ 4.19 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)

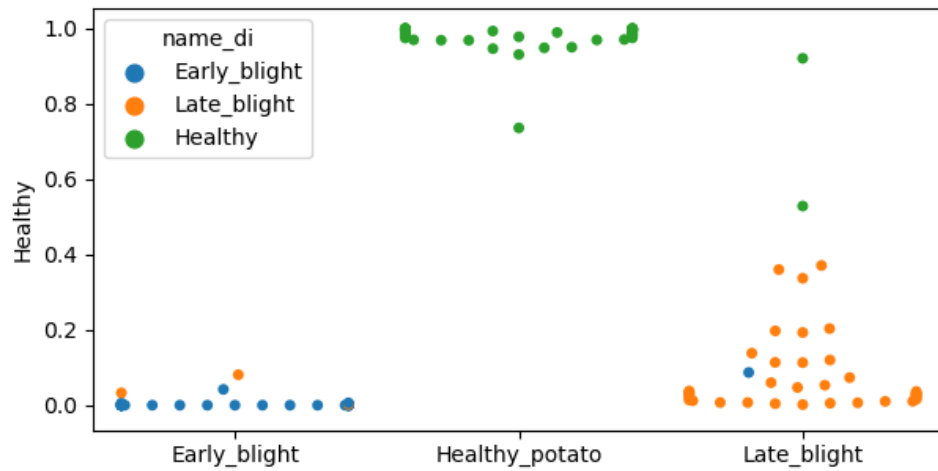
จากผลลัพธ์ในรูปที่ 4.19 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)



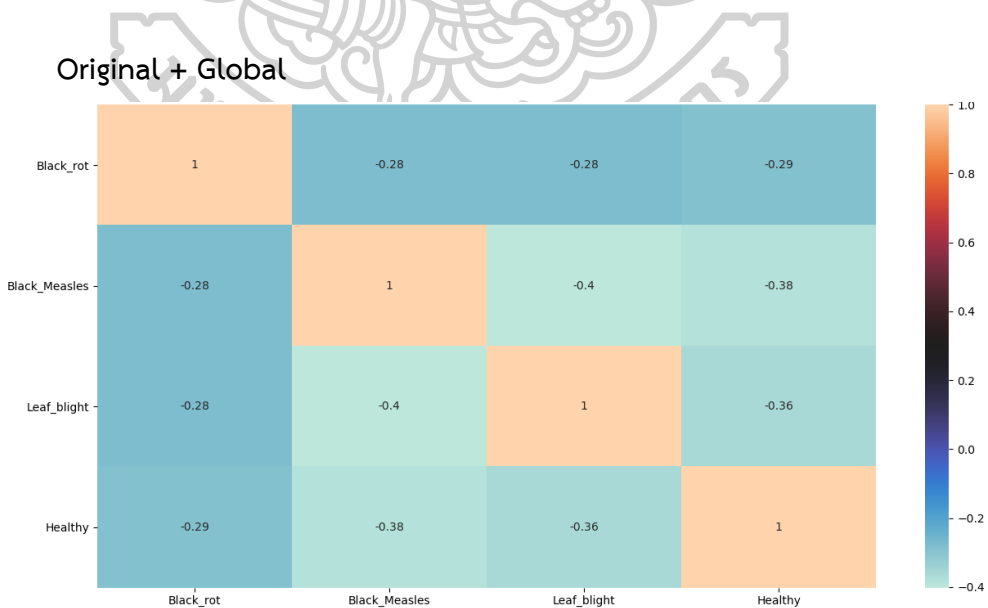
(ข)



(ค)

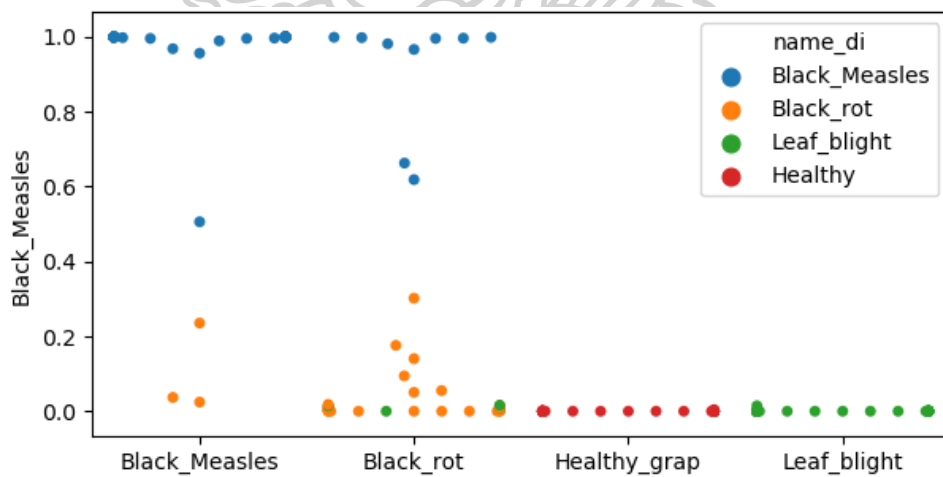
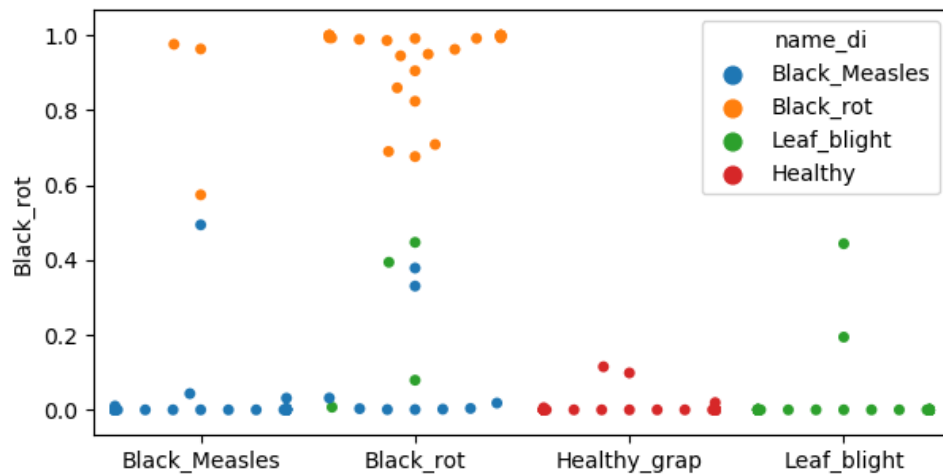
รูปที่ 4.20 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

ในรูปที่ 4.20 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.20(ข) ผลของการจัดกลุ่มโรค Late blight แสดงเป็นสีส้มนั้นส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้

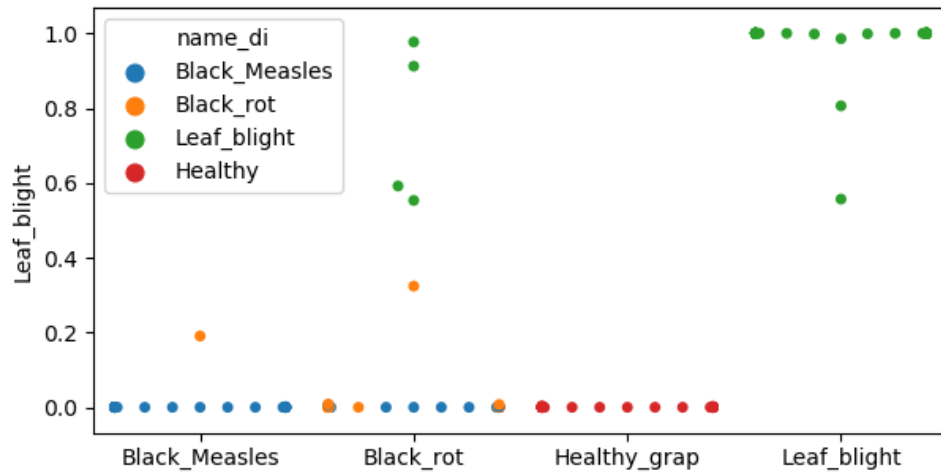


รูปที่ 4.21 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global)

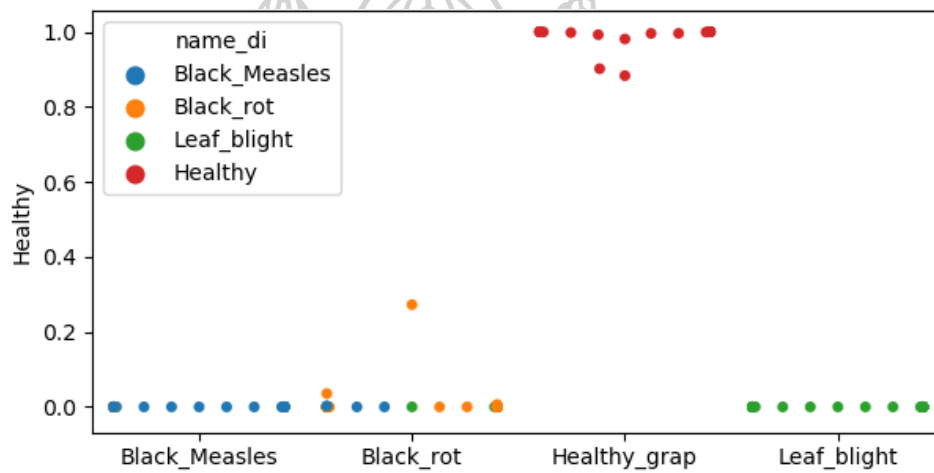
จากผลลัพธ์ในรูปที่ 4.21 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบของุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ข)



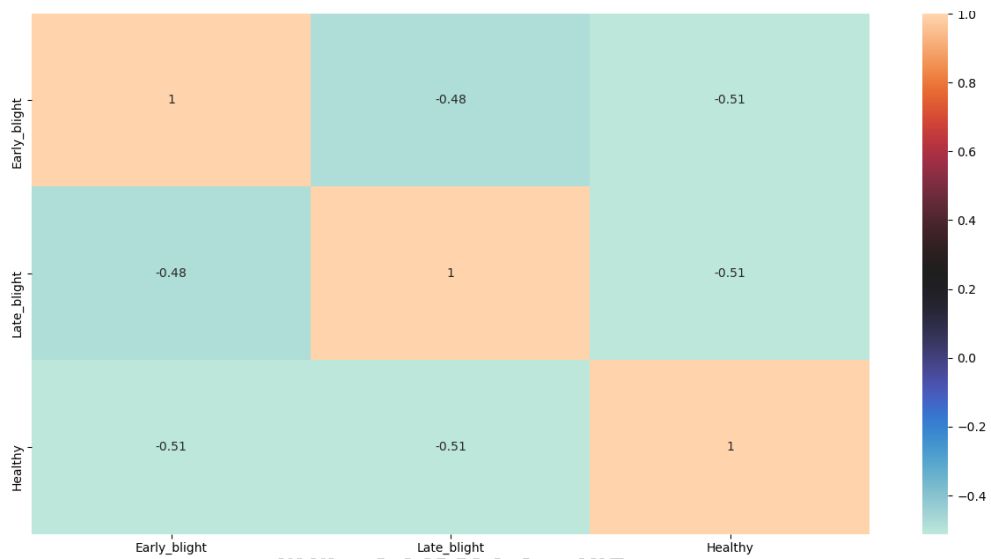
(ค)



(ง)

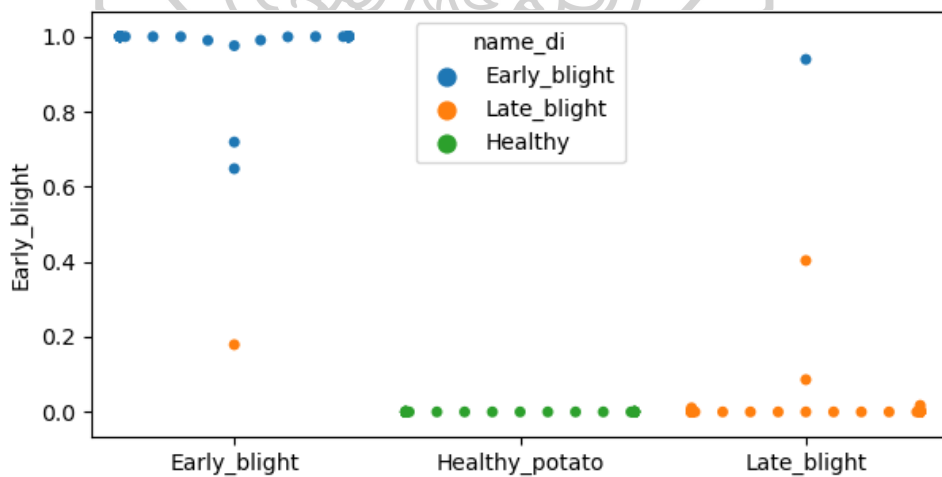
รูปที่ 4.22 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight (ง) Healthy

ในรูปที่ 4.22 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นค่าตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าค่าตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original+Global นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.22(ก) ผลของการจัดกลุ่มโรค Black rot แสดงเป็นสีส้มนั้นส่วนมากจะถูกจัดกลุ่มตรงกลับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้



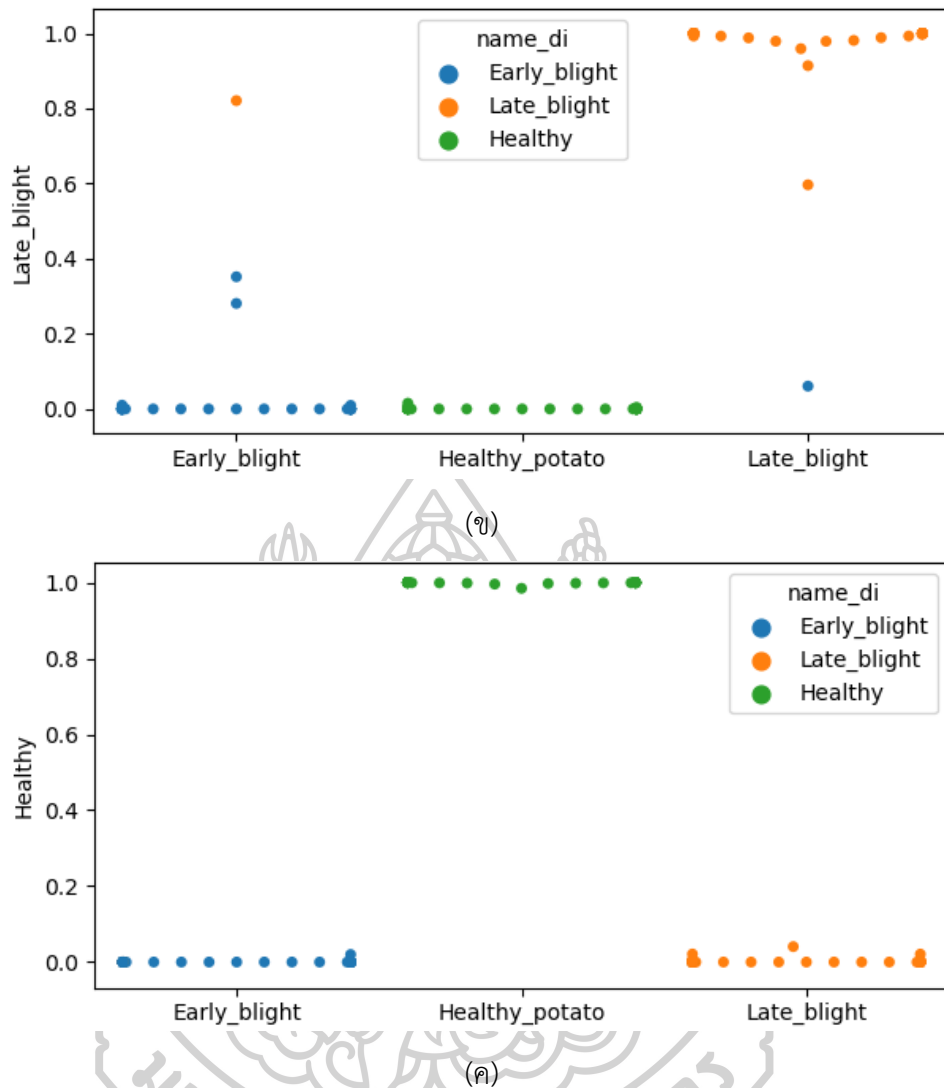
รูปที่ 4.23 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global)

จากผลลัพธ์ในรูปที่ 4.23 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)





รูปที่ 4.24 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

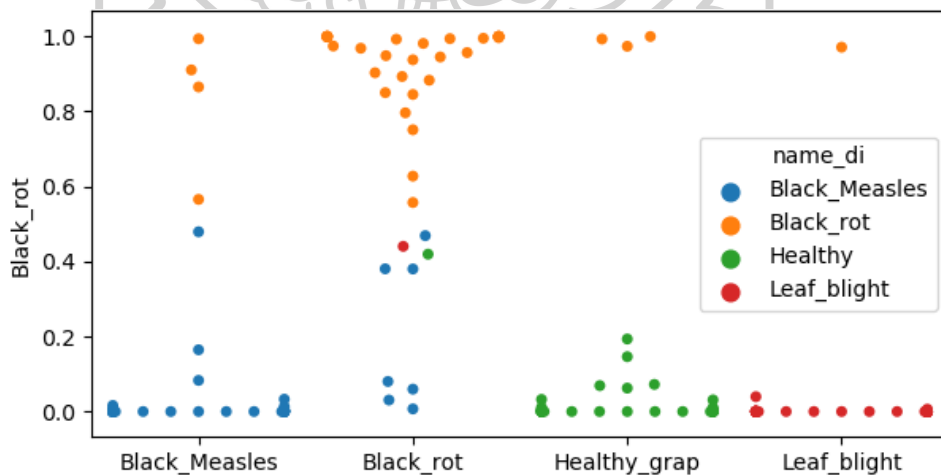
ในรูปที่ 4.24 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original+Global นั้นให้ผลที่ถูกต้องค่อนข้างสูงกว่าโมเดลอื่น ๆ แต่ก็ยังมีการทำนายผิดอยู่บางส่วนกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.24(ก) และรูปที่ 4.24(ข) ผลของการจัดกลุ่มโรคนั้นส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย

## Original + LoBIA

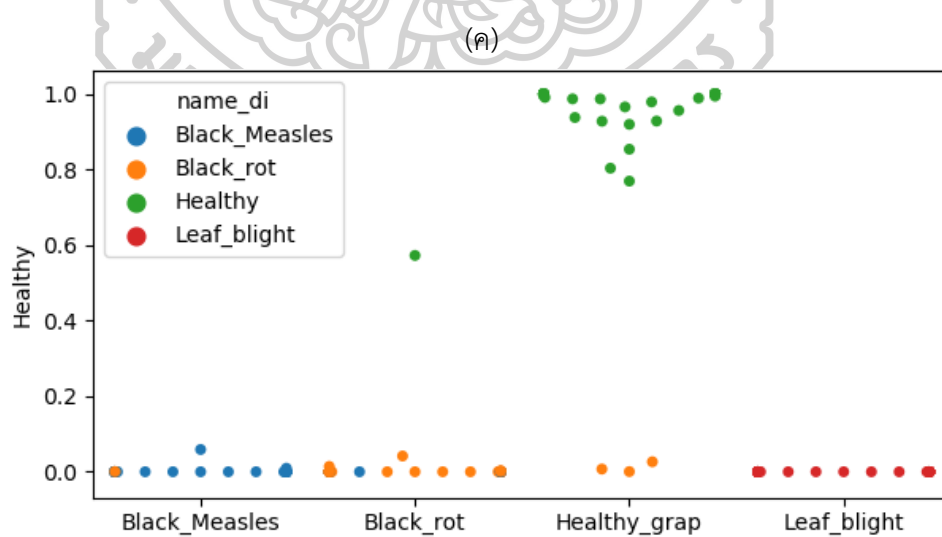
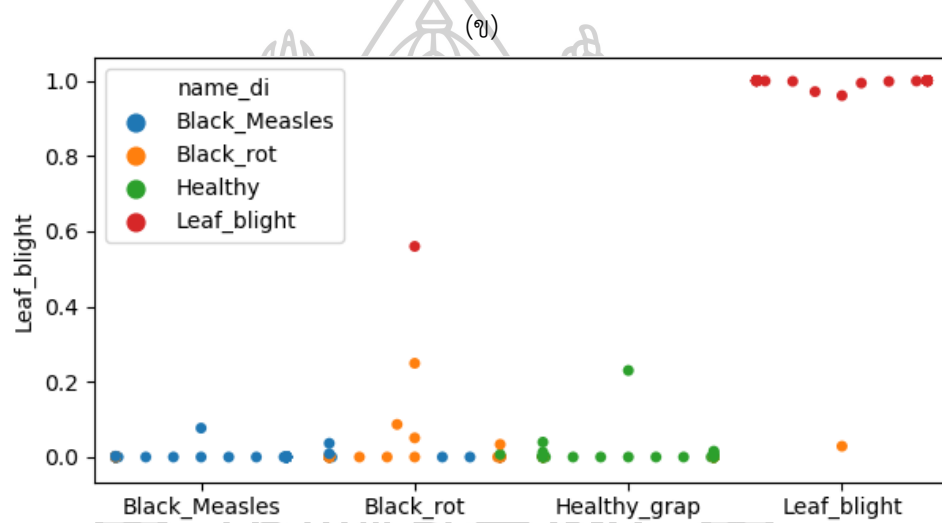
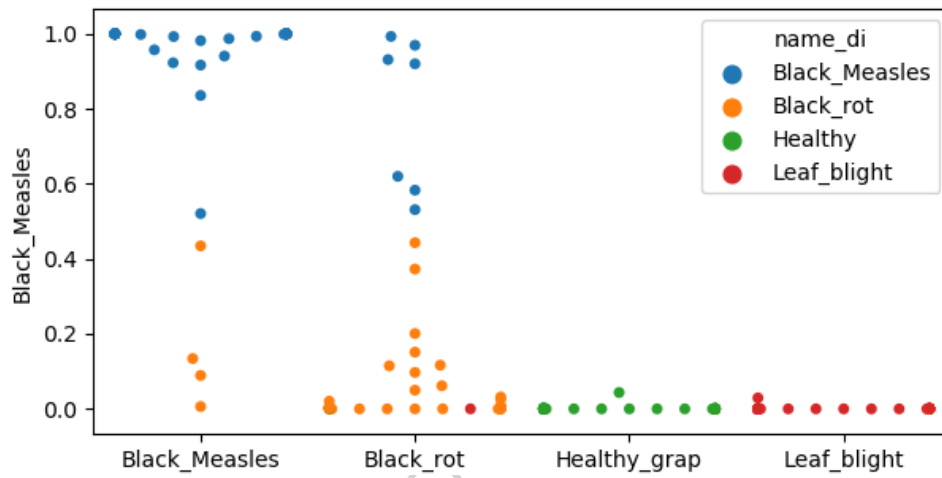


รูปที่ 4.25 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIA)

จากผลลัพธ์ในรูปที่ 4.25 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)

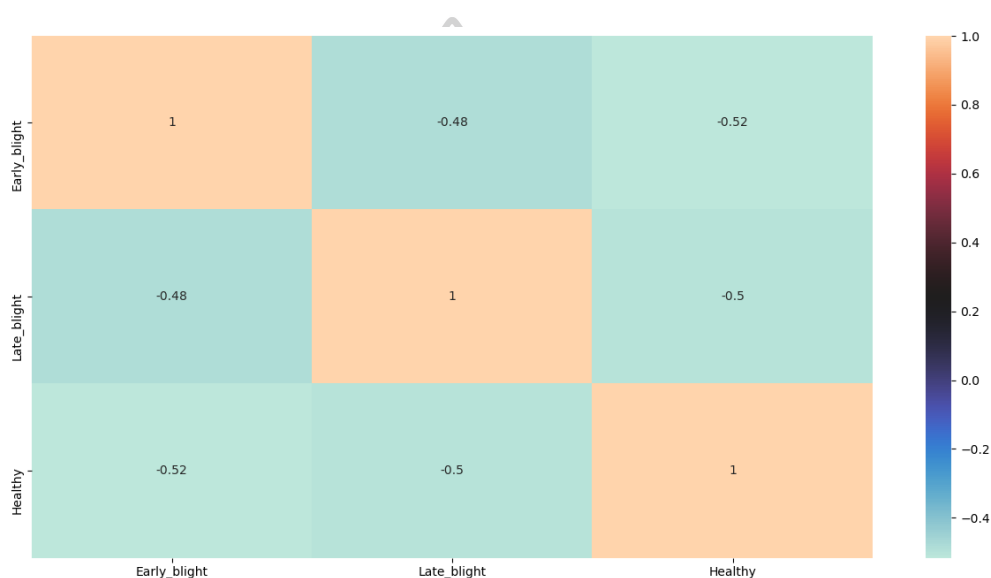


(ง)

รูปที่ 4.26 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight

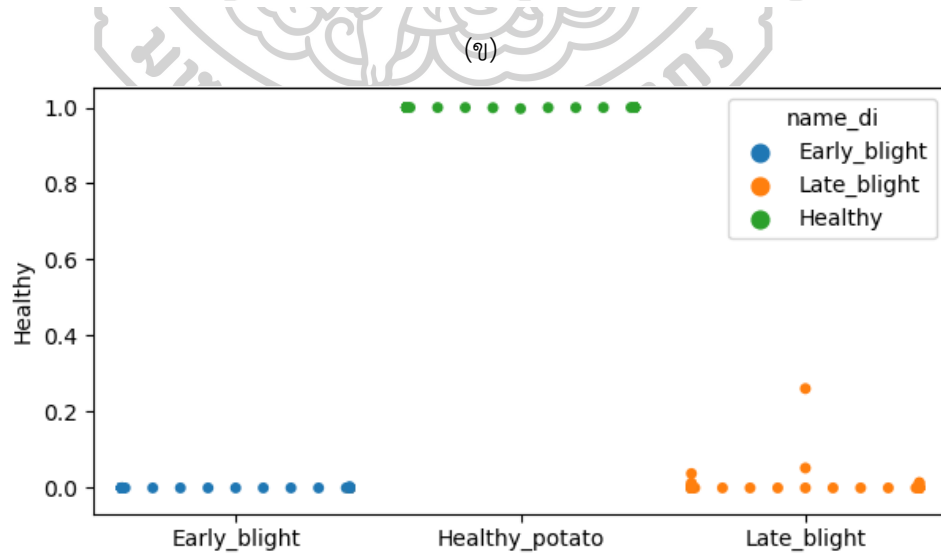
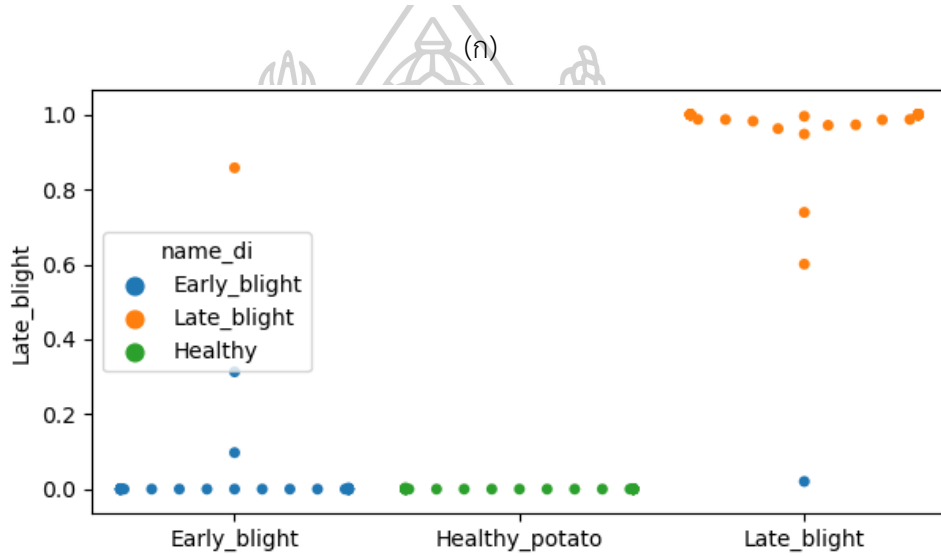
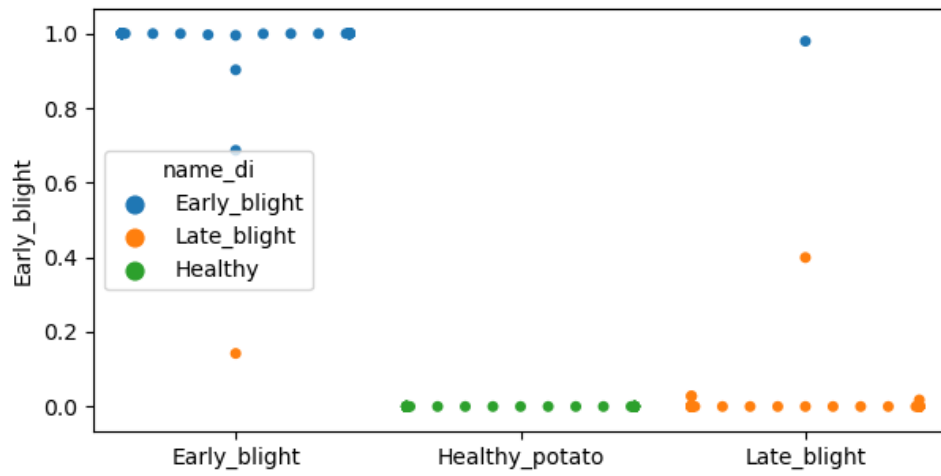
(ง) Healthy

จากรูปที่ 4.26 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคได้ค่อนข้างมีความถูกต้องสูง แต่อาจจะมีโรคบางชนิดที่อาจยังมีการทำนายผิดอยู่เล็กน้อย ดังตัวอย่างรูปที่ 4.26(ก) แสดงผลของการจัดกลุ่มโรค Black rot แสดงเป็นจุดสีส้ม เห็นได้ว่าผลของการทำนายสามารถที่จะทำนายชนิดของโรคได้ถูกต้องจำนวน 39 ภาพจาก 40 ภาพ



รูปที่ 4.27 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA)

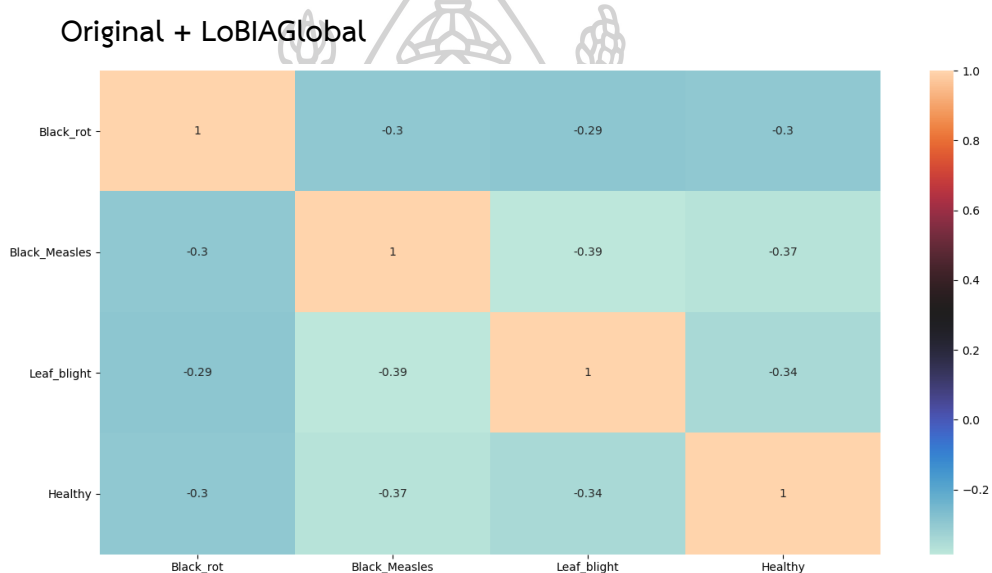
จากผลลัพธ์ในรูปที่ 4.27 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ค)

รูปที่ 4.28 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.28 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบมันฝรั่ง โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคได้ค่อนข้างมีความถูกต้องสูง แต่อาจจะมีโรคบางชนิดที่อาจยังมีการทำนายผิดอยู่เล็กน้อย โดยจากรูปจะเห็นได้ว่าผลของการจัดกลุ่มนั้นมีข้อมูลที่ผิดพลาดเพียงเล็กน้อย และอีกทั้งผลของการทำนายที่ถูกต่อนั้นกลุ่มข้อมูลยังอยู่กันเป็นกลุ่มไม่กระจายสูงต่ำมากนักแสดงถึงว่าข้อมูลที่ทำนายถูกนั้นทำนายถูกต้องในเปอร์เซ็นต์ที่สูงและใกล้เคียงกันเกือบทั้งหมด

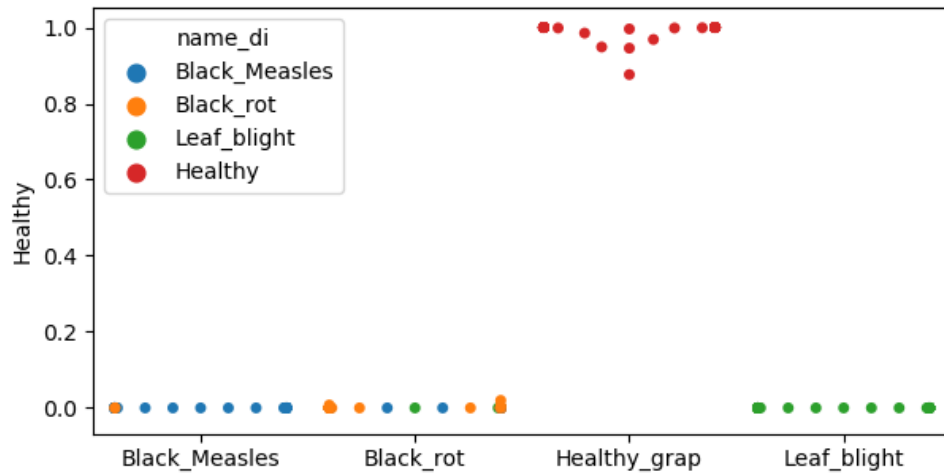


รูปที่ 4.29 Heatmap correlation matrix accuracy classification ภาพใบอ่อน (Original+LoBIAGlobal)

จากผลลัพธ์ในรูปที่ 4.29 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบอ่อนแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง

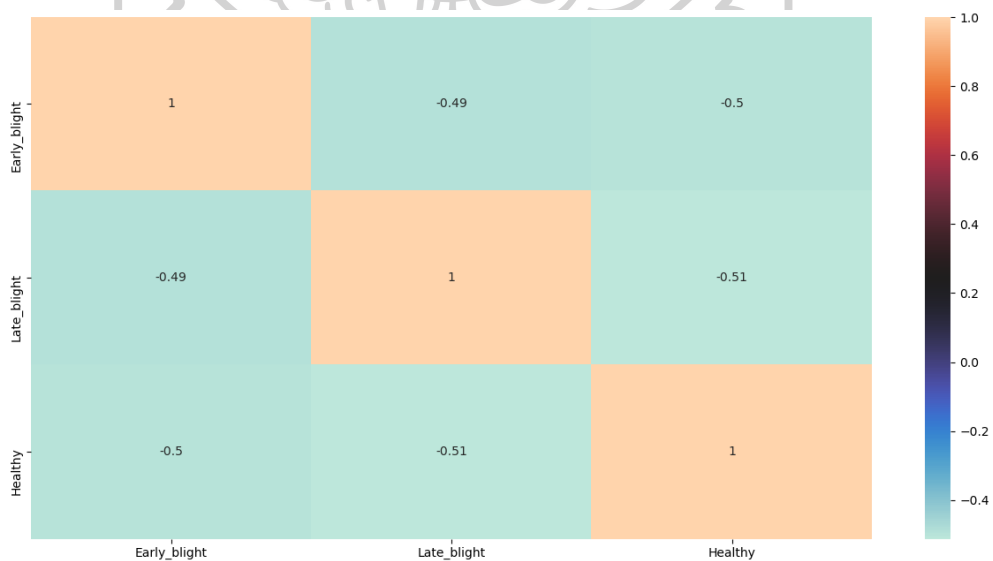






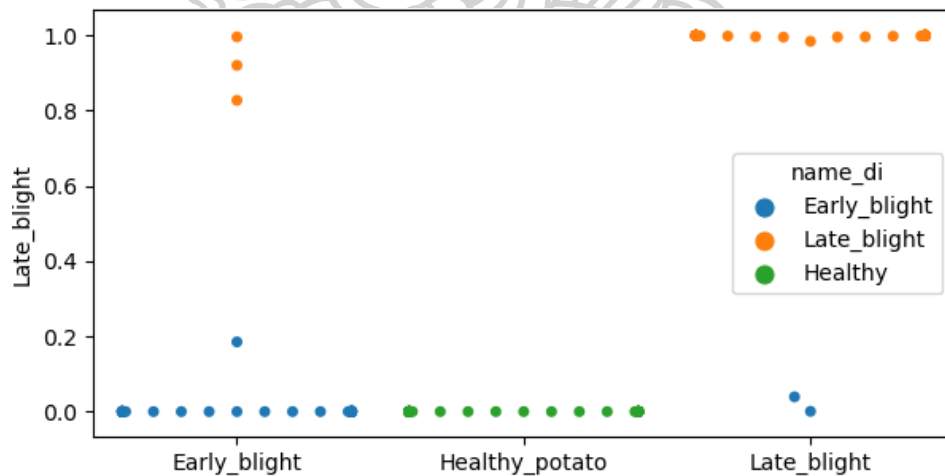
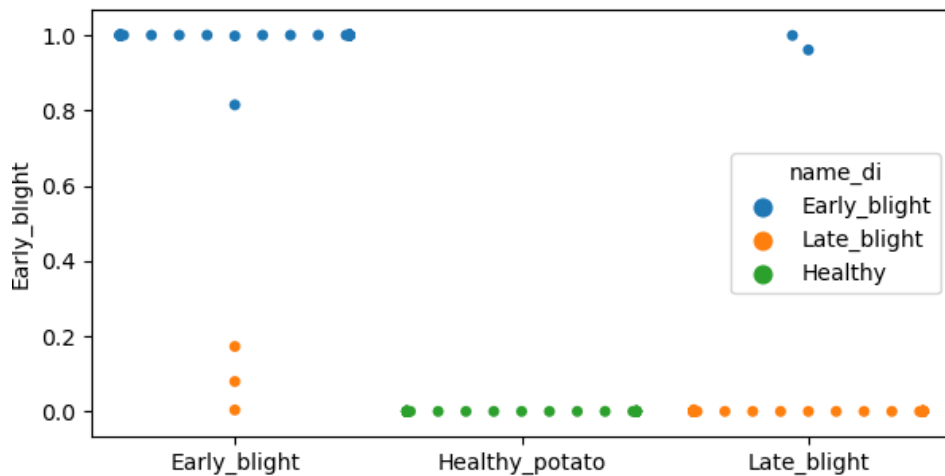
รูปที่ 4.30 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight (ง) Healthy

จากรูปที่ 4.30 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่การทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่ไม่เดลต้องการทำนายการจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่น่ามาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจจะเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้

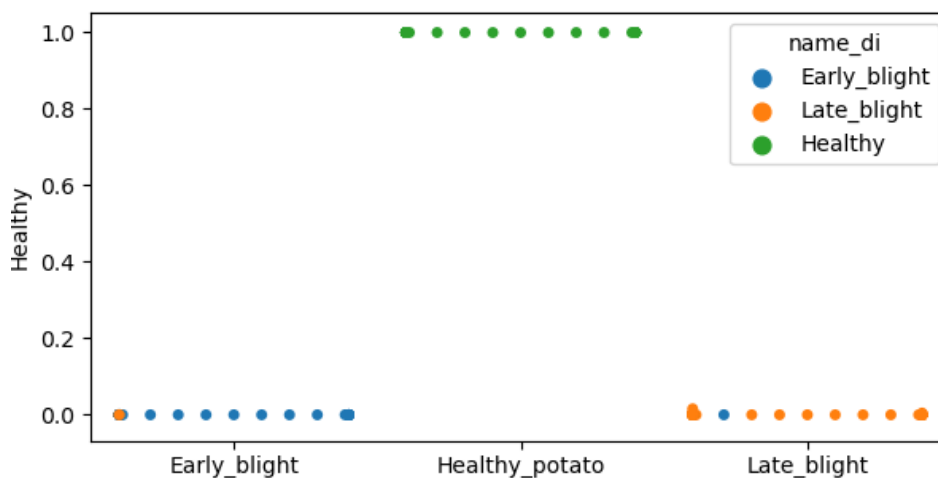


รูปที่ 4.31 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal)

จากผลลัพธ์ในรูปที่ 4.31 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ข)



(ค)

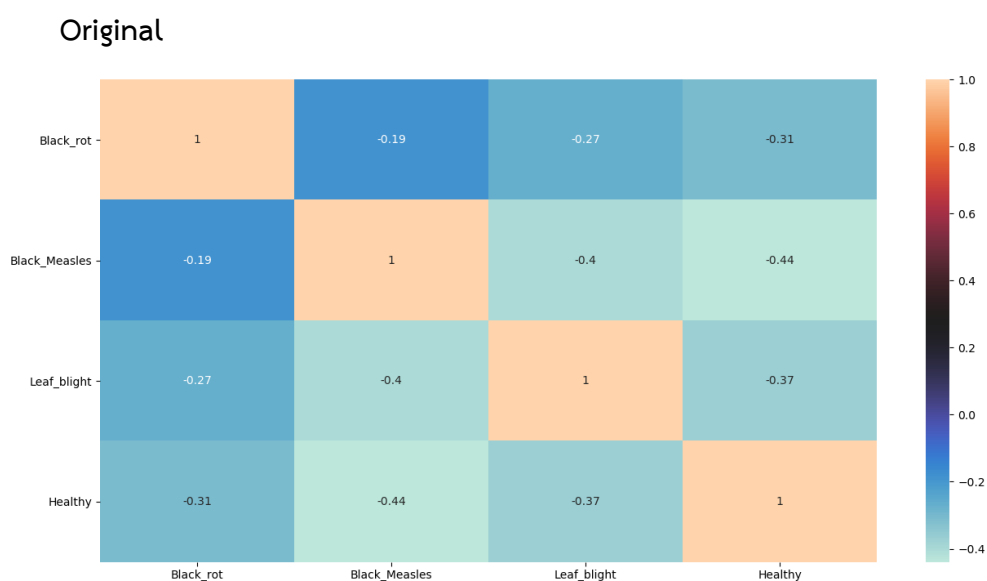
รูปที่ 4.32 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.32 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบมันฝรั่ง โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่นำมาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้

จากผลการทดสอบของแต่ละกลุ่มข้อมูลฝึกสอนที่ฝึกสอนกับโมเดล CNN model (hidden layer 7 layers) ในรูปที่ 4.17 ถึงรูปที่ 4.32 เห็นได้ว่าเมื่อทำการทดสอบภาพของกลุ่มโรคของใบองุ่นและใบมันฝรั่งในกลุ่มของ Original + Global กับ Original + LoBIA และ Original + LoBIA นั้นให้ผลลัพธ์ในการจัดกลุ่มได้ที่ใกล้เคียงกัน แต่จำนวนภาพที่นำมาฝึกสอนนั้นกลุ่มข้อมูลภาพ Original + LoBIA นั้นมีจำนวนที่น้อยกว่ากลุ่มทั้งสองค่อนข้างมาก และเมื่อนำผลการจัดกลุ่มที่ได้มาคำนวณหาค่าเฉลี่ยของความถูกต้องในการจัดกลุ่มแล้ว ได้ผลค่าเฉลี่ยแสดงตารางที่ 4.2 (ใบองุ่น) ตารางที่ 4.3 (ใบมันฝรั่ง) และตารางที่ 4.4 ตารางผลทดสอบของแต่ละชนิดของชุดข้อมูลฝึกสอน

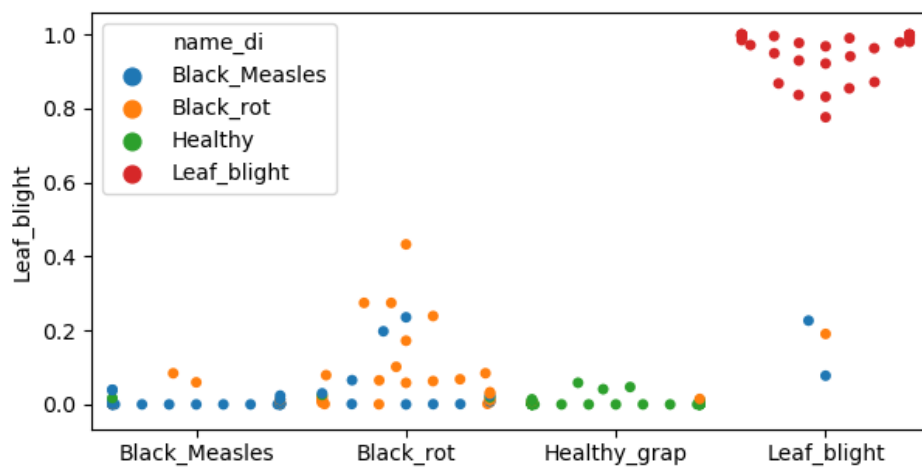
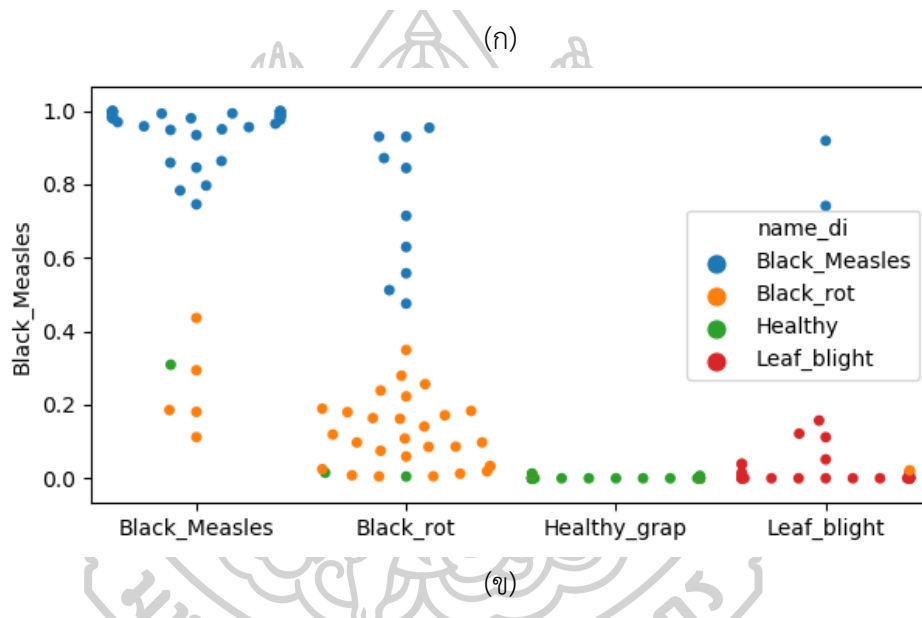
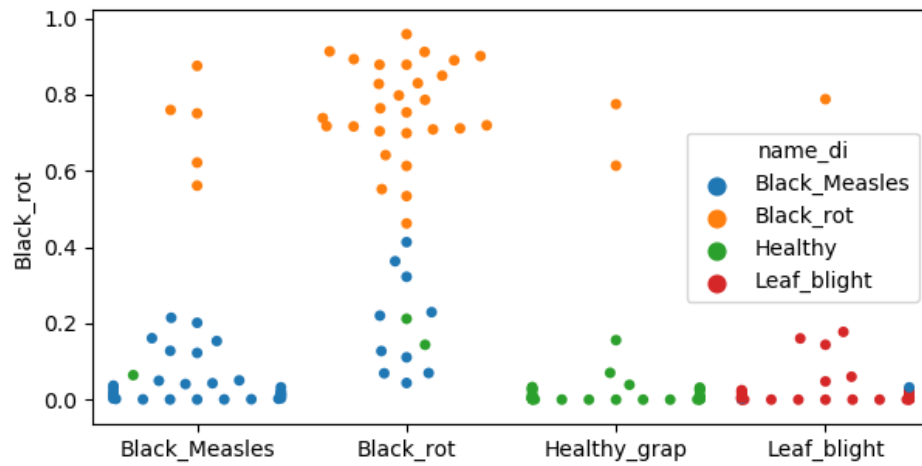
#### 4.4.2 CNN model (hidden layer 9 layers)

ในการทดสอบนี้เป็นการทดสอบวัดประสิทธิภาพในการจำแนกชนิดของโรคด้วยโมเดล CNN model มีชั้นของ Hidden layer จำนวน 9 ชั้น โดยจะใช้จำนวนรอบ (Epoch) ในการประมวลผล 50 รอบ หรือ 50 Epoch ผลลัพธ์ประสิทธิภาพของการจำแนกจะแสดงในรูปที่ 4.33 ถึงรูปที่ 4.48



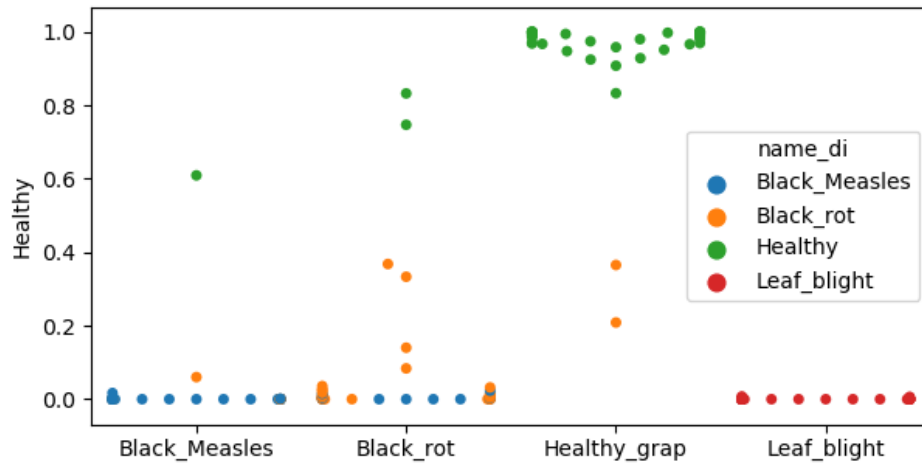
รูปที่ 4.33 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original)

จากผลลัพธ์ในรูปที่ 4.33 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(a)



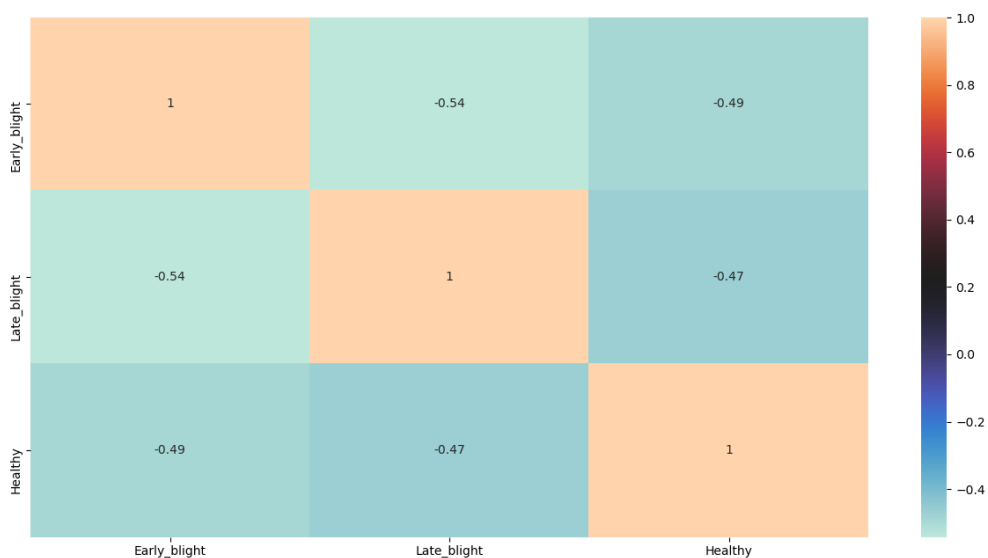


(ง)

รูปที่ 4.34 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight

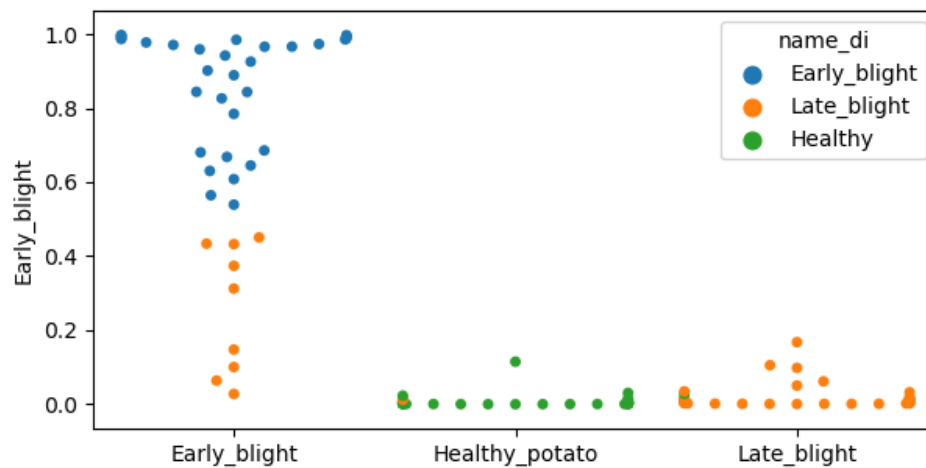
(ง) Healthy

ในรูปที่ 4.34 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นค่าตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าค่าตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.34(ข) ผลของการจัดกลุ่มโรค Black measles แสดงเป็นสีน้ำเงินนั้นส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้

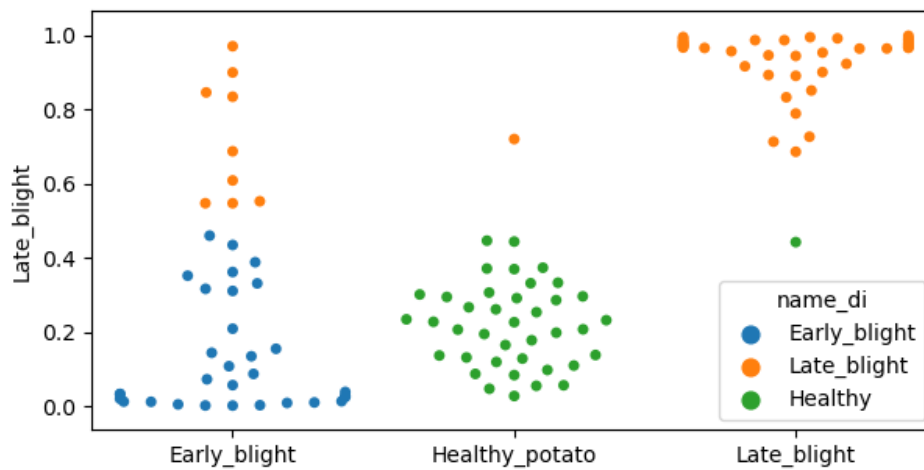


รูปที่ 4.35 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)

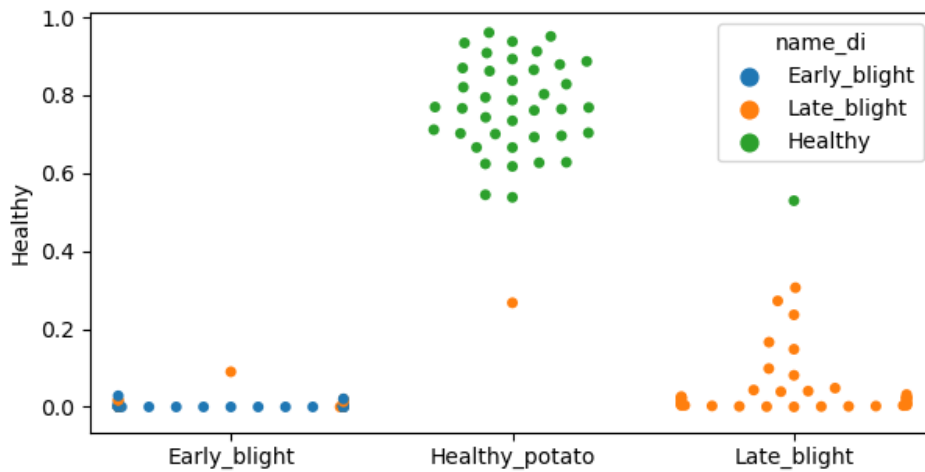
จากผลลัพธ์ในรูปที่ 4.35 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบของต้นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)



(ข)



(ค)

รูปที่ 4.36 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

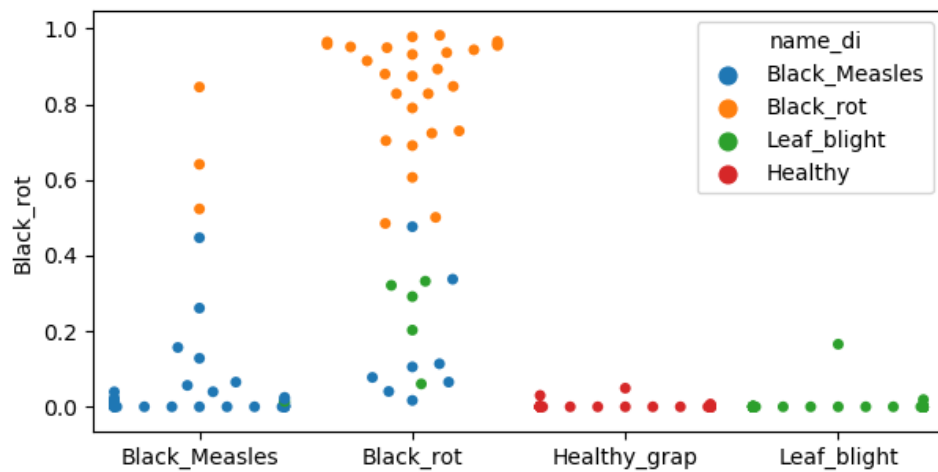
ในรูปที่ 4.36 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.36(ข) ผลของการจัดกลุ่มโรค Late blight แสดงเป็นสีส้มนั้นส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้

Original + Global

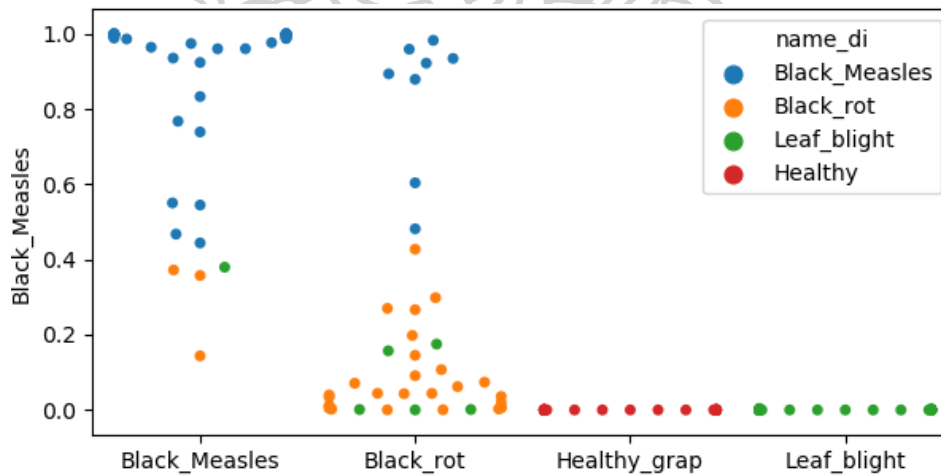


รูปที่ 4.37 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global)

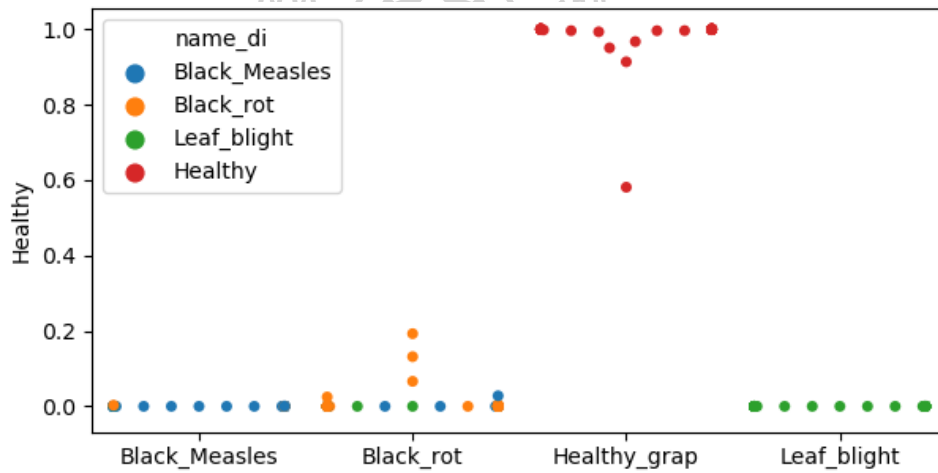
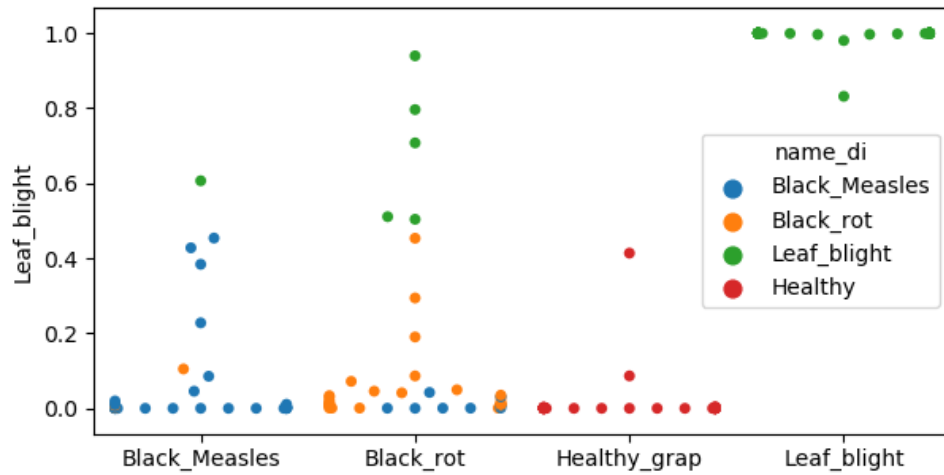
จากผลลัพธ์ในรูปที่ 4.37 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบของุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)

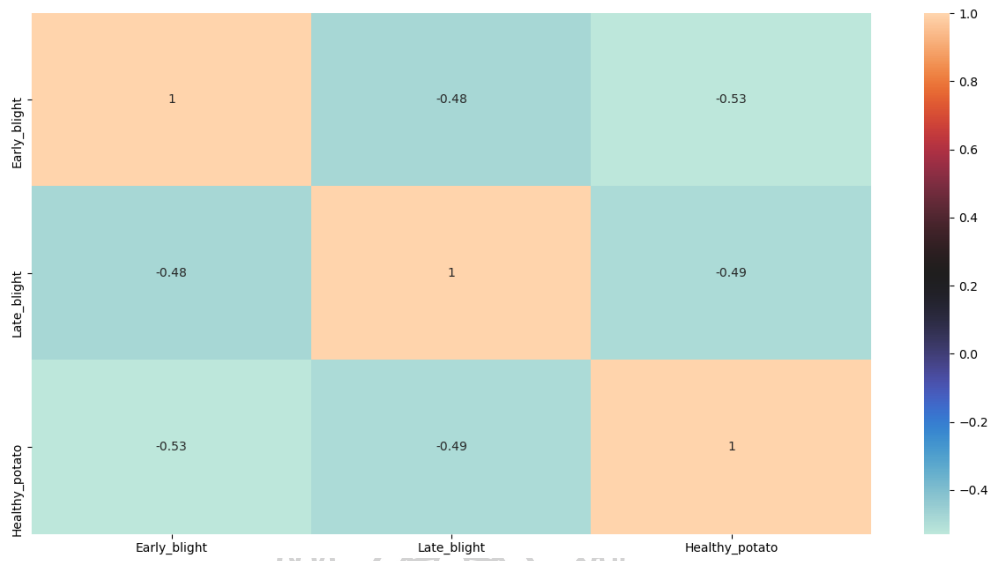


(ข)



รูปที่ 4.38 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight  
(ง) Healthy

ในรูปที่ 4.38 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นค่าตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าค่าตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original+Global นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จาก รูปที่ 4.38(ก) ผลของการจัดกลุ่มโรค Black rot แสดงเป็นสีส้มนั้นส่วนมากจะถูกจัดกลุ่มตรงกลับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้

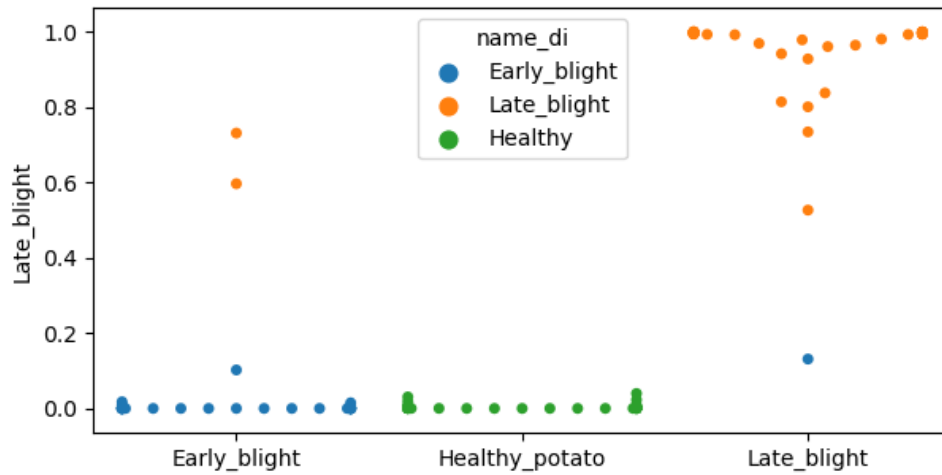


รูปที่ 4.39 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global)

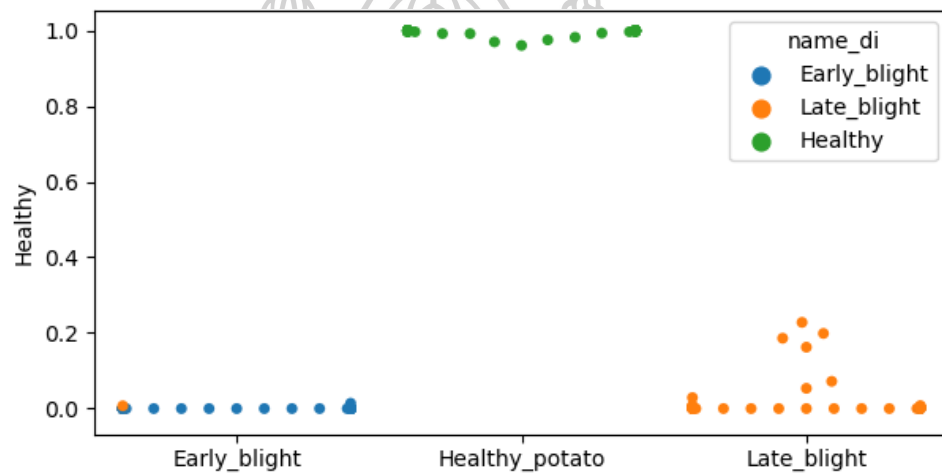
จากผลลัพธ์ในรูปที่ 4.39 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)



(ข)



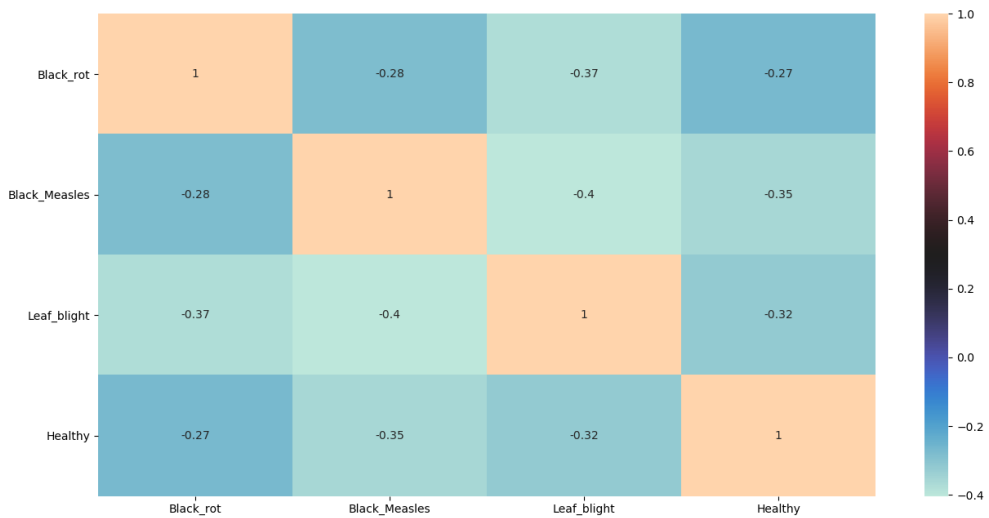
(ค)

รูปที่ 4.40 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

ในรูปที่ 4.40 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original+Global นั้นให้ผลที่ถูกต้องค่อนข้างสูงกว่าโมเดลอื่น ๆ แต่ก็ยังมีการทำนายผิดอยู่บางส่วนกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.40(ก) และรูปที่ 4.40(ข) ผลของการจัดกลุ่มโรคนั้นส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย

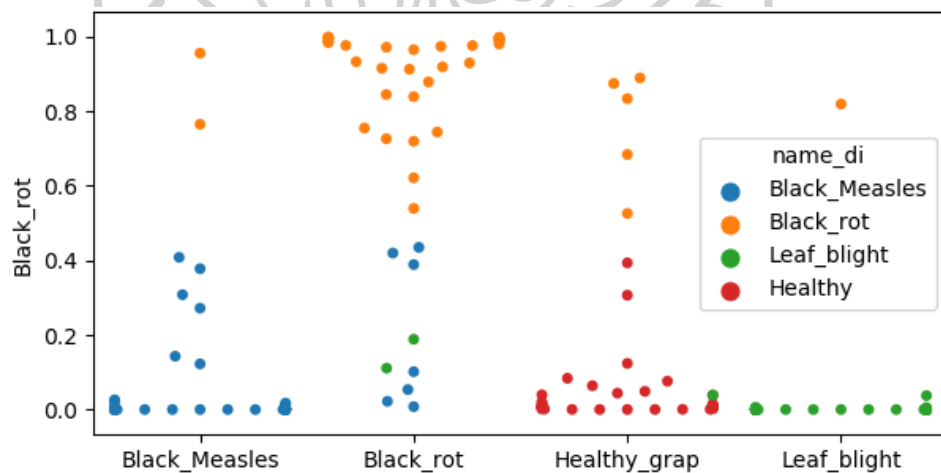


## Original + LoBIA

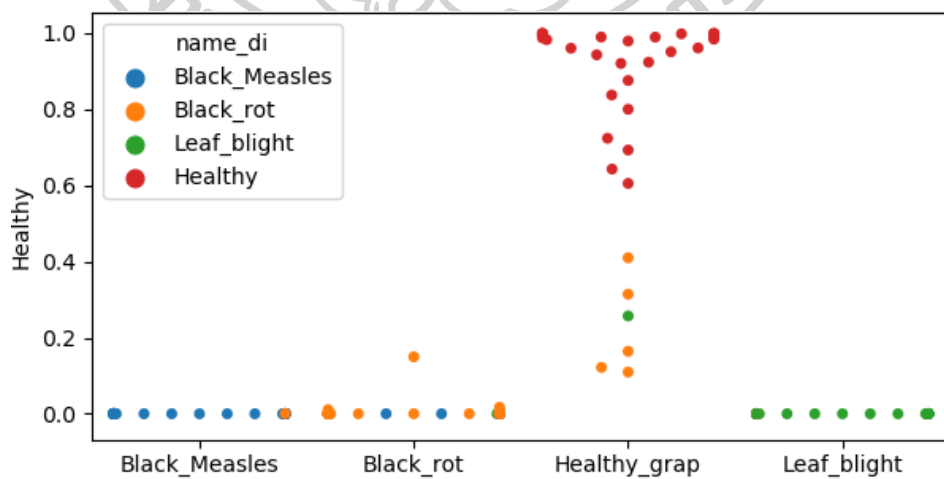
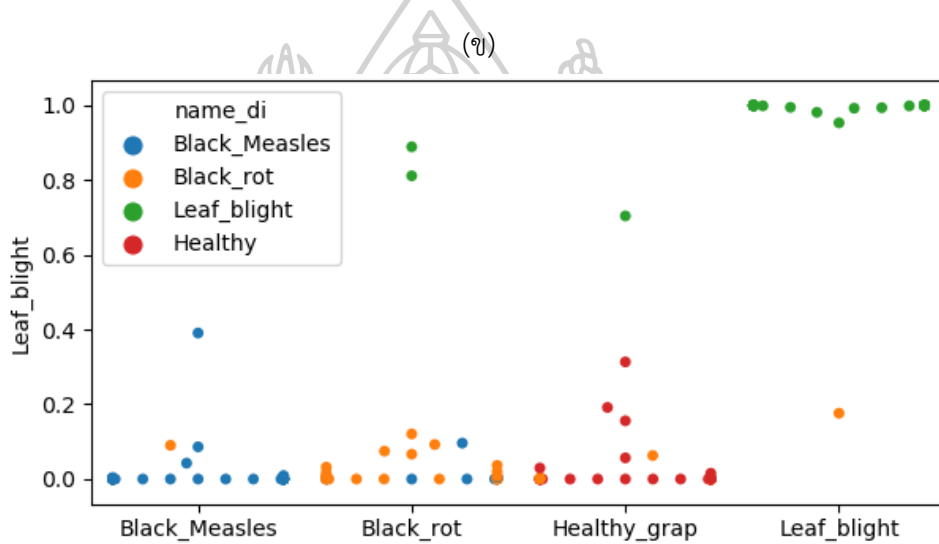
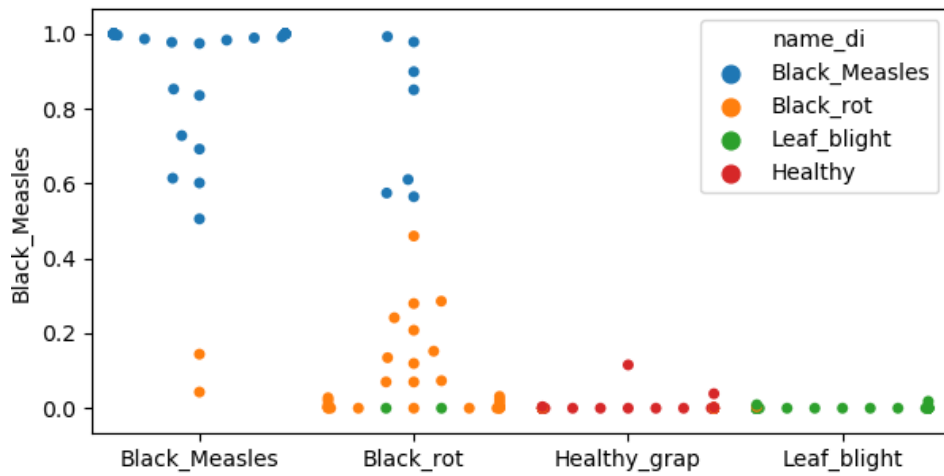


รูปที่ 4.41 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIA)

จากผลลัพธ์ในรูปที่ 4.41 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)

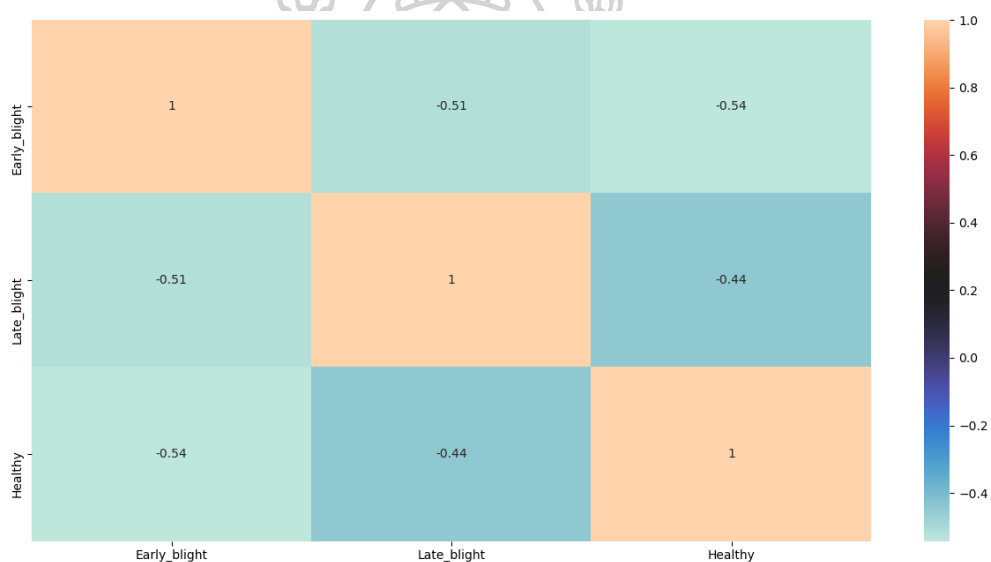


(ง)

รูปที่ 4.42 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight

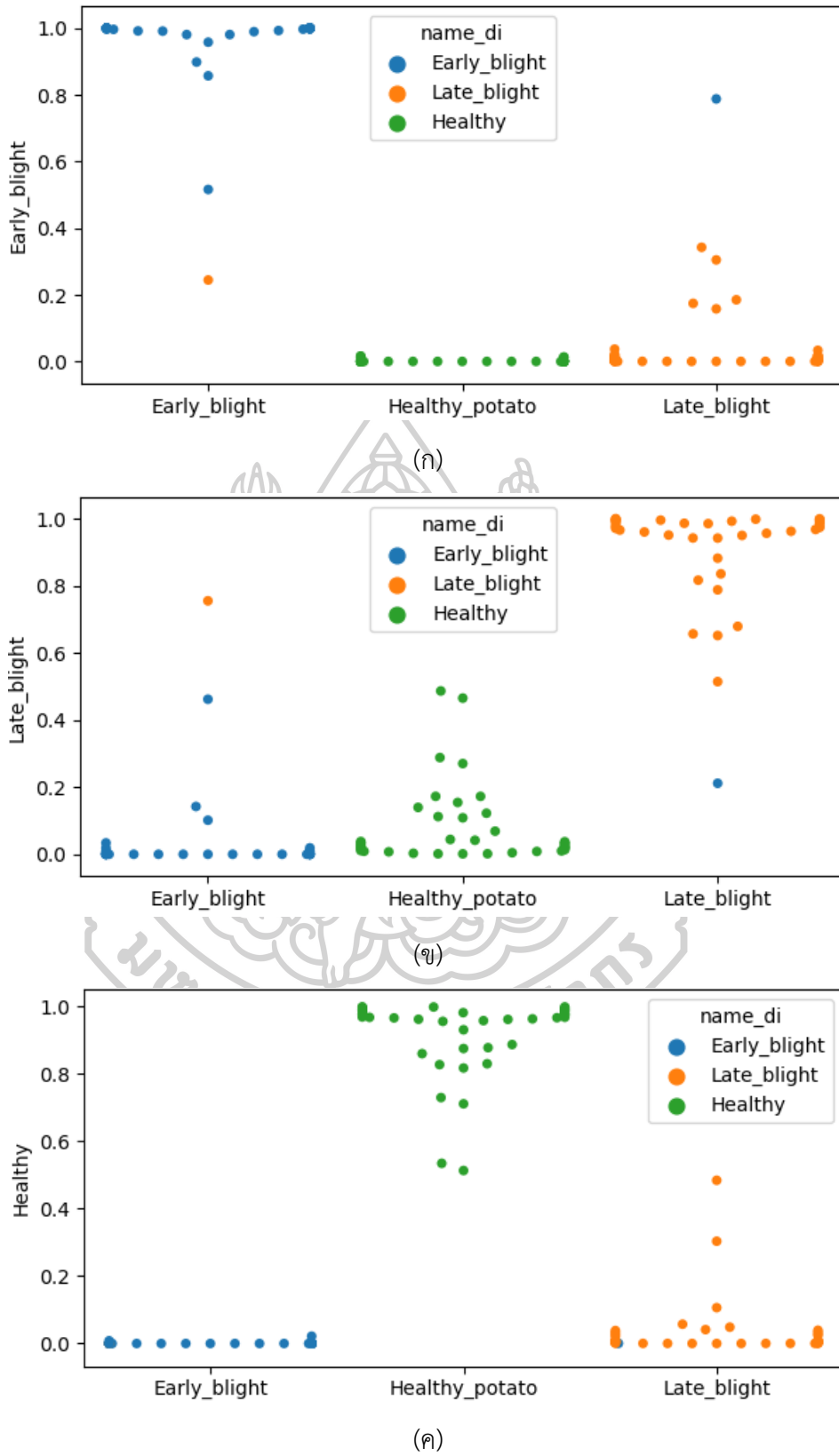
(ง) Healthy

จากรูปที่ 4.42 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคได้ค่อนข้างมีความถูกต้องสูง แต่อาจจะมีโรคบางชนิดที่อาจยังมีการทำนายผิดอยู่เล็กน้อย ดังตัวอย่างรูปที่ 4.42(ก) แสดงผลของการจัดกลุ่มโรค Black rot แสดงเป็นจุดสีส้ม เห็นได้ว่าผลของการทำนายสามารถที่จะทำนายชนิดของโรคได้ถูกต้องจำนวน 39 ภาพจาก 40 ภาพ และรูปที่ 4.42(ง) แสดงผลของการจัดกลุ่มโรค Healthy แสดงจุดสีแดง เห็นได้ว่าข้อมูลส่วนใหญ่ที่ทำการทำนายผลได้อย่างถูกต้องและใกล้เคียงกัน แต่อาจจะมีภาพบางส่วนที่ทำนายผิด



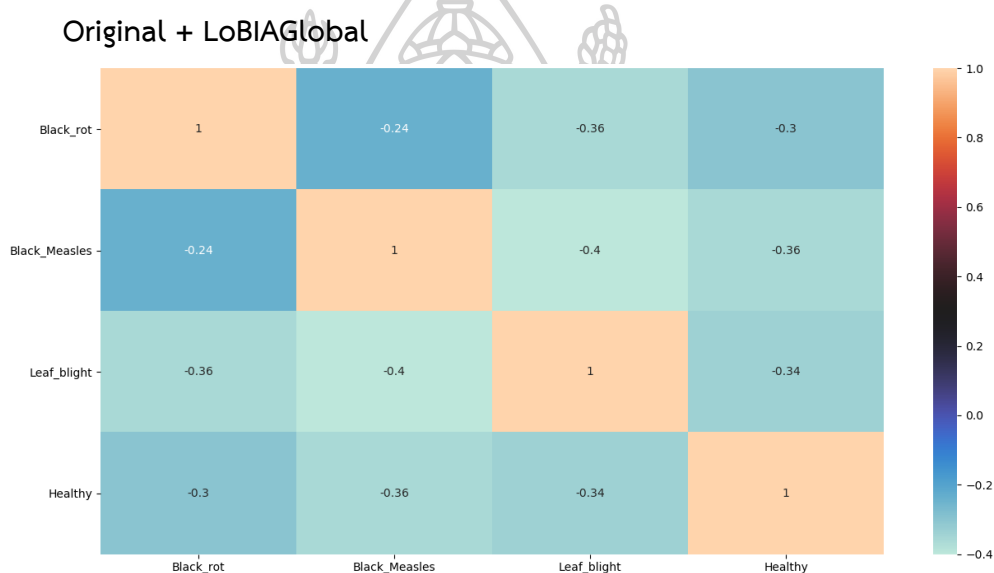
รูปที่ 4.43 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA)

จากผลลัพธ์ในรูปที่ 4.43 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



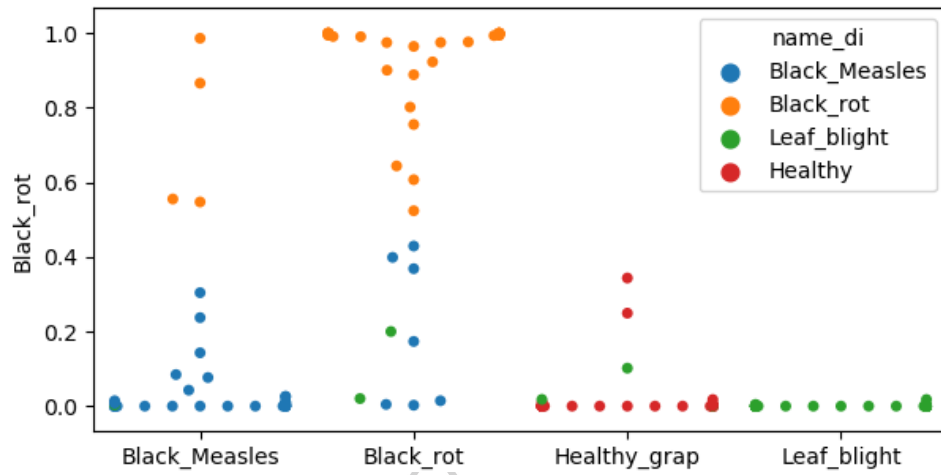
รูปที่ 4.44 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.44 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบมันฝรั่ง โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคได้ค่อนข้างมีความถูกต้องสูง แต่อาจจะมีโรคบางชนิดที่อาจยังมีการทำนายผิดอยู่เล็กน้อย โดยจากรูปจะเห็นได้ว่าผลของการจัดกลุ่มนั้นมีข้อมูลที่ผิดพลาดเพียงเล็กน้อย และอีกทั้งผลของการทำนายที่ถูกต่อนั้นกลุ่มข้อมูลยังอยู่กันเป็นกลุ่มไม่กระจายสูงต่ำมากนักแสดงถึงว่าข้อมูลที่ทำนายถูกนั้นทำนายถูกต้องในเปอร์เซ็นต์ที่สูงและใกล้เคียงกันเกือบทั้งหมด

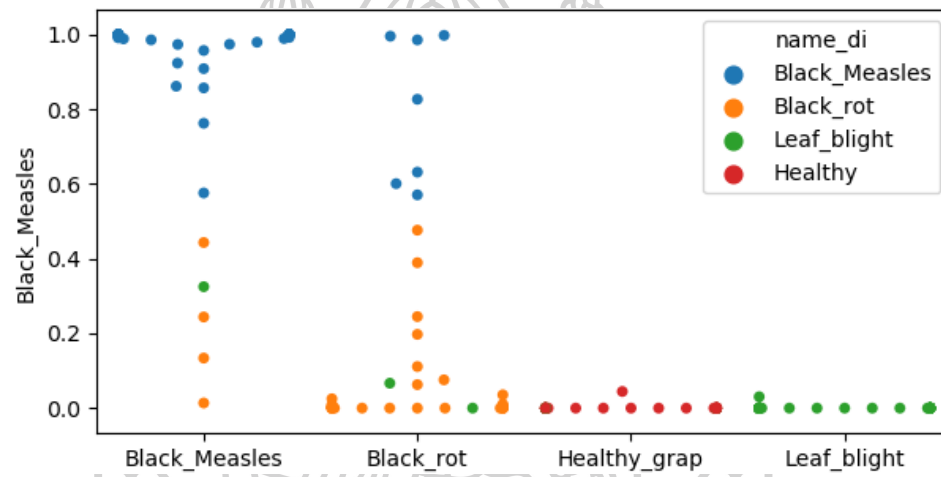


รูปที่ 4.45 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIAGlobal)

จากผลลัพธ์ในรูปที่ 4.45 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



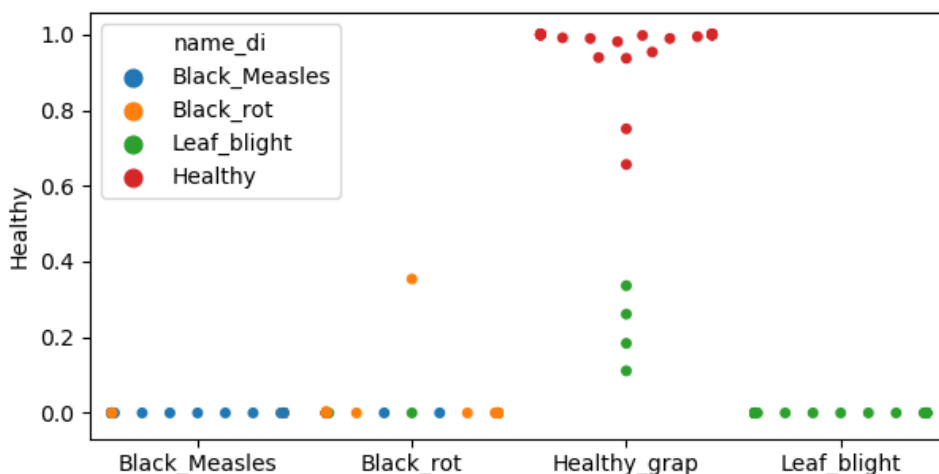
(a)



(b)

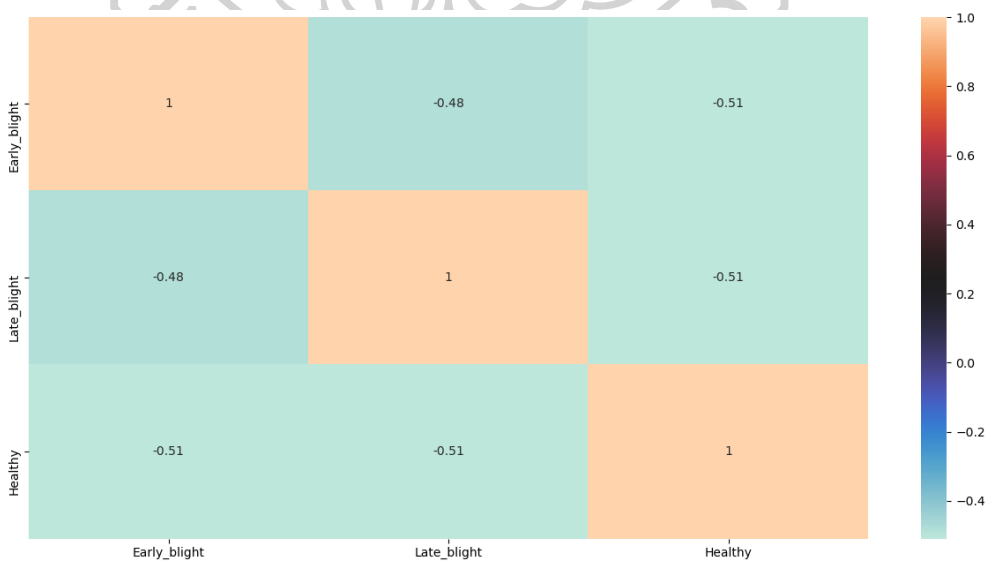


(c)



รูปที่ 4.46 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight (ง) Healthy

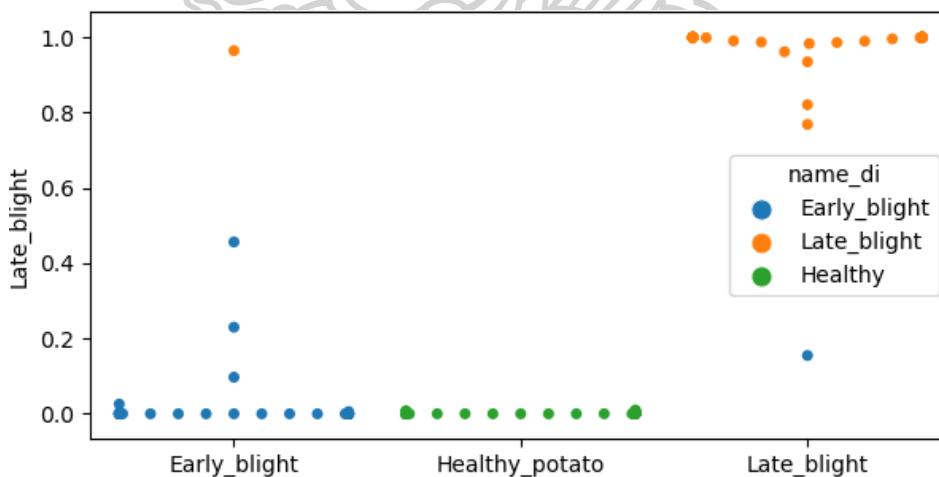
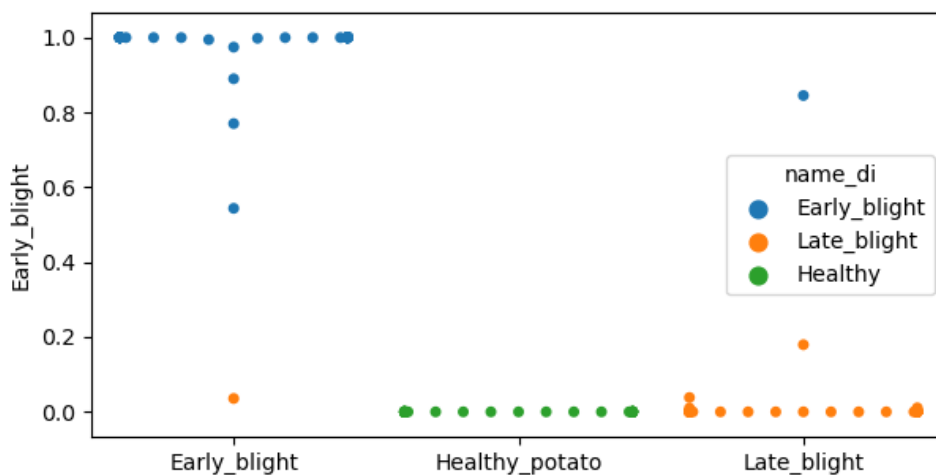
จากรูปที่ 4.46 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำการทำนายและแกนนอนบ่งบอกถึงกลุ่มค่าตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่นำมาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจจะเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้



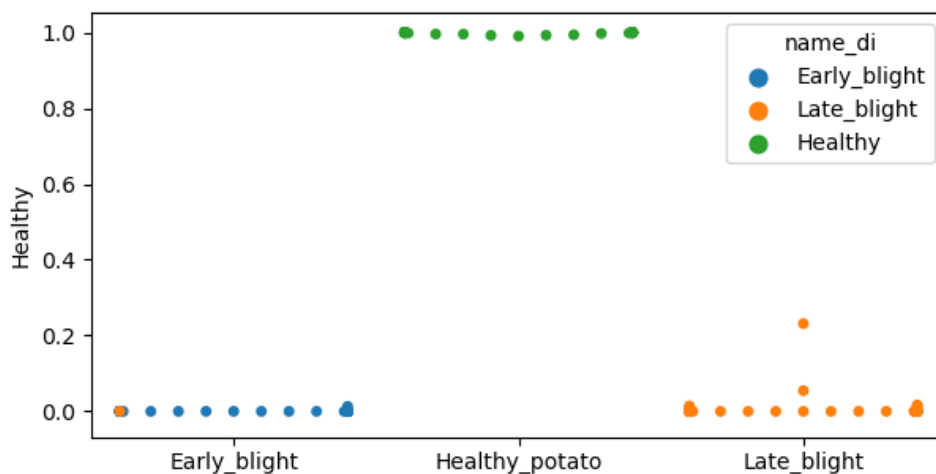
รูปที่ 4.47 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal)



จากผลลัพธ์ในรูปที่ 4.47 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ข)



(ง)

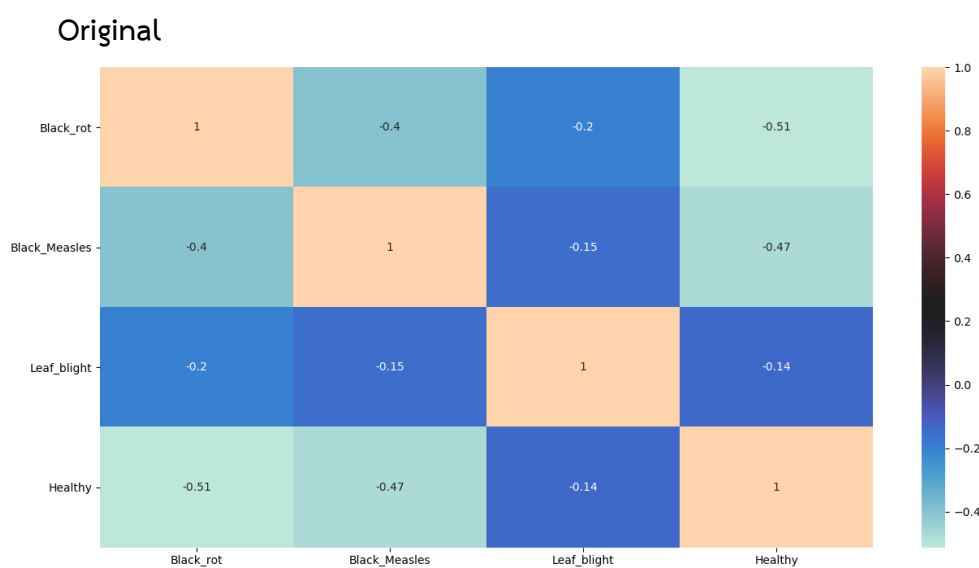
รูปที่ 4.48 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.48 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบมันฝรั่ง โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำการทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่นำมาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจจะเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้

จากผลการทดสอบของแต่ละกลุ่มข้อมูลฝึกสอนที่ฝึกสอนกับโมเดล CNN model (hidden layer 9 layers) ในรูปที่ 4.33 ถึงรูปที่ 4.48 เห็นได้ว่าเมื่อทำการทดสอบภาพของกลุ่มโรคของใบองุ่นและใบมันฝรั่งในกลุ่มของ Original + Global กับ Original + LoBIA และ Original + LoBIA นั้นให้ผลลัพธ์ในการจัดกลุ่มได้ใกล้เคียงกัน แต่จำนวนภาพที่นำมาฝึกสอนนั้นกลุ่มข้อมูลภาพ Original + LoBIA นั้นมีจำนวนที่น้อยกว่ากลุ่มทั้งสองค่อนข้างมาก และเมื่อนำผลการจัดกลุ่มที่ได้มาคำนวณหาค่าเฉลี่ยของความถูกต้องในการจัดกลุ่มแล้ว ได้ผลค่าเฉลี่ยแสดงตารางที่ 4.2 (ใบองุ่น) ตารางที่ 4.3 (ใบมันฝรั่ง) และตารางที่ 4.4 ตารางผลทดสอบของแต่ละชนิดของชุดข้อมูลฝึกสอน

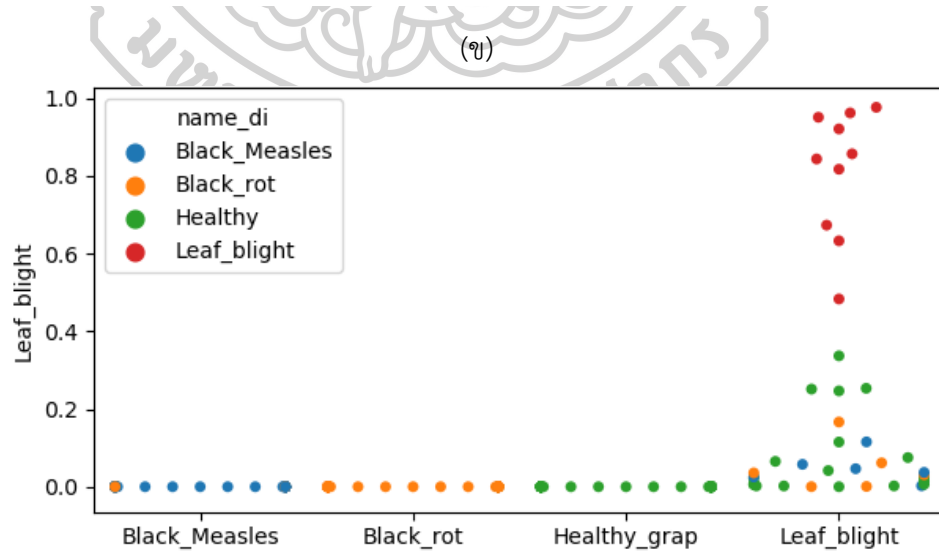
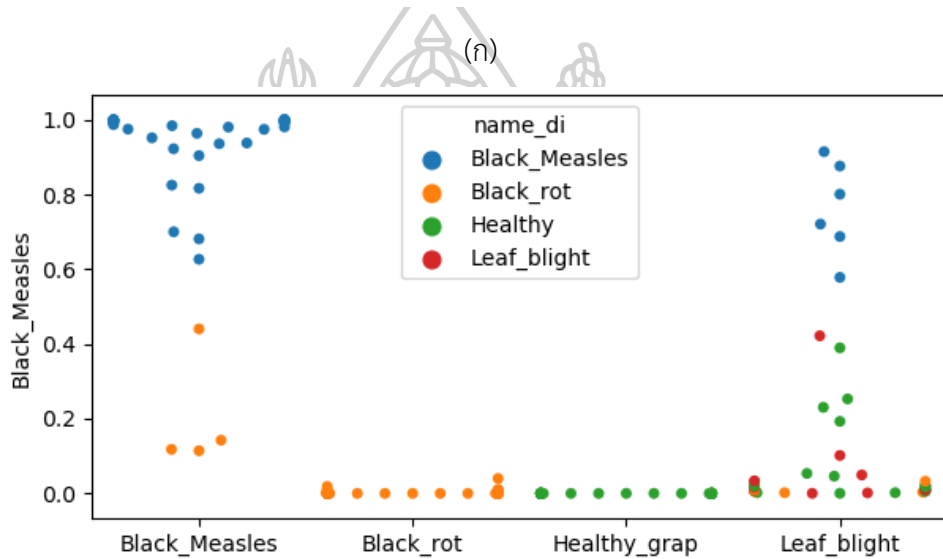
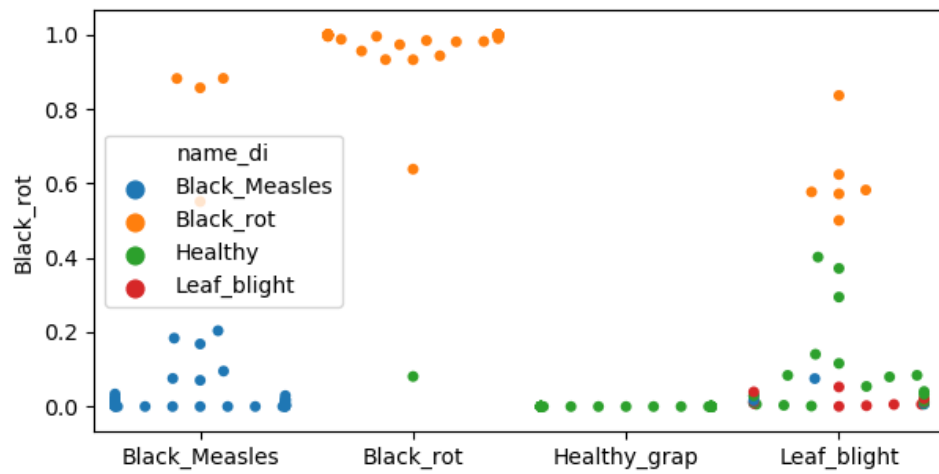
#### 4.4.3 MobileNet

ในการทดสอบนี้เป็นการทดสอบวัดประสิทธิภาพในการจำแนกชนิดของโรคด้วยโมเดล MobileNet model มีชั้นของ Hidden layer จำนวน 88 ชั้น ใช้จำนวนรอบ (Epoch) ในการประมวลผล 50 รอบ หรือ 50 Epoch ผลลัพธ์ประสิทธิภาพของการจำแนกจะแสดงในรูปที่ 4.49 ถึงรูปที่ 4.64

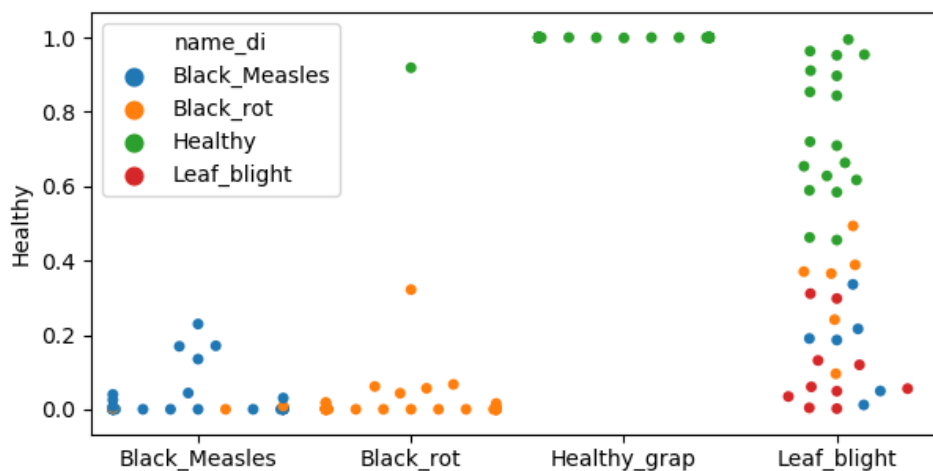


รูปที่ 4.49 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original)

จากผลลัพธ์ในรูปที่ 4.49 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกต้องทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง

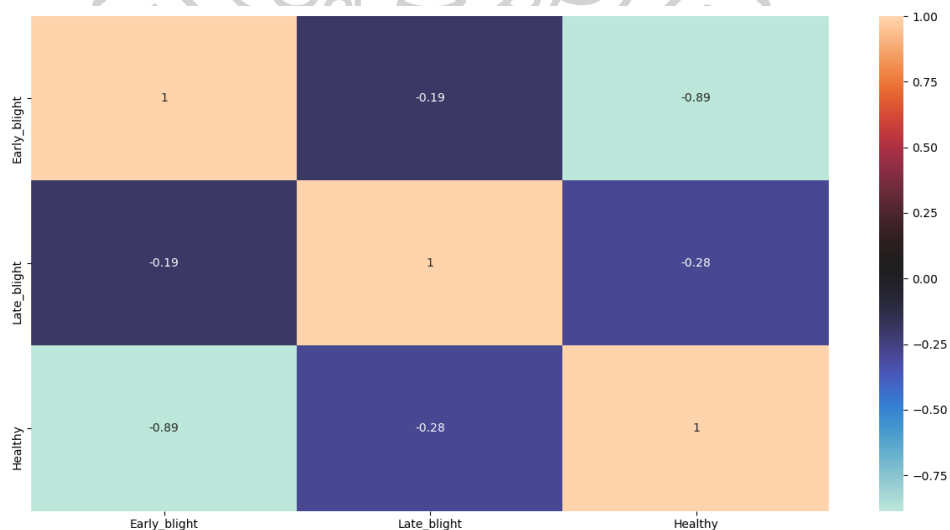


(a)



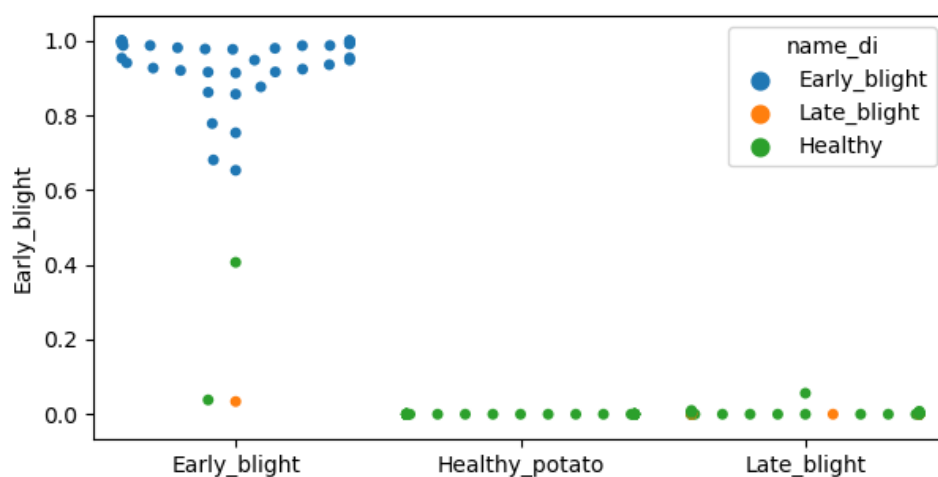
(ง)  
รูปที่ 4.50 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight (ง) Healthy

ในรูปที่ 4.50 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นค่าตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าค่าตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.50(ค) ผลของการจัดกลุ่มโรค Leaf blight แสดงเป็นสีแดงนั้นค่าตอบที่ถูกจัดกลุ่มตรงกับผลเฉลยมีค่อนข้างน้อย โดยภาพส่วนใหญ่ได้ทำนายเป็นโรคชนิดอื่น ๆ ค่อนข้างมาก

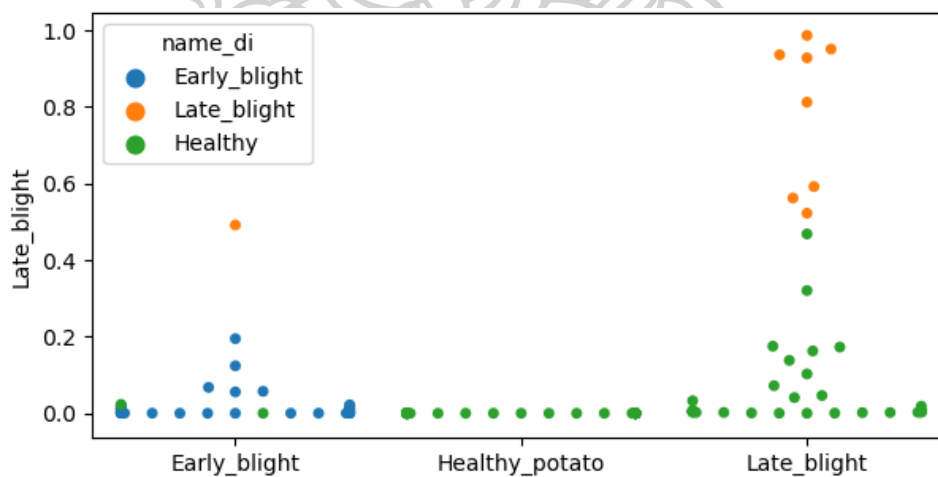


รูปที่ 4.51 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)

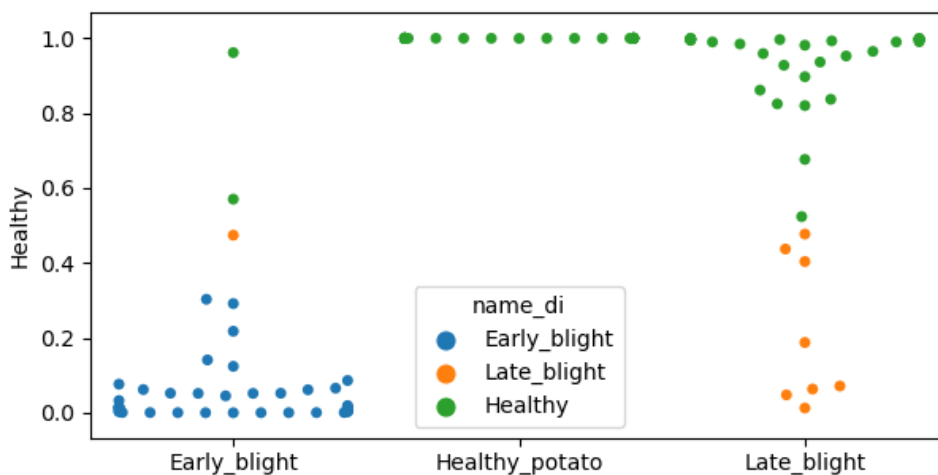
จากผลลัพธ์ในรูปที่ 4.51 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์หรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)



(ข)

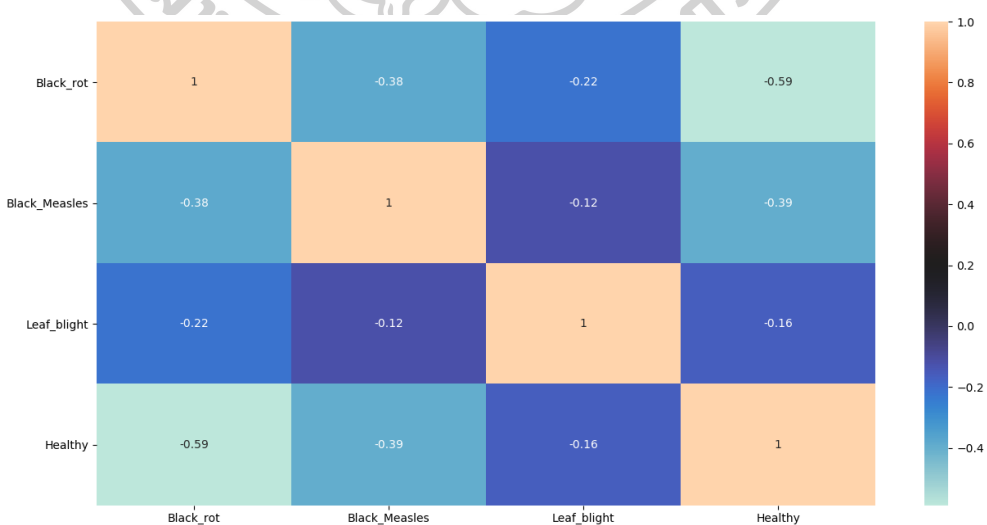


(ค)

รูปที่ 4.52 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

ในรูปที่ 4.52 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม จะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.52(ข) ผลของการจัดกลุ่มโรค Late blight แสดงเป็นสีส้มนั้นคำตอบที่ถูกจัดกลุ่มตรงกลับผลเฉยมีค่อนข้างน้อย โดยภาพส่วนใหญ่ได้ทำนายเป็นโรคชนิดอื่น ๆ ค่อนข้างมาก

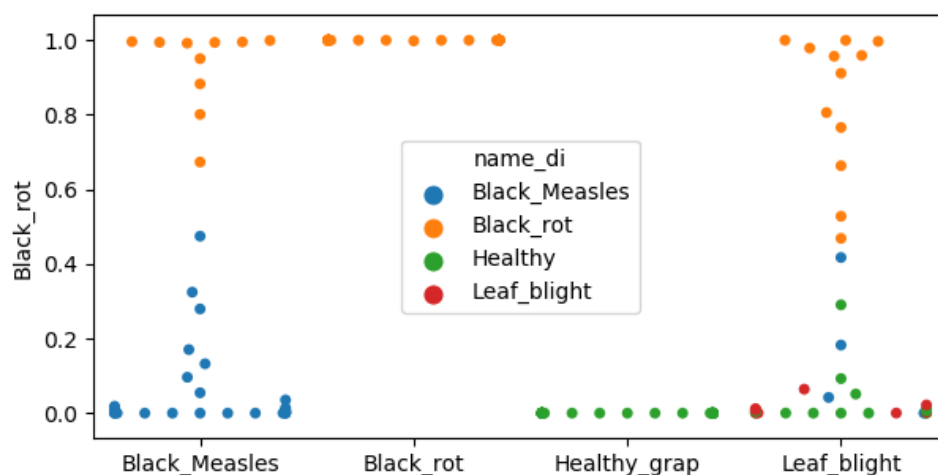
Original + Global



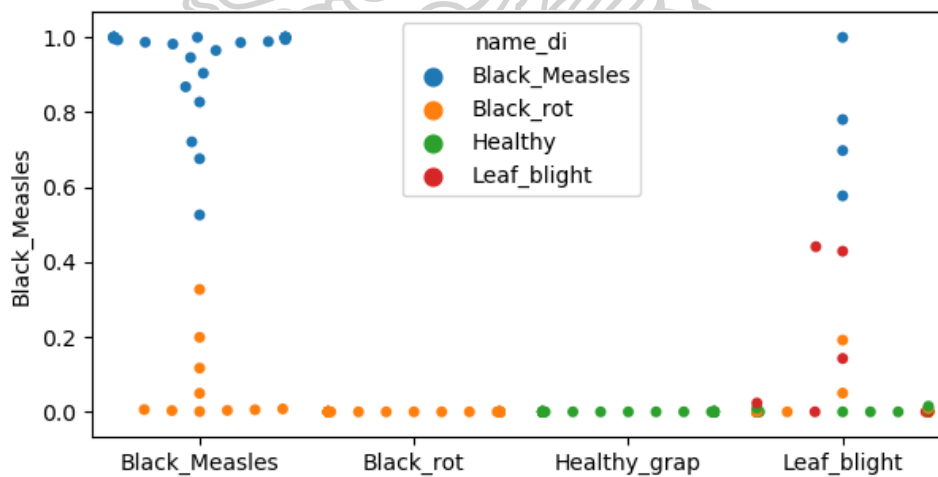
รูปที่ 4.53 Heatmap correlation matrix accuracy classification ภาพใบอ่อน (Original+Global)



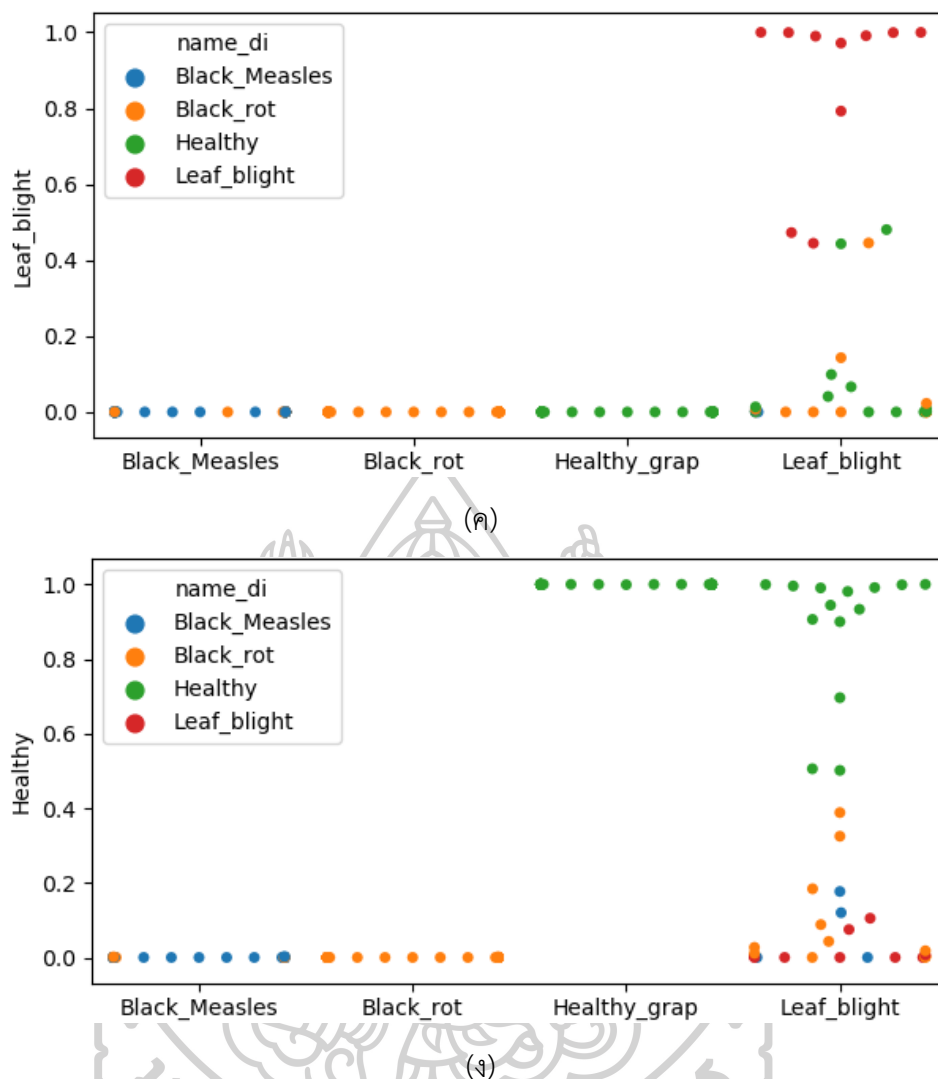
จากผลลัพธ์ในรูปที่ 4.53 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบของุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ก)

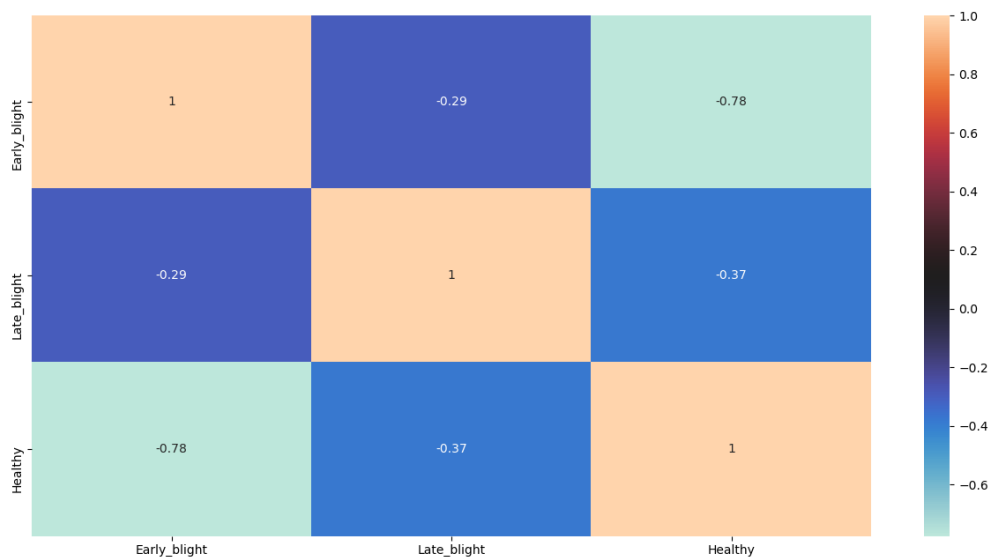


(ข)



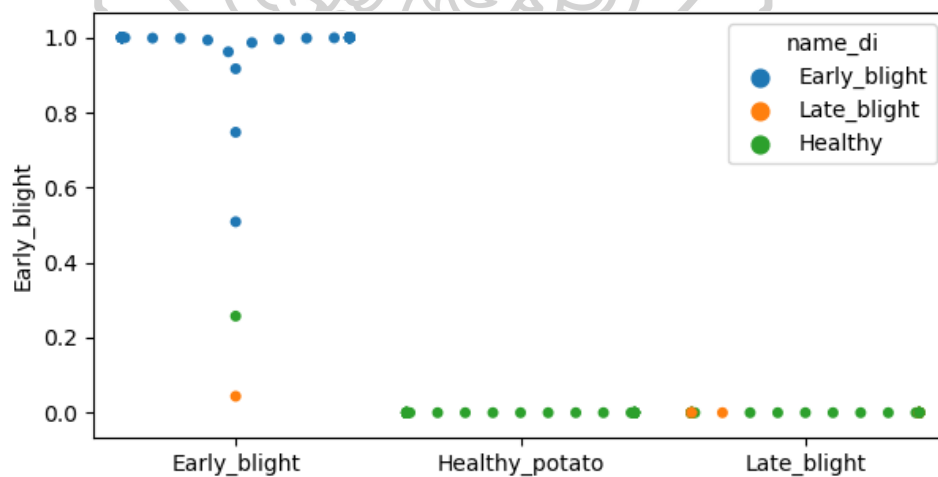
รูปที่ 4.54 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight  
(ง) Healthy

ในรูปที่ 4.54 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original+Global นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.54(ค) ผลของการจัดกลุ่มโรค Leaf blight แสดงเป็นสีแดงนั้นคำตอบที่ถูกจัดกลุ่มตรงกับผลเฉลยมีค่อนข้างน้อย โดยภาพส่วนใหญ่ได้ทำนายเป็นโรคชนิดอื่น ๆ ค่อนข้างมาก

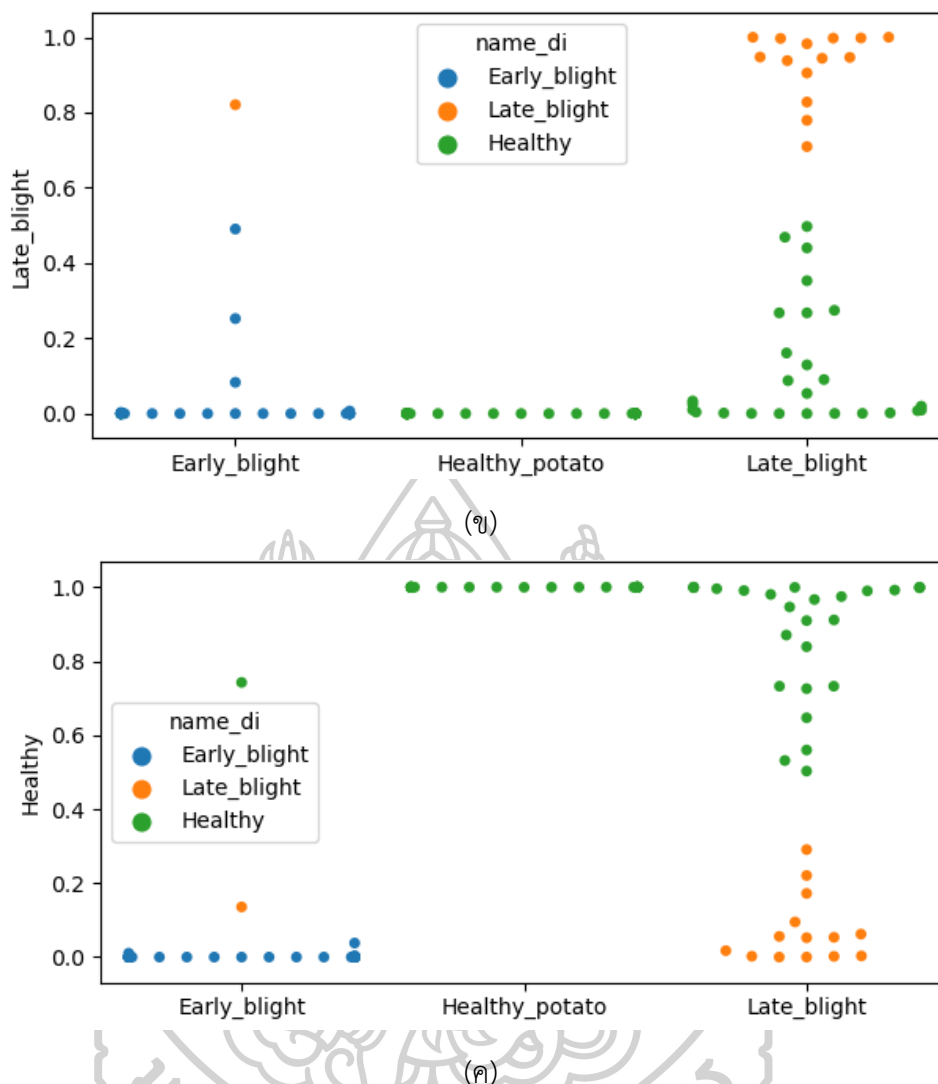


รูปที่ 4.55 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+Global)

จากผลลัพธ์ในรูปที่ 4.55 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง

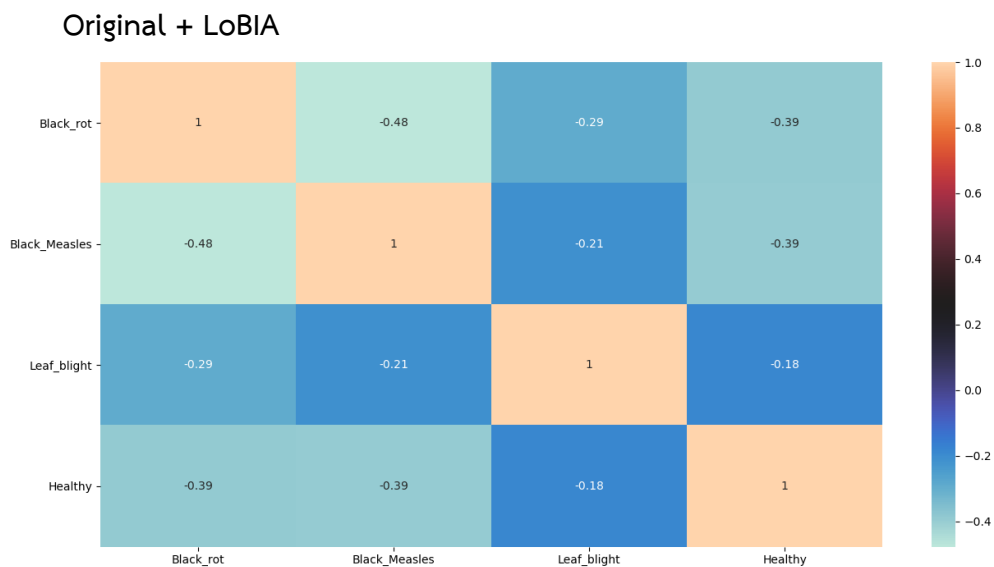


(ก)



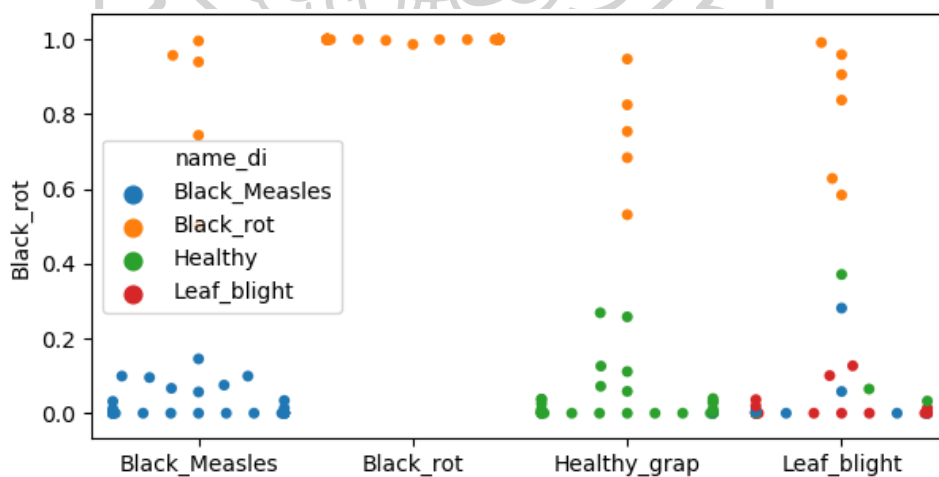
รูปที่ 4.56 ผลการจัดกลุ่ม Swarm plot (ข) Early blight (ค) Late blight (ค) Healthy

ในรูปที่ 4.56 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม จะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.56(ข) ผลของการจัดกลุ่มโรค Late blight แสดงเป็นสีส้มนั้นคำตอบที่ถูกจัดกลุ่มตรงกลับผลเฉลยมีค่อนข้างน้อย โดยภาพส่วนใหญ่ได้ทำนายเป็นโรคชนิดอื่น ๆ ค่อนข้างมาก

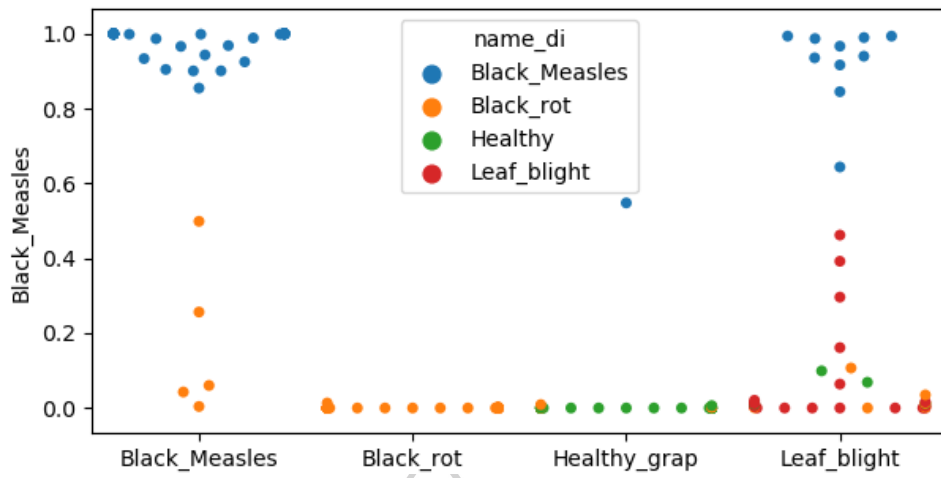


รูปที่ 4.57 Heatmap correlation matrix accuracy classification ภาพใบงุ่น (Original+LoBIA)

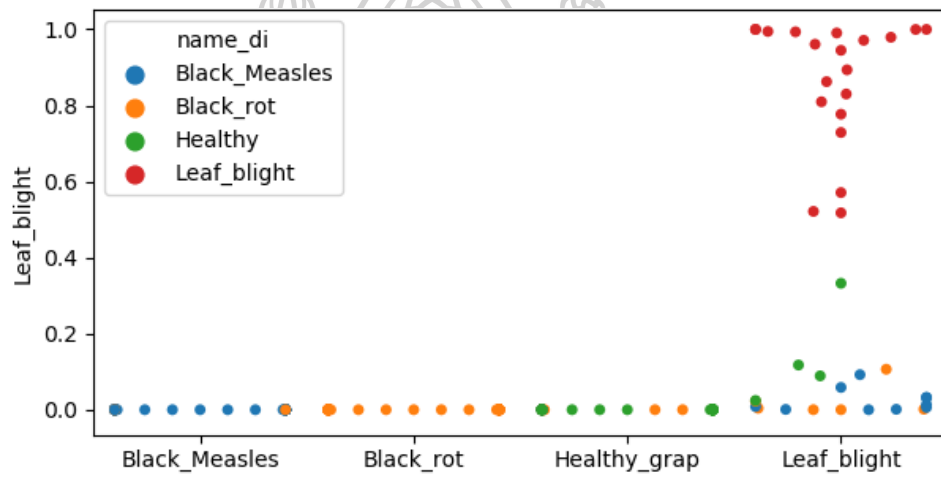
จากผลลัพธ์ในรูปที่ 4.57 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบงุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



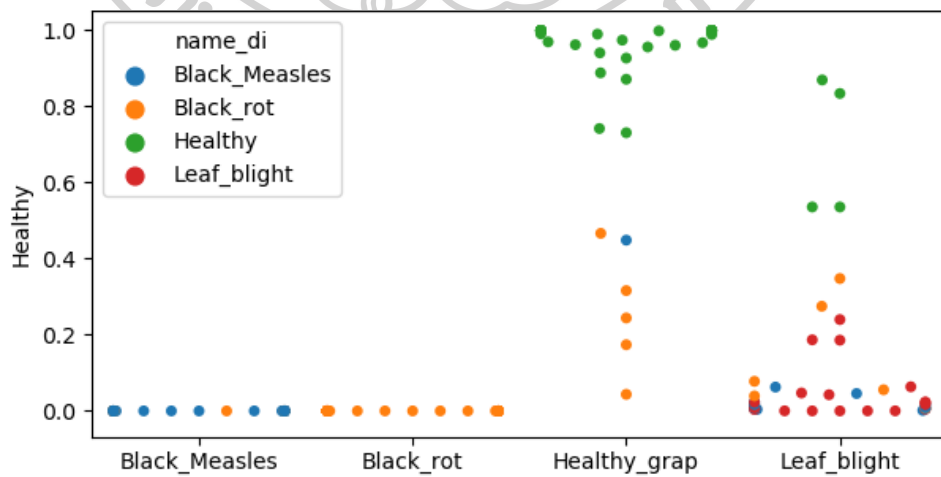
(ก)



(ข)



(ค)

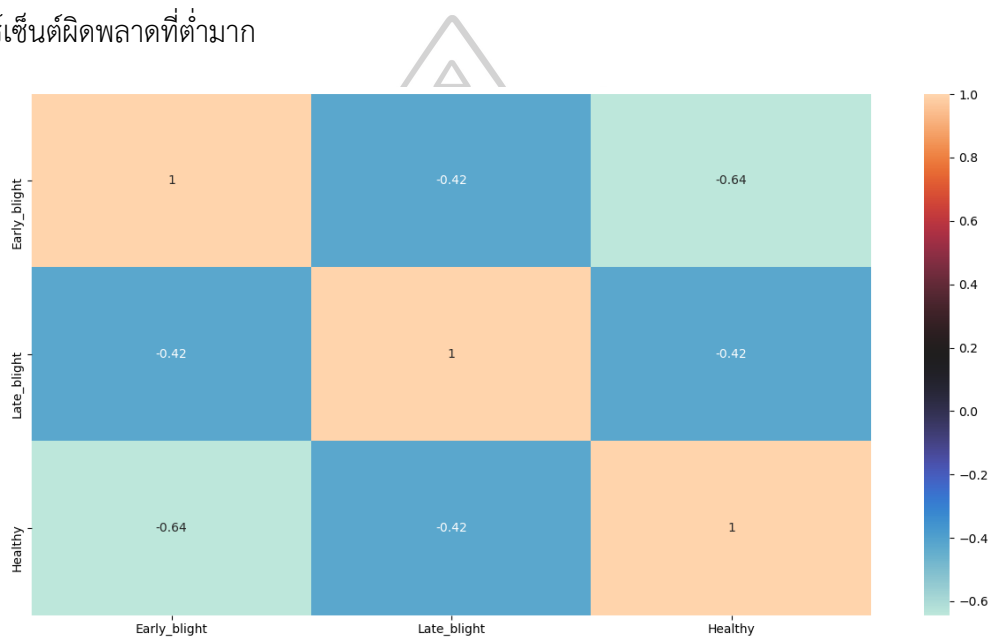


(ง)

รูปที่ 4.58 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight

(ง) Healthy

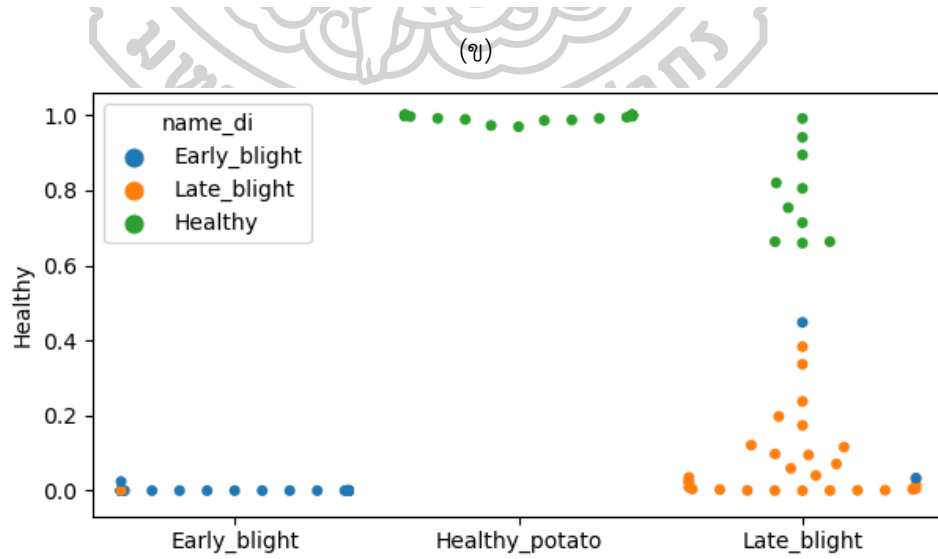
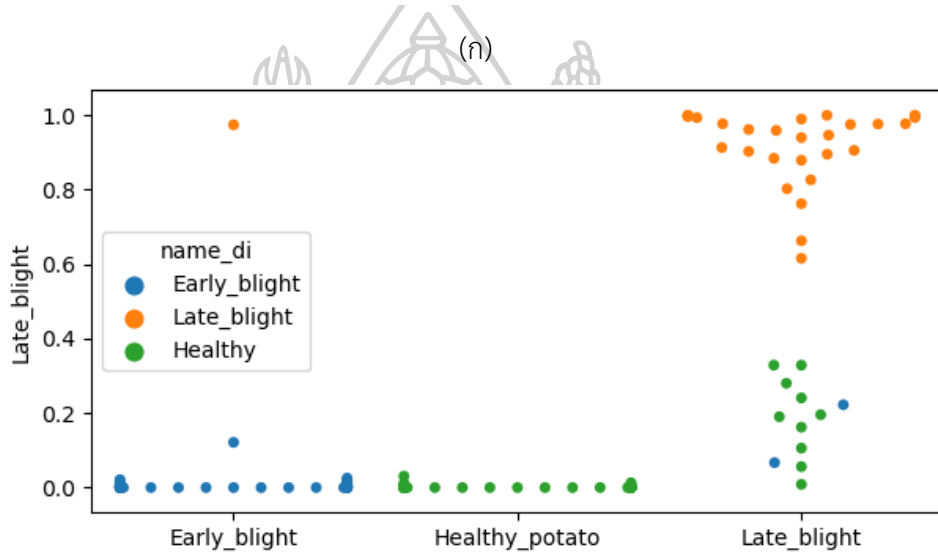
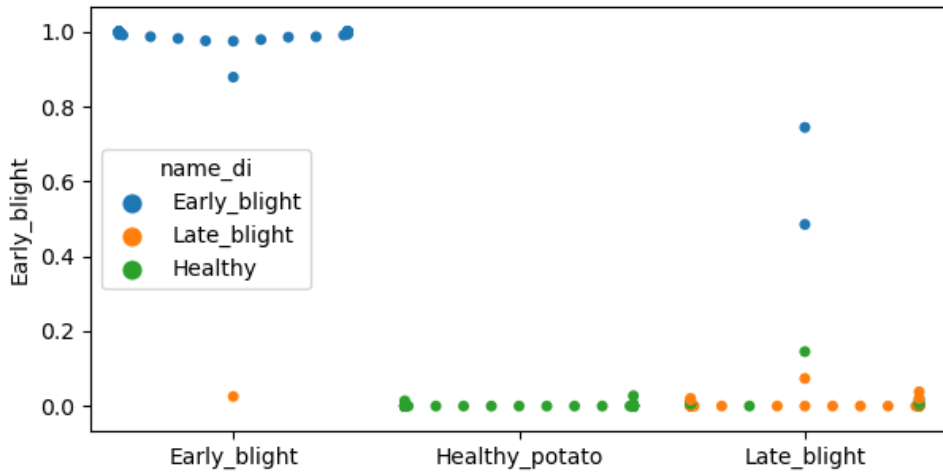
จากรูปที่ 4.58 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคมีความถูกต้องสูง แต่อาจมีโรคบางชนิดที่อาจยังมีการทำนายผิดอยู่เพียงเล็กน้อย ดังตัวอย่างรูปที่ 4.58(ก) แสดงผลของการจัดกลุ่มโรค Black rot แสดงเป็นจุดสีส้ม เห็นได้ว่าผลของการทำนายสามารถที่จะทำนายชนิดของโรคได้ถูกต้องทั้งหมด และรูปที่ 4.58(ข)(ค)(ง) ยังมีการทำนายที่ผิดอยู่บางส่วนแต่มีเปอร์เซ็นต์ผิดพลาดที่ต่ำมาก



รูปที่ 4.59 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA)

จากผลลัพธ์ในรูปที่ 4.59 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกต้องทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง

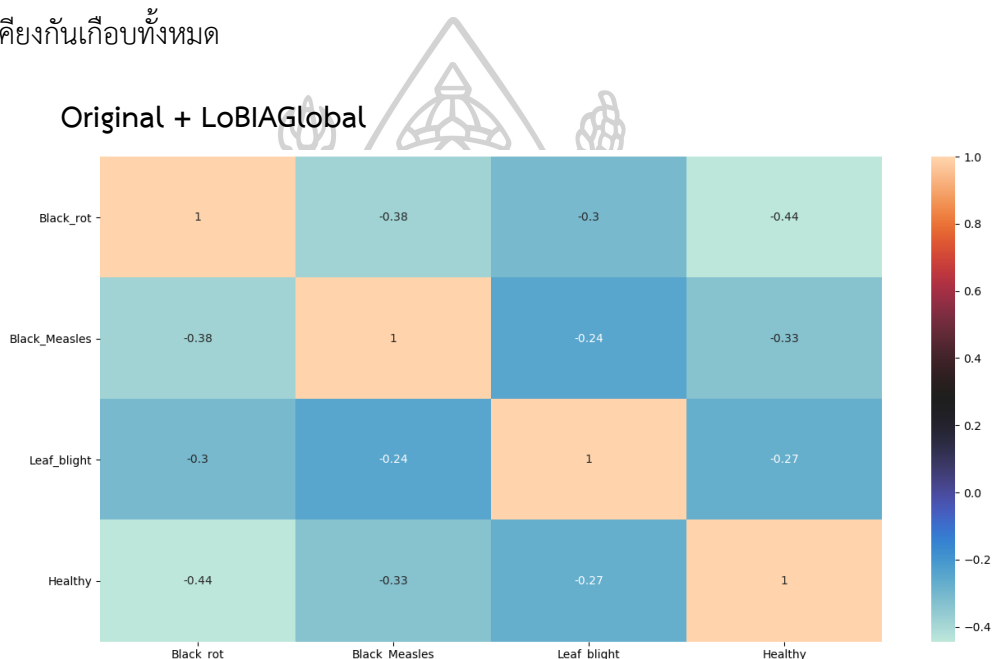




(ค)

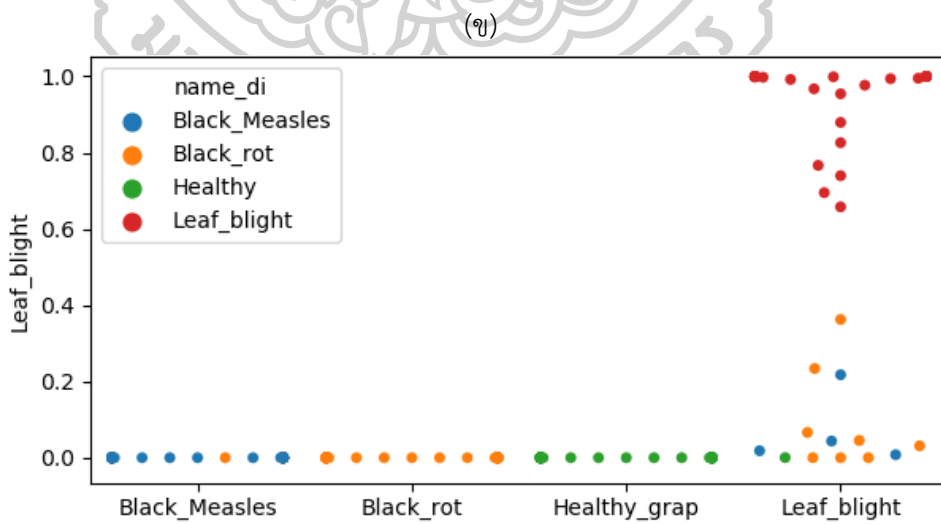
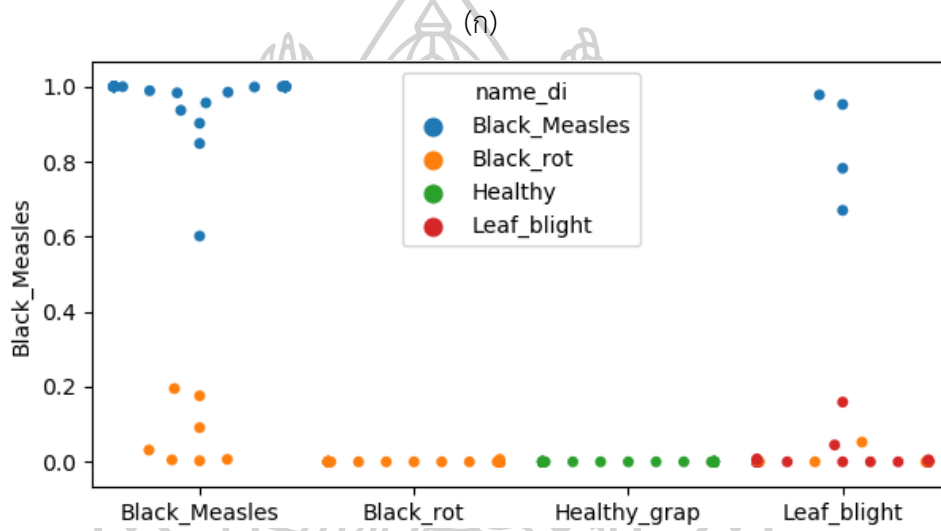
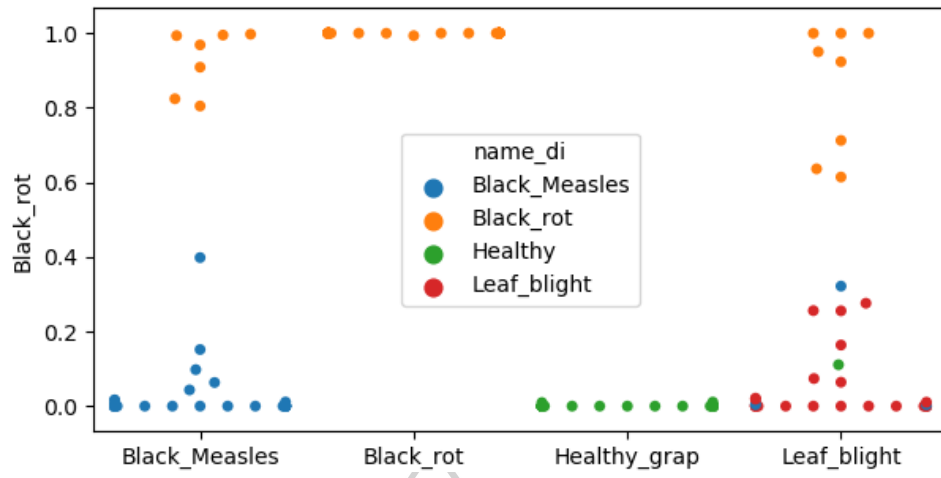
รูปที่ 4.60 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.60 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบมันฝรั่ง โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคได้มีความถูกต้องสูง แต่อาจมีโรคบางชนิดที่ยังทำนายผิดอยู่เล็กน้อย โดยจากรูปจะเห็นได้ว่าผลของการจัดกลุ่มนั้นมีข้อมูลที่ทำนายผิดพลาดเพียงเล็กน้อย และในส่วนผลที่ทำนายถูกต้องนั้นกลุ่มข้อมูลจะอยู่กันเป็นกลุ่มไม่กระจายสูงต่ำมากนักแสดงว่าข้อมูลที่ทำนายถูกนั้นทำนายถูกต้องในเปอร์เซ็นต์ที่สูงและใกล้เคียงกันเกือบทั้งหมด

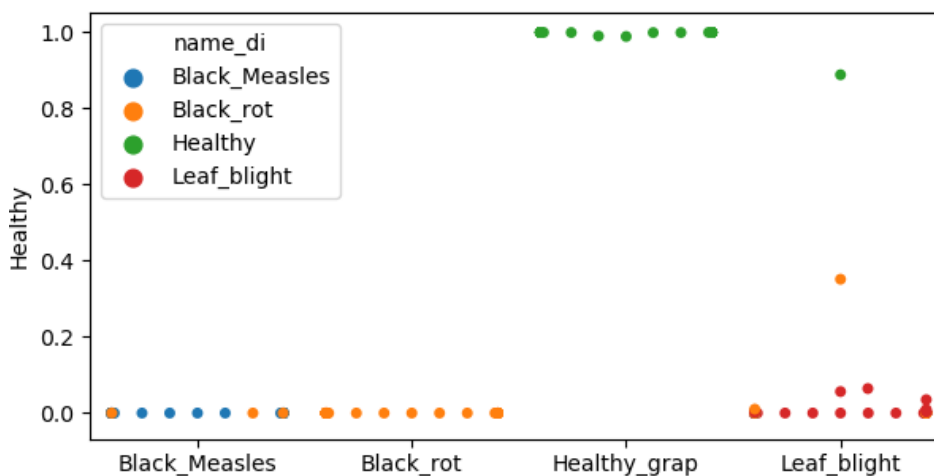


รูปที่ 4.61 Heatmap correlation matrix accuracy classification ภาพใบอ่อน (Original+LoBIAGlobal)

จากผลลัพธ์ในรูปที่ 4.61 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบอ่อนแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



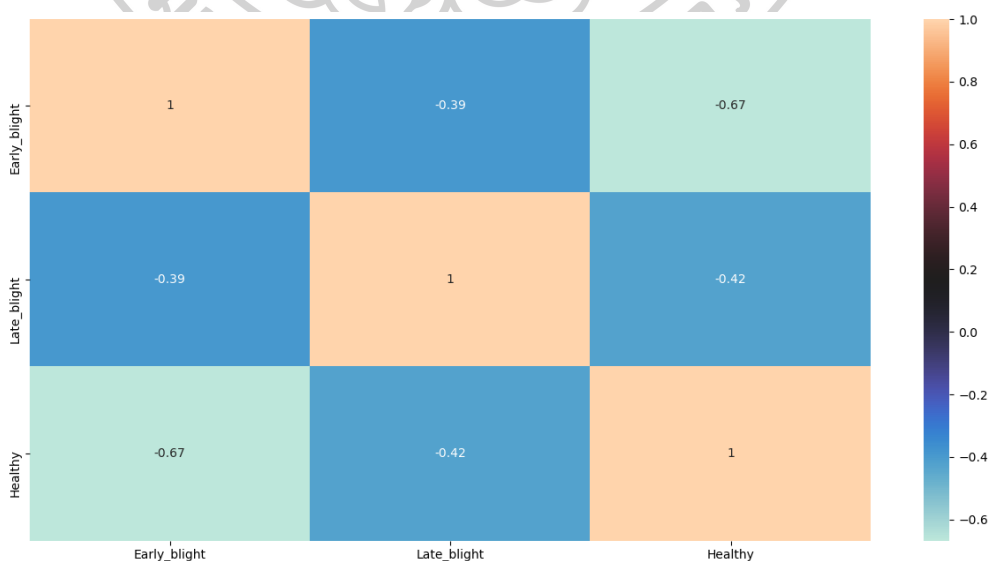
(a)



รูปที่ 4.62 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight

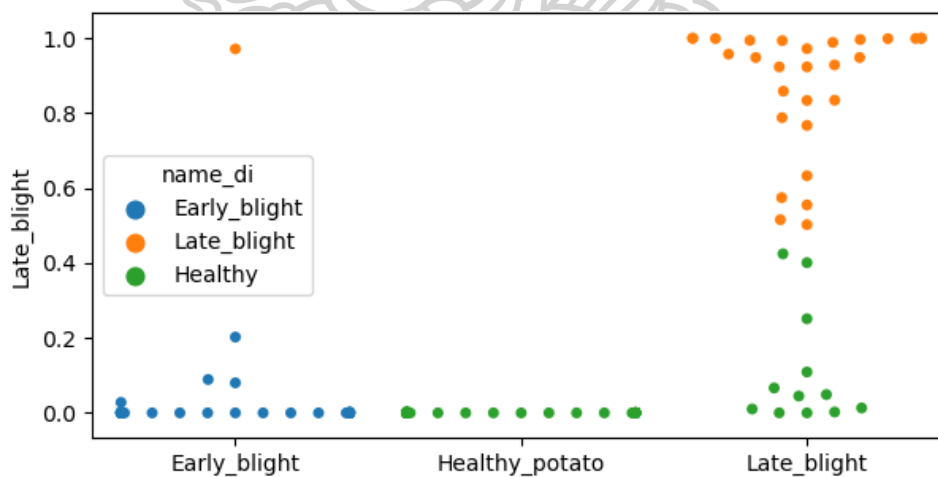
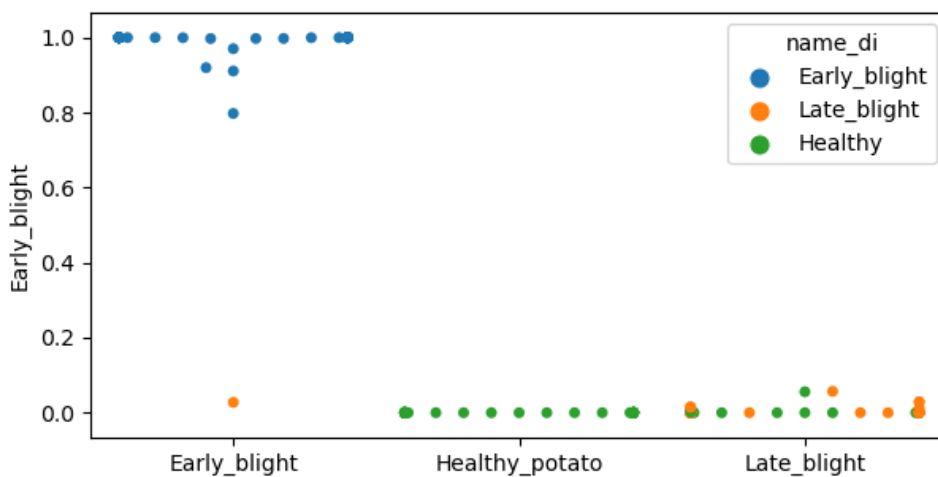
(ง) Healthy

จากรูปที่ 4.62 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำการทำนายและแกนนอนบ่งบอกถึงกลุ่มค่าตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่นำมาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจจะเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้

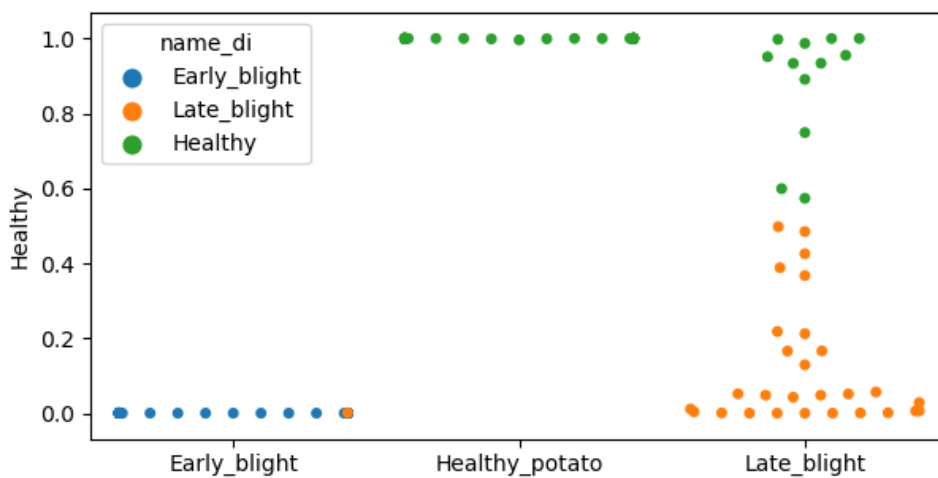


รูปที่ 4.63 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal)

จากผลลัพธ์ในรูปที่ 4.63 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ข)



(ค)

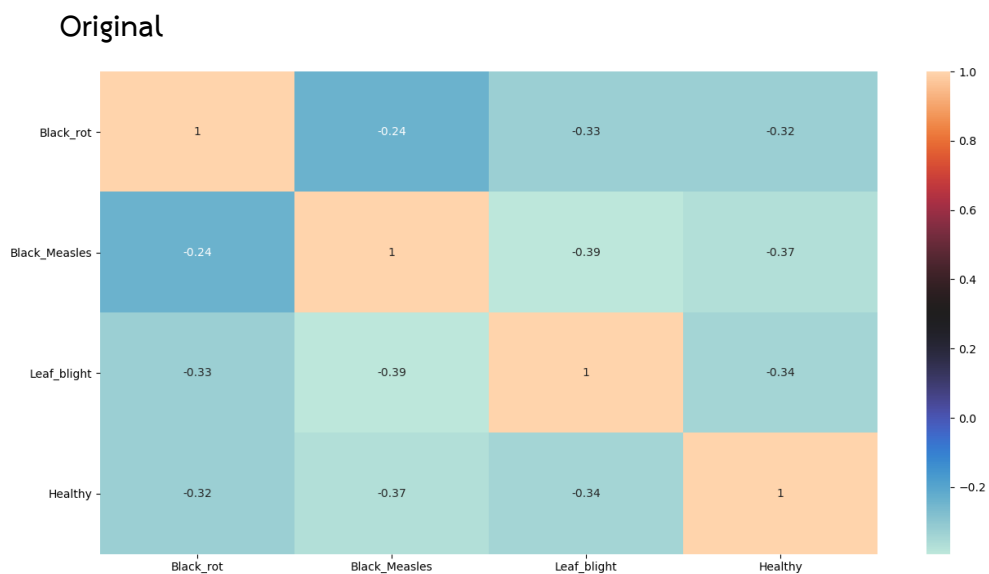
รูปที่ 4.64 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.64 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำการทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่นำมาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจจะเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้

จากผลการทดสอบของแต่ละกลุ่มข้อมูลฝึกสอนที่ฝึกสอนกับโมเดล MobileNet model ในรูปที่ 4.49 ถึงรูปที่ 4.64 เห็นได้ว่าเมื่อทำการทดสอบภาพของกลุ่มโรคของใบองุ่นและใบมันฝรั่งในกลุ่มของ Original + Global กับ Original + LoBIA และ Original + LoBIA นั้นให้ผลลัพธ์ในการจัดกลุ่มได้ที่ใกล้เคียงกัน แต่จำนวนภาพที่นำมาฝึกสอนนั้นกลุ่มข้อมูลภาพ Original + LoBIA นั้นมีจำนวนที่น้อยกว่ากลุ่มทั้งสองค่อนข้างมาก และเมื่อนำผลการจัดกลุ่มที่ได้มาคำนวณค่าเฉลี่ยของความถูกต้องในการจัดกลุ่มแล้ว ได้ผลค่าเฉลี่ยแสดงตารางที่ 4.2 (ใบองุ่น) ตารางที่ 4.3 (ใบมันฝรั่ง) และตารางที่ 4.4 ตารางผลทดสอบของแต่ละชนิดของชุดข้อมูลฝึกสอน

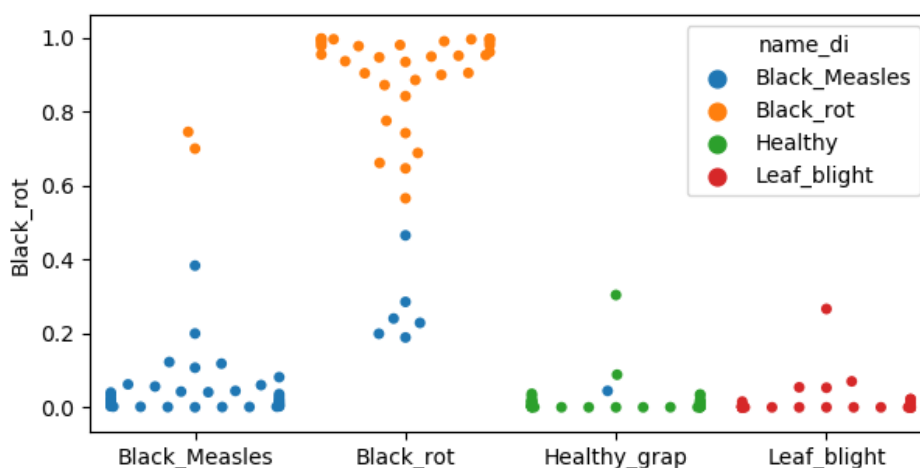
#### 4.4.4 VGG19

ในการทดสอบนี้เป็นการทดสอบวัดประสิทธิภาพในการจำแนกชนิดของโรคด้วยโมเดล VGG19 model มีชั้นของ Hidden layer จำนวน 19 ชั้น ใช้จำนวนรอบ (Epoch) ในการประมวลผล 50 รอบ หรือ 50 Epoch ผลลัพธ์ประสิทธิภาพของการจำแนกจะแสดงในรูปที่ 4.65 ถึงรูปที่ 4.80



รูปที่ 4.65 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original)

จากผลลัพธ์ในรูปที่ 4.65 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง

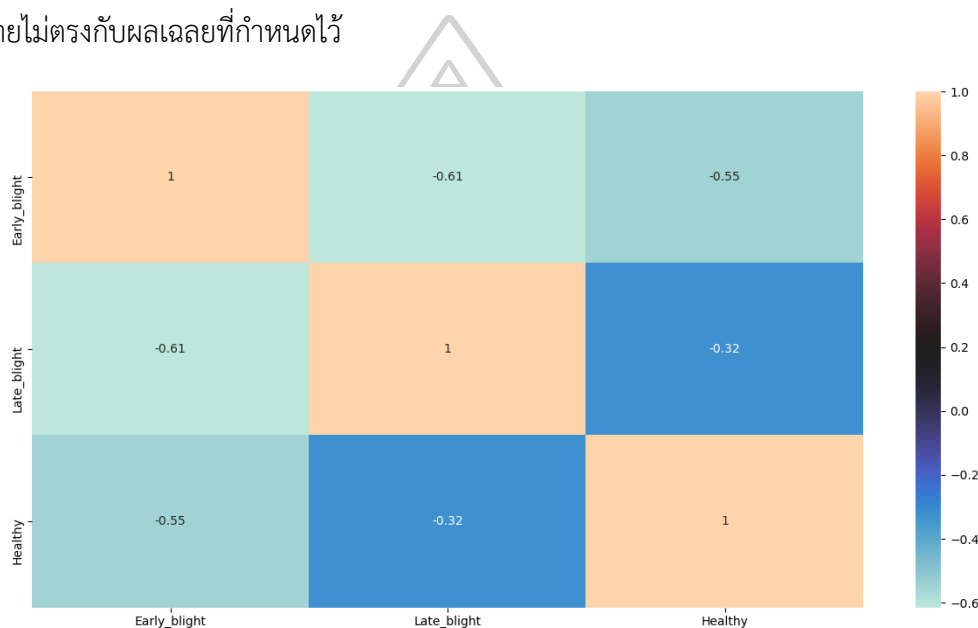






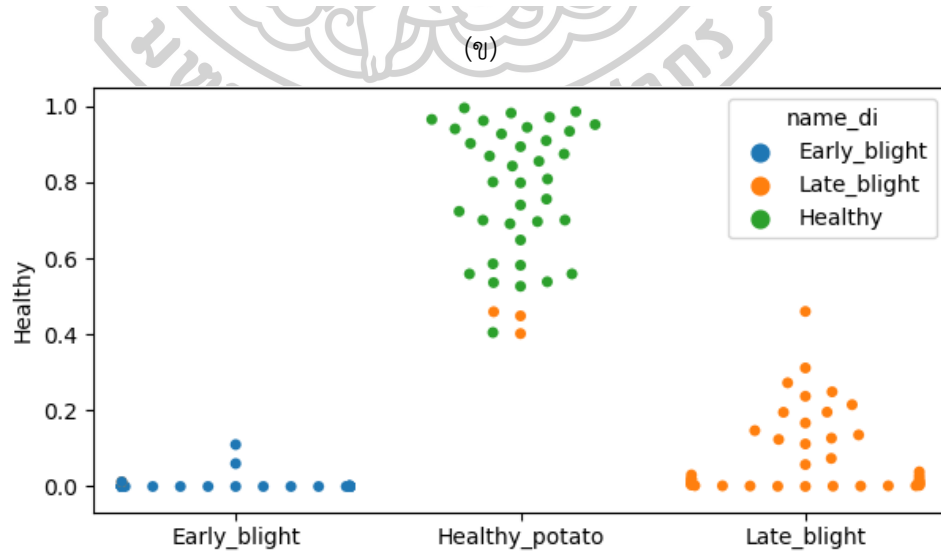
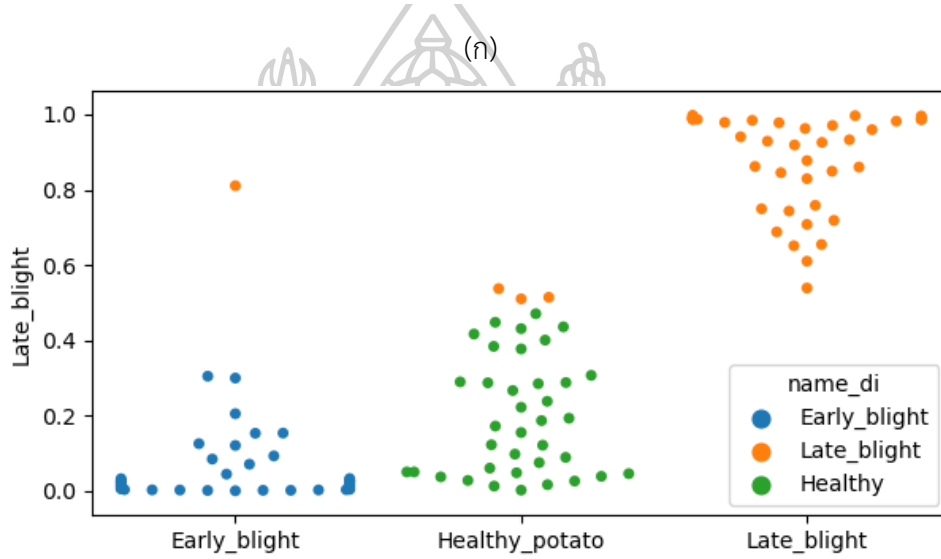
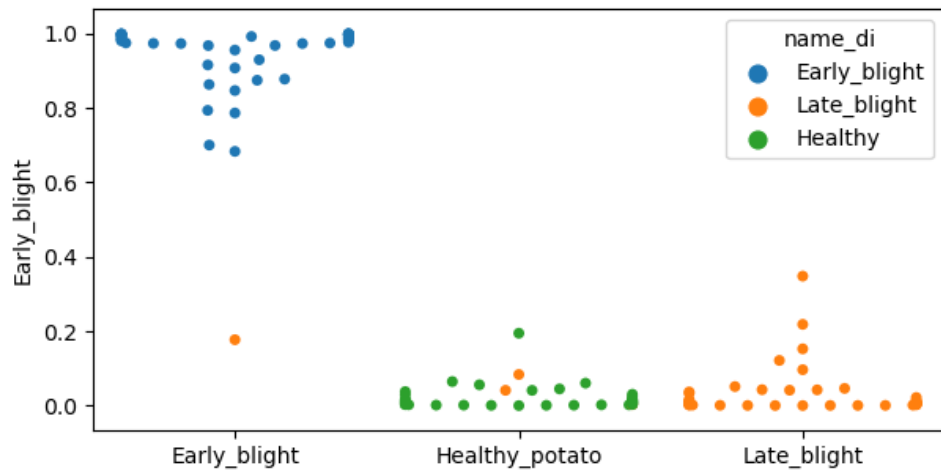
รูปที่ 4.66 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight  
(ง) Healthy

ในรูปที่ 4.66 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบองุ่นแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นค่าตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าค่าตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูง แต่ก็ยังมีการทำนายผิดอยู่ค่อนข้างมากกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.66(ก) ผลของการจัดกลุ่มโรค Black rot แสดงเป็นสีส้มนั้นส่วนมากจะถูกจัดกลุ่มตรงกลับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้



รูปที่ 4.67 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original)

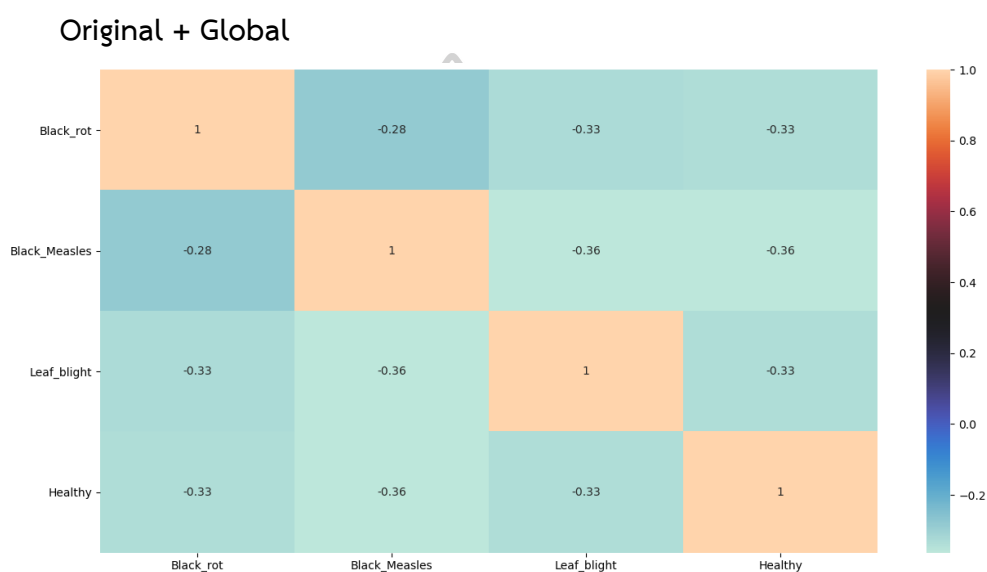
จากผลลัพธ์ในรูปที่ 4.67 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ค)

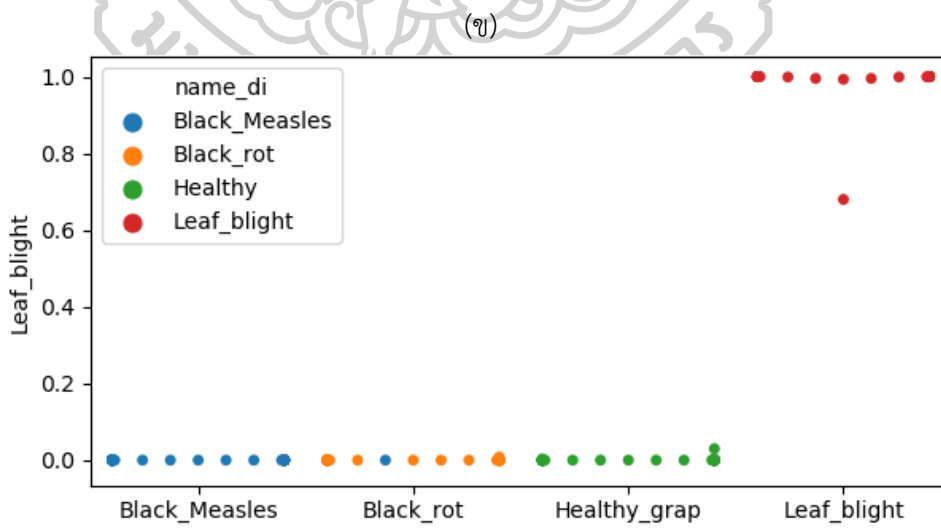
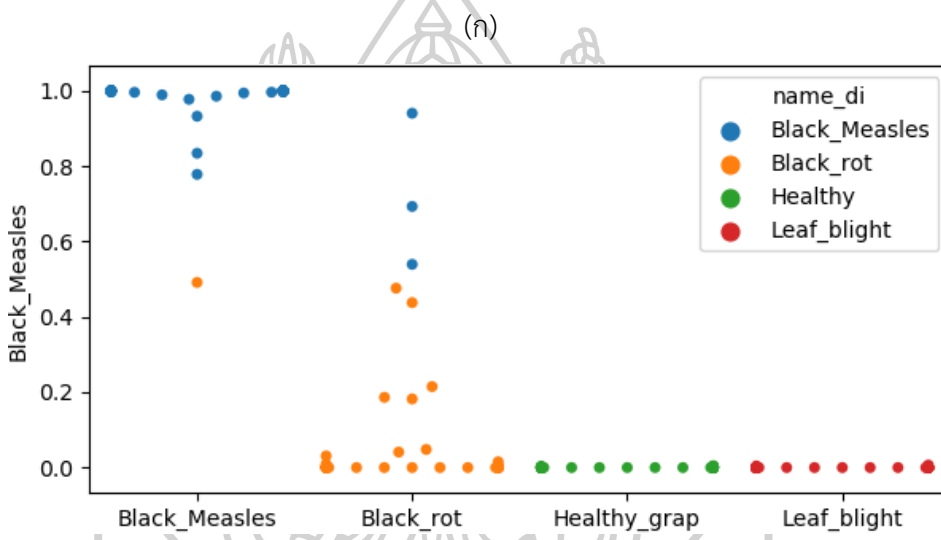
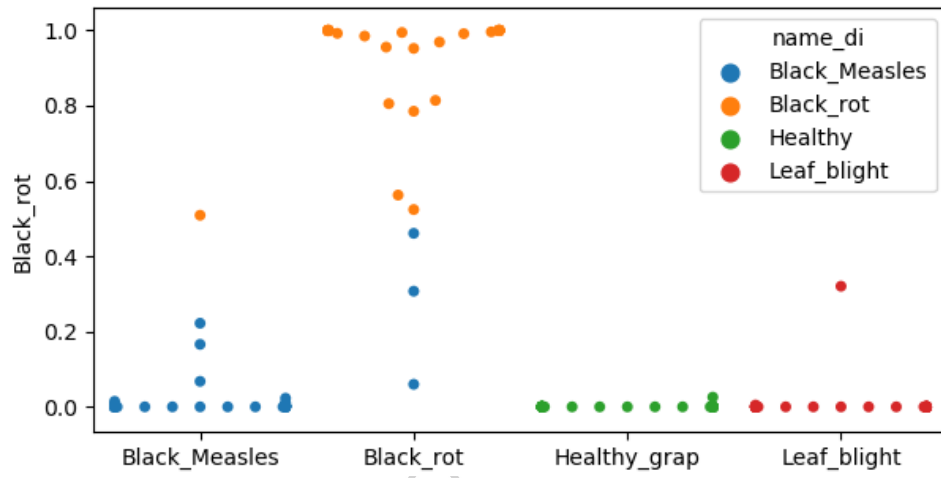
รูปที่ 4.68 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

ในรูปที่ 4.68 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original นั้นให้ผลที่ถูกต้องค่อนข้างสูงกว่า โมเดลอื่น ๆ แต่ก็ยังมีการทำนายผิดอยู่บางส่วนกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.68(ค) ผลของการจัดกลุ่มโรค Healthy แสดงเป็นสีเขียวนี้ส่วนมากจะถูกจัดกลุ่มตรงกับผลเฉลย แต่ก็ยังมีอีกบางส่วนที่ทำนายไม่ตรงกับผลเฉลยที่กำหนดไว้



รูปที่ 4.69 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+Global)

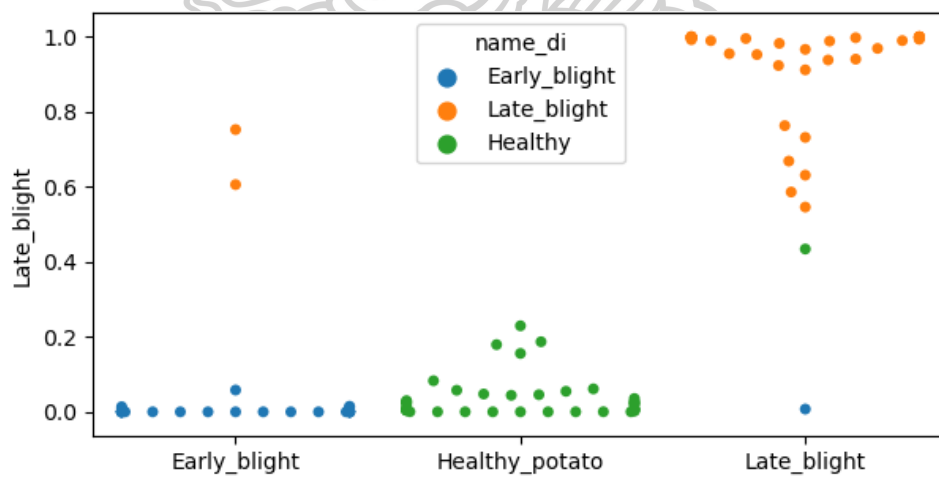
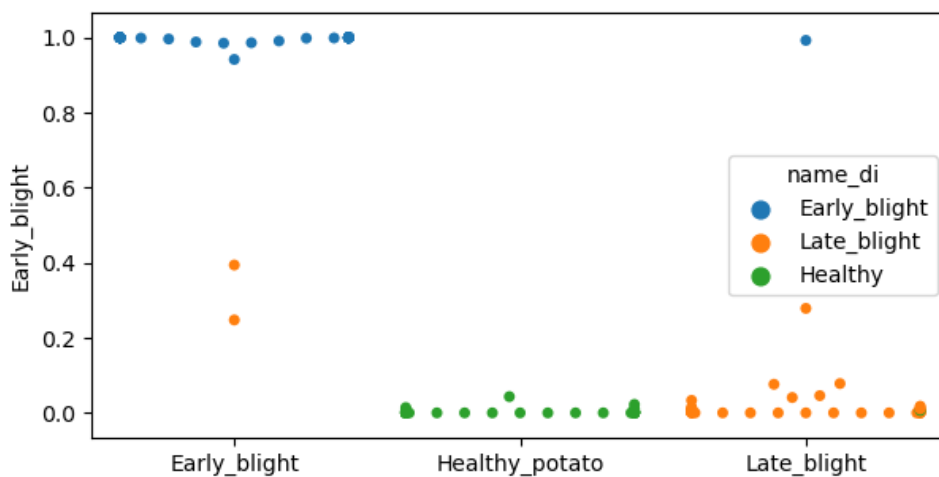
จากผลลัพธ์ในรูปที่ 4.69 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่เราทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(a)

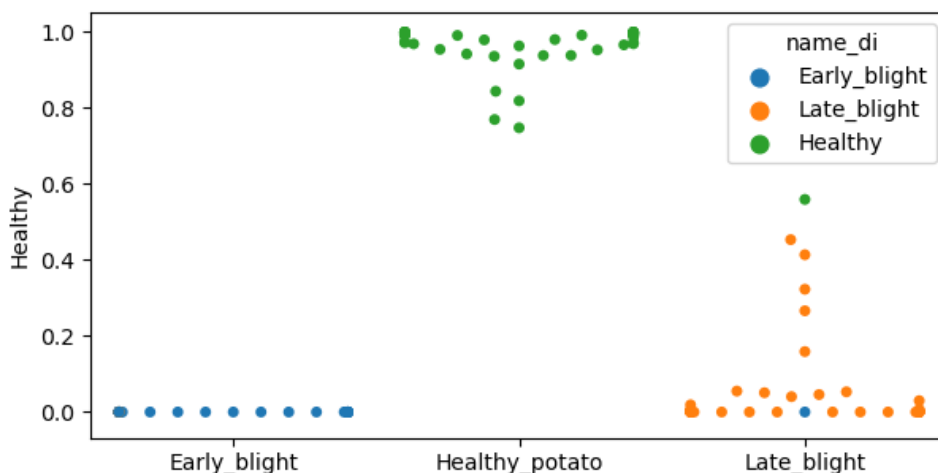


จากผลลัพธ์ในรูปที่ 4.71 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ข)



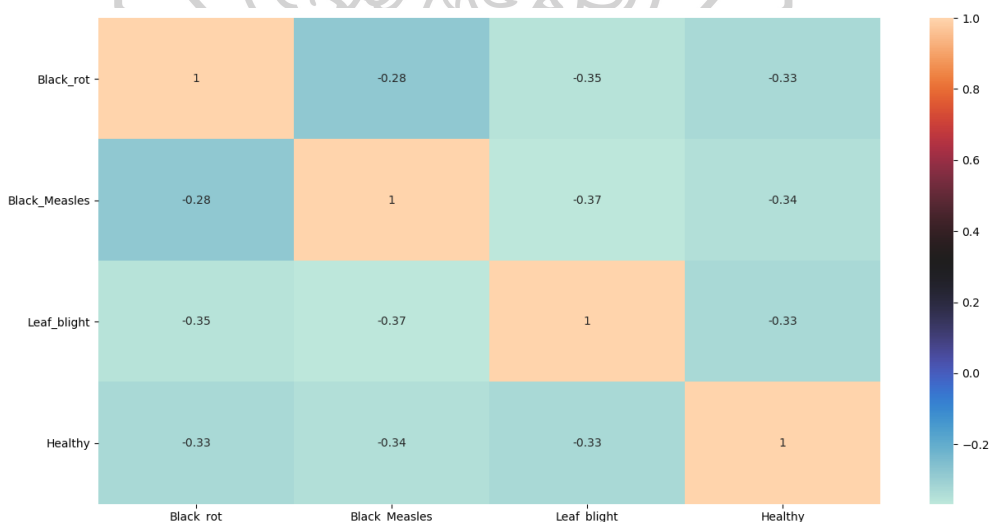


(ค)

รูปที่ 4.72 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

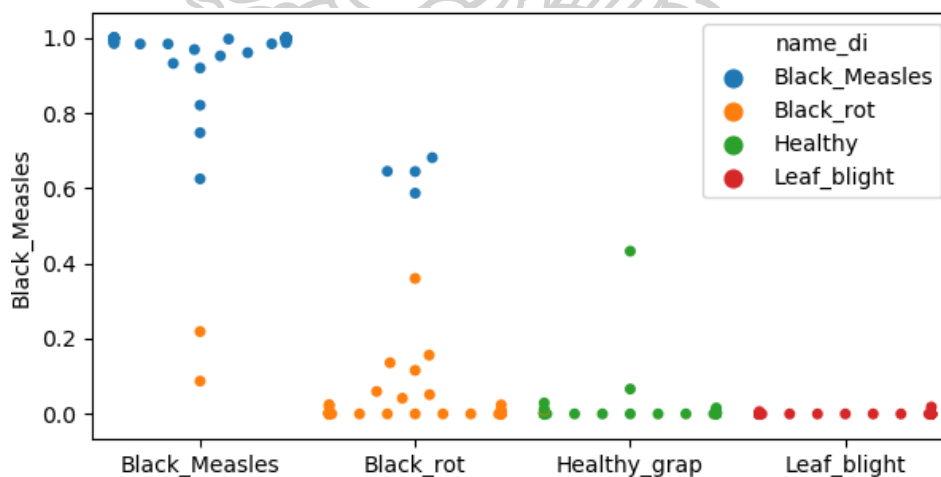
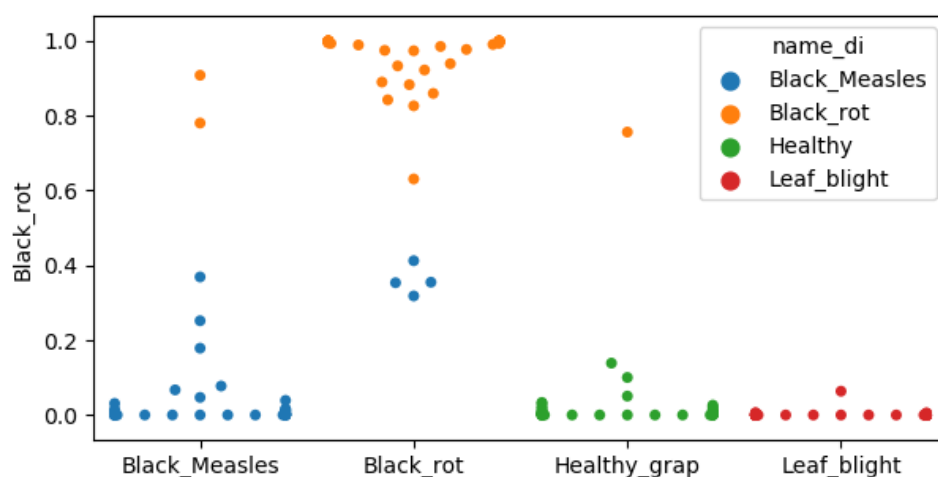
ในรูปที่ 4.72 แสดงถึงข้อมูลของการจัดกลุ่มการกระจายตัวของโรคบนใบมันฝรั่งแต่ละชนิด โรคโดยในแกนตั้งเป็นชนิดของโรคที่นำมาทดสอบและในแกนนอนจะเป็นคำตอบของโรคที่ถูกจัดกลุ่ม โดยจะเห็นได้ว่าคำตอบของการจัดกลุ่มของข้อมูลฝึกสอน Original+Global นั้นให้ผลที่ถูกต้องค่อนข้างสูงกว่าโมเดลอื่น ๆ แต่ก็ยังมีการทำนายผิดอยู่บางส่วนกับโรคบางชนิด โดยจะเห็นได้จากรูปที่ 4.72(ก) และรูปที่ 4.72(ข) ผลของการจัดกลุ่มโรคนั้นส่วนมากจะถูกจัดกลุ่มตรงกลับผลเฉลย

Original + LoBIA

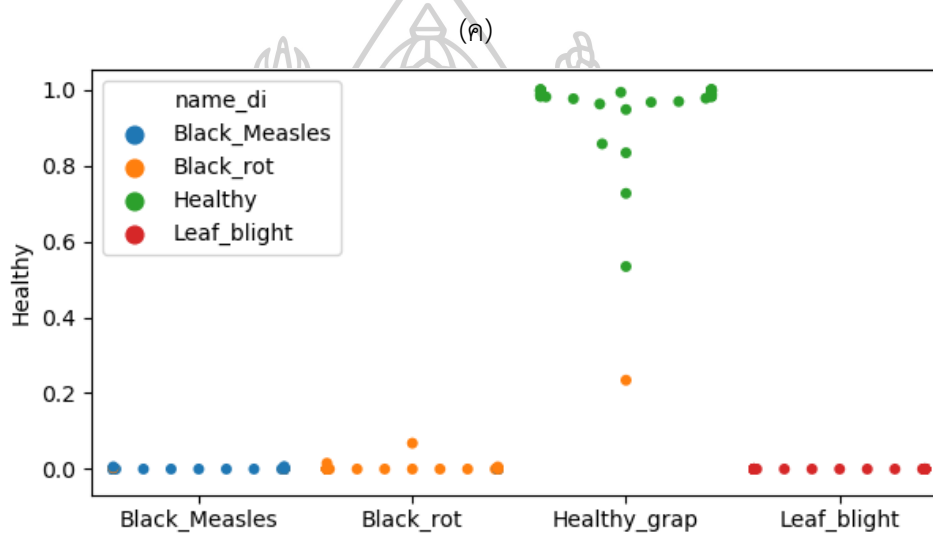
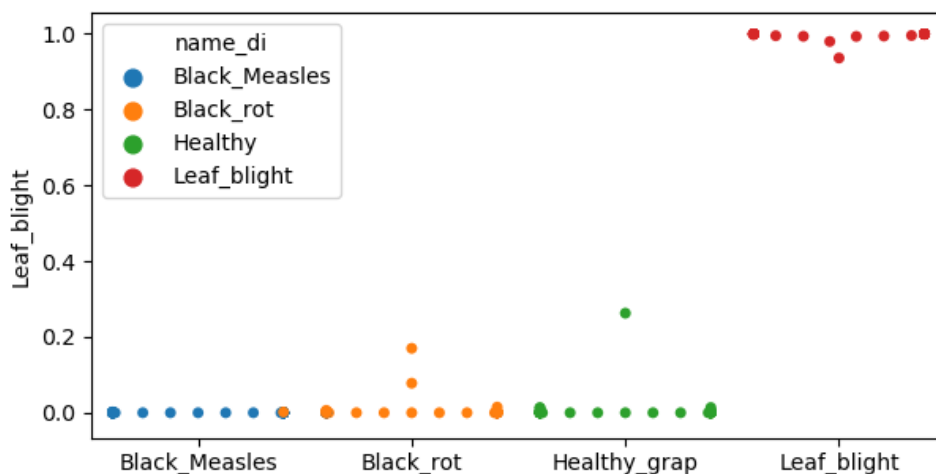


รูปที่ 4.73 Heatmap correlation matrix accuracy classification ภาพใบอ่อน (Original+LoBIA)

จากผลลัพธ์ในรูปที่ 4.73 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบของุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลลัพธ์จำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง

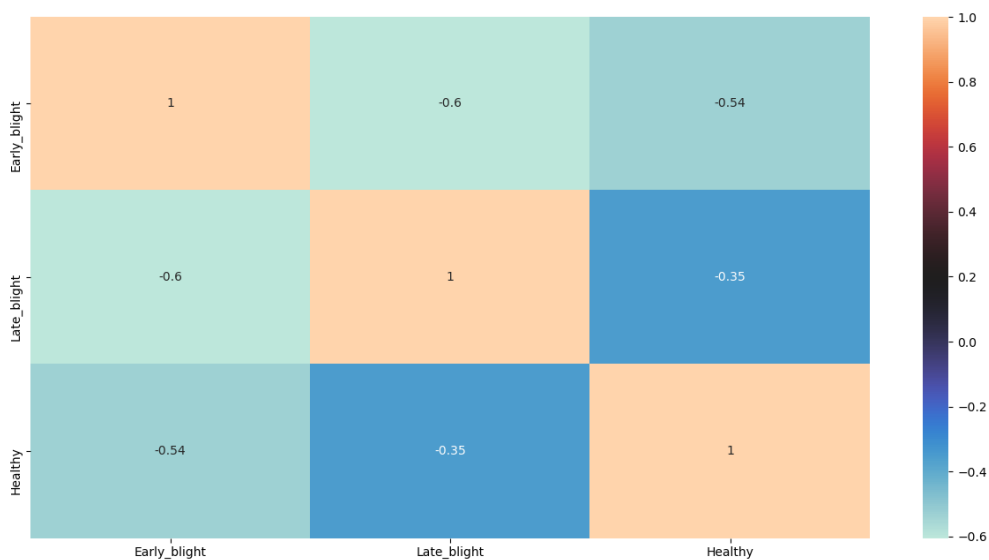


(ข)



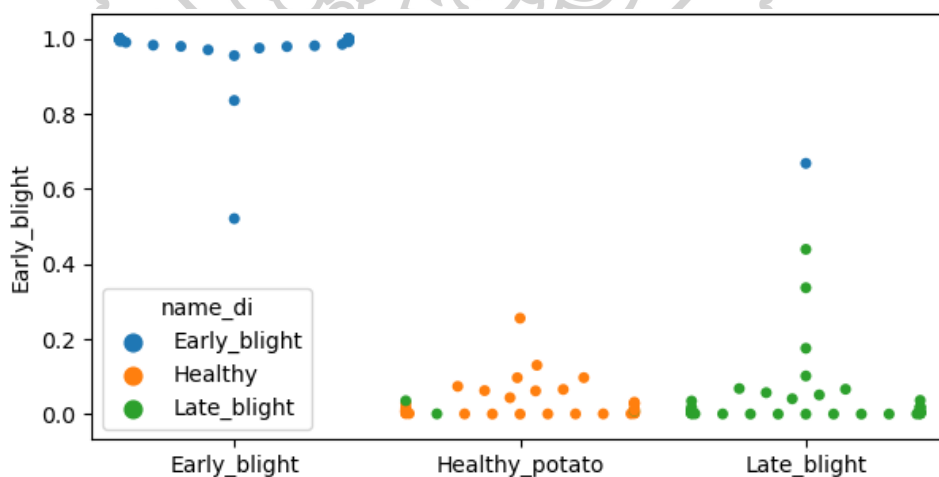
รูปที่ 4.74 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight  
(ง) Healthy

จากรูปที่ 4.74 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคมีความถูกต้องสูง แต่อาจมีโรคบางชนิดที่อาจยังมีการทำนายผิดอยู่เพียงเล็กน้อย ดังตัวอย่างรูปที่ 4.74(ก)(ข)(ค)(ง) เห็นได้ว่าผลของการทำนายสามารถที่จะทำนายชนิดของโรคได้ถูกต้องเกือบทั้งหมดแต่อาจยังมีการทำนายที่ผิดอยู่บางส่วนแต่มีเปอร์เซ็นต์ผิดพลาดที่ต่ำมาก

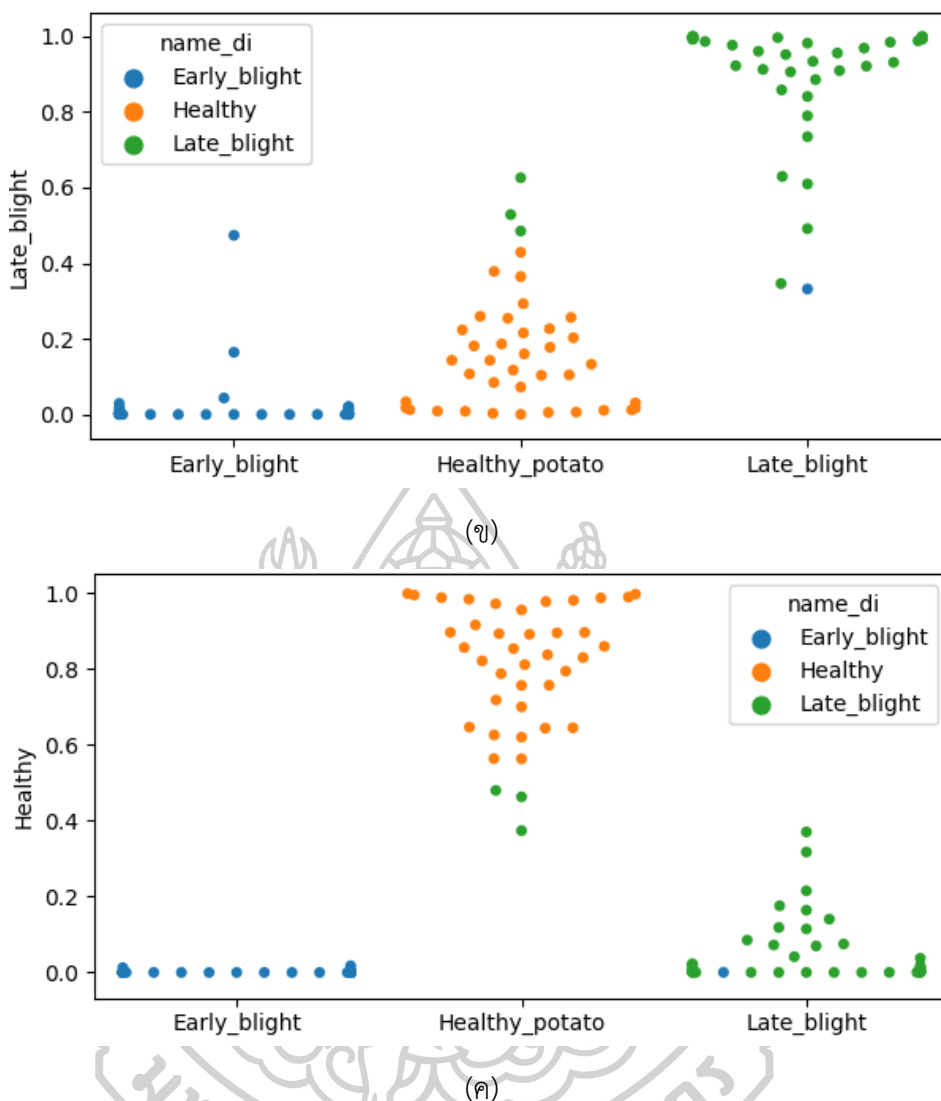


รูปที่ 4.75 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIA)

จากผลลัพธ์ในรูปที่ 4.75 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



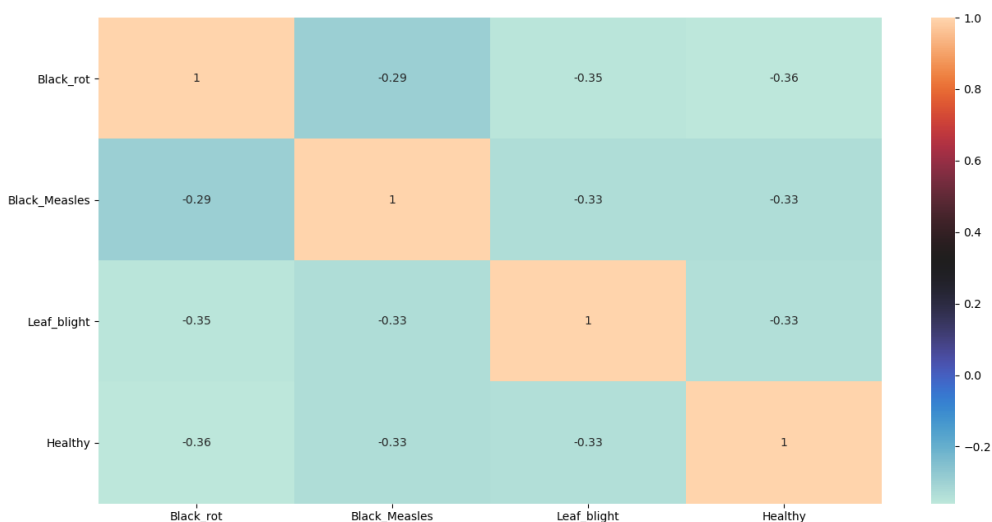
(ก)



รูปที่ 4.76 ผลการจัดกลุ่ม Swarm plot (ข) Early blight (ค) Healthy

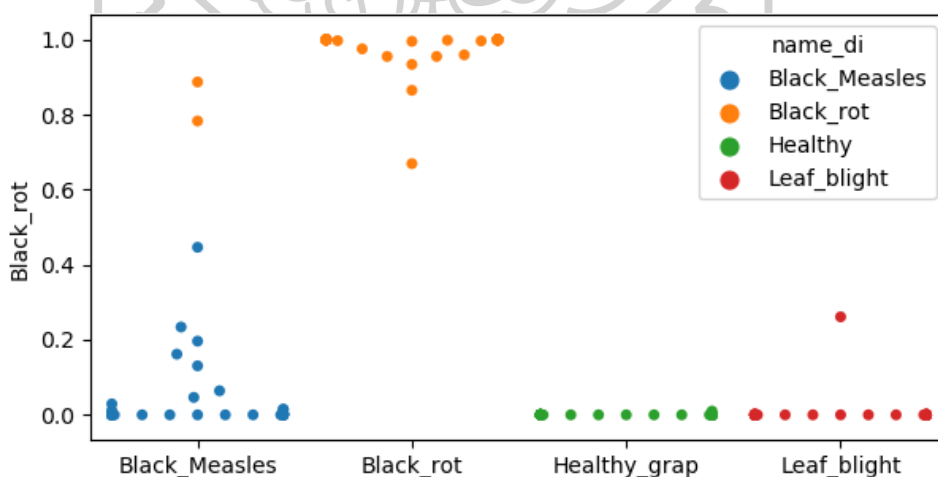
จากรูปที่ 4.76 แสดงผลการจัดกลุ่มที่โมเดลทำนายไว้โดยแบ่งผลของการทำนายออกเป็นแต่ละชนิดของโรคบนใบมันฝรั่ง โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำกรทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIA นั้นสามารถที่จะตอบหรือทำนายชนิดของโรคได้มีความถูกต้องสูง แต่อาจมีโรคบางชนิดที่ยังทำนายผิดอยู่เล็กน้อย โดยจากรูปจะเห็นได้ว่าผลของการจัดกลุ่มนั้นมีข้อมูลที่ทำนายผิดพลาดเพียงเล็กน้อย และในส่วนผลที่ทำนายถูกต้องนั้นกลุ่มข้อมูลจะอยู่กันเป็นกลุ่มไม่กระจายสูงต่ำมากนักแสดงว่าข้อมูลที่ทำนายถูกนั้นทำนายถูกต้องในเปอร์เซ็นต์ที่สูงและใกล้เคียงกันเกือบทั้งหมด

## Original + LoBIAGlobal

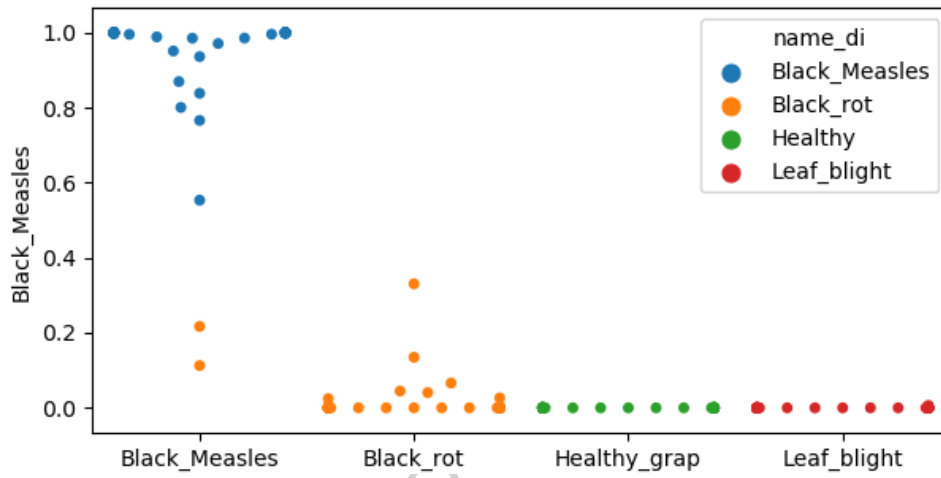


รูปที่ 4.77 Heatmap correlation matrix accuracy classification ภาพใบองุ่น (Original+LoBIAGlobal)

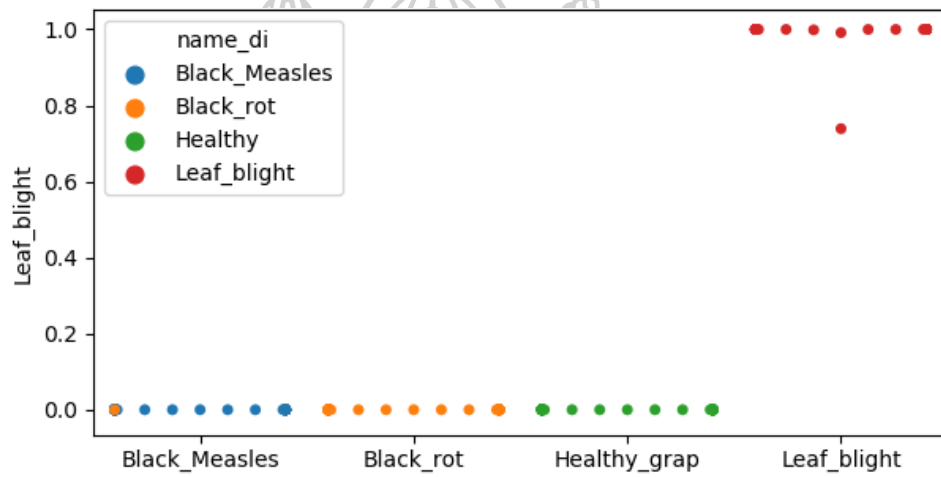
จากผลลัพธ์ในรูปที่ 4.77 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบองุ่นแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



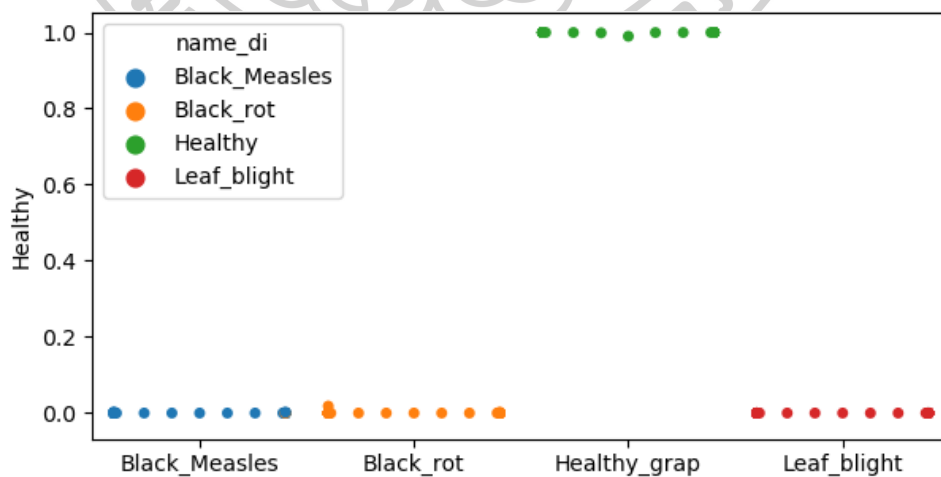
(ก)



(ก)



(ข)



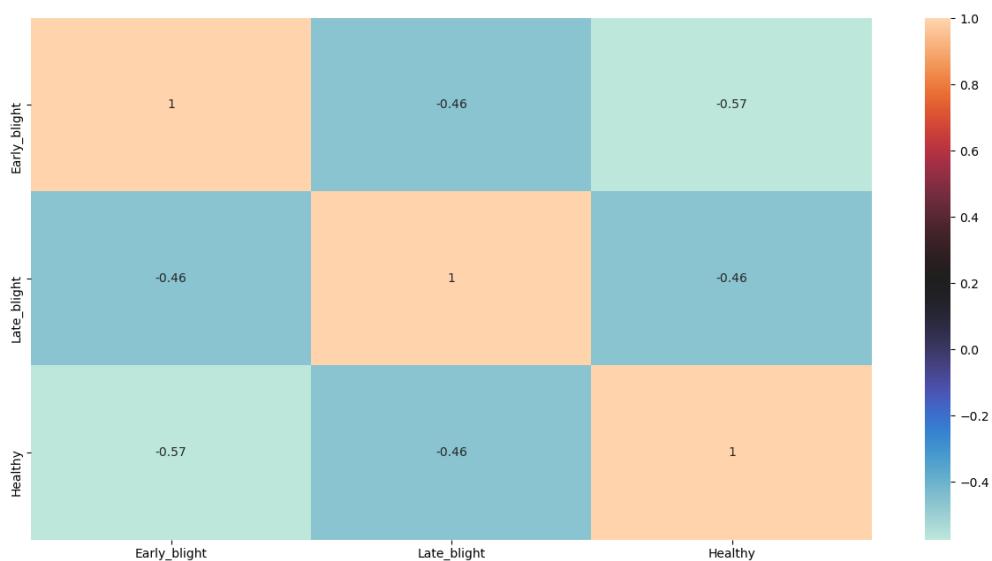
(ค)

รูปที่ 4.78 ผลการจัดกลุ่ม Swarm plot (ก) Black rot (ข) Black measles (ค) Leaf blight

(ง) Healthy

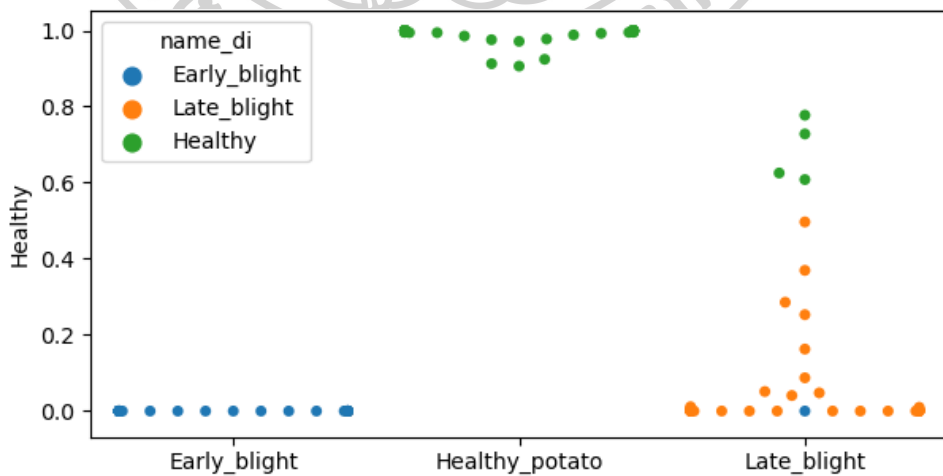
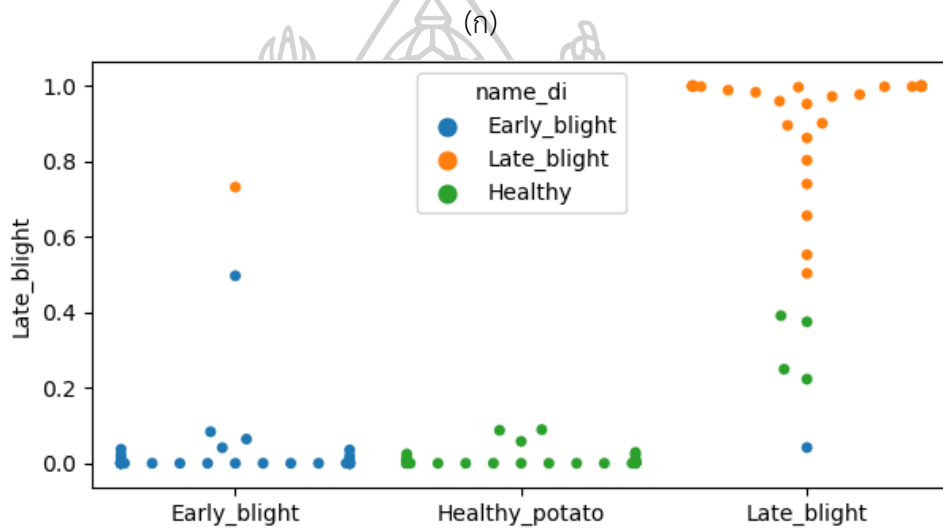
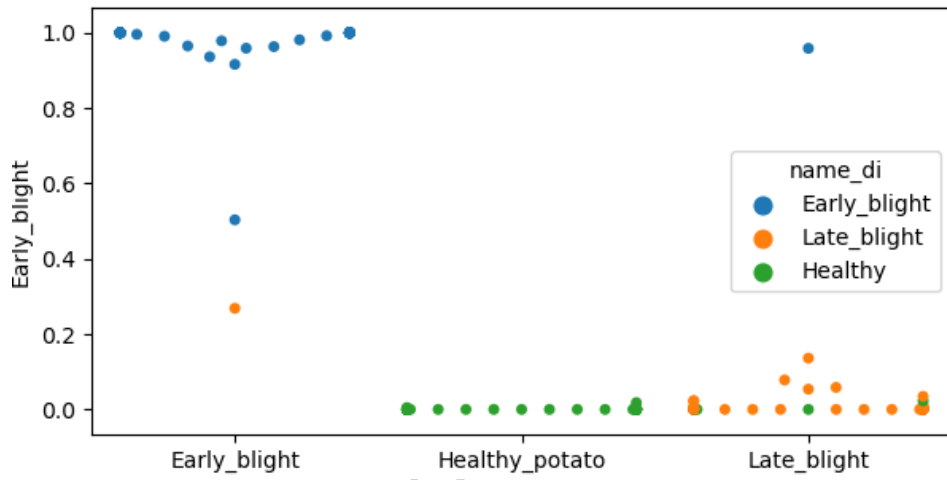


จากรูปที่ 4.78 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของใบองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำการทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่น่ามาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้



รูปที่ 4.79 Heatmap correlation matrix accuracy classification ภาพใบมันฝรั่ง (Original+LoBIAGlobal)

จากผลลัพธ์ในรูปที่ 4.79 แสดงให้เห็นว่าผลของการจัดกลุ่มโรคบนใบมันฝรั่งแต่ละโรคให้ผลลัพธ์ของปริมาณของข้อมูลที่ทำนายในแต่ละคลาสเป็นโรคชนิดใดในแต่ละคลาส โดยในแกนตั้งเป็นผลหรือจำนวนของข้อมูลที่ทำนายได้ และแกนนอนเป็นผลของกลุ่มข้อมูลที่ควรทำนายได้ จะเห็นได้ว่าผลลัพธ์มีค่าเป็น 1 แสดงถึงปริมาณของกลุ่มข้อมูลที่ถูกทำนายมากที่สุด โดยผลที่ได้ควรออกมาเป็นเลขสูงสุดในแนวทแยง



(ค)

รูปที่ 4.80 ผลการจัดกลุ่ม Swarm plot (ก) Early blight (ข) Late blight (ค) Healthy

จากรูปที่ 4.80 แสดงถึงผลของการจัดกลุ่มของข้อมูลที่ถูกทำนายด้วยโมเดลข้างต้นโดยแบ่งผลของการจัดกลุ่มออกเป็นแต่ละชนิดโรคของไบโองุ่น โดยในแกนตั้งบ่งบอกถึงชนิดของโรคที่ทำการทำนายและแกนนอนบ่งบอกถึงกลุ่มคำตอบที่โมเดลต้องทำการทำนายจัดกลุ่ม จากผลที่แสดงข้างต้นเห็นได้ว่าโมเดลที่ฝึกสอนโดยกลุ่มข้อมูลฝึกสอน Original+LoBIAGlobal นั้นสามารถที่จะทำนายข้อมูลที่นำมาทดสอบได้โดยมีความถูกต้องที่ดี แต่ก็อาจยังมีการตอบผิดพลาดอยู่บ้างเพียงเล็กน้อย ซึ่งอาจจะเกิดจากจำนวนข้อมูลฝึกสอนที่มากเกินไปหรือปัจจัยอื่น ๆ ก็ได้

จากผลการทดสอบของแต่ละกลุ่มข้อมูลฝึกสอนที่ฝึกสอนกับโมเดล VGG19 model ในรูปที่ 4.65 ถึงรูปที่ 4.80 เห็นได้ว่าเมื่อทำการทดสอบภาพของกลุ่มโรคของไบโองุ่นและใบมันฝรั่งในกลุ่มของ Original + Global กับ Original + LoBIA และ Original + LoBIAGlobal นั้นให้ผลลัพธ์ในการจัดกลุ่มได้ที่ใกล้เคียงกัน แต่จำนวนภาพที่นำมาฝึกสอนนั้นกลุ่มข้อมูลภาพ Original + LoBIA นั้นมีจำนวนที่น้อยกว่ากลุ่มทั้งสองค่อนข้างมาก และเมื่อนำผลการจัดกลุ่มที่ได้มาคำนวณค่าเฉลี่ยของความถูกต้องในการจัดกลุ่มแล้ว ได้ผลค่าเฉลี่ยแสดงตารางที่ 4.2 (ไบโองุ่น) ตารางที่ 4.3 (ใบมันฝรั่ง) และตารางที่ 4.4 ตารางผลทดสอบของแต่ละชนิดของชุดข้อมูลฝึกสอน

จากผลการทดลองทั้งหมดของโมเดลที่ได้นำชุดข้อมูลฝึกสอนที่เตรียมไว้ทั้ง 4 กลุ่มชุดข้อมูลมาฝึกสอนนั้นจะเห็นได้ว่ากราฟในรูปที่ 4.18, 4.19, 4.33, 4.35, 4.49, 4.51, 4.65, 4.67 กลุ่มของข้อมูลฝึกสอนต้นฉบับ (Original) ให้ผลการจัดกลุ่มโรคได้ไม่ค่อยดีโดยที่ยังมีค่าผิดพลาดที่ค่อนข้างสูง อาจเกิดมาจากจำนวนของข้อมูลที่ไม่มีความหลากหลายและน้อยเกินไป แต่กราฟในรูปที่ 4.22, 4.23, 4.37, 4.39, 4.53, 4.55, 4.69, 4.71 กลุ่มของข้อมูลฝึกสอน Original + Global และกราฟในรูปที่ 4.26, 4.27, 4.41, 4.43, 4.57, 4.59, 4.73, 4.75 ของกลุ่มข้อมูลฝึกสอน Original + LoBIA ให้ผลลัพธ์ที่ค่อนข้างใกล้เคียงกัน แต่ในทางจำนวนฝึกสอนแล้วนั้น Original + LoBIA นั้นมีจำนวนที่น้อยกว่ามาก และกราฟในรูปที่ 4.30, 4.31, 4.45, 4.47, 4.61, 4.63, 4.77, 4.79 กลุ่มข้อมูลฝึกสอน Original + LoBIAGlobal นั้นให้ผลลัพธ์ของการจำแนกข้อมูลได้ค่อนข้างสูงและมีค่าผิดพลาดที่น้อยกว่ากลุ่มข้อมูลทั้งสาม แต่อาจเนื่องมาจากว่ามีข้อมูลภาพฝึกสอนที่หลากหลายและจำนวนข้อมูลภาพที่มากกว่ากลุ่มข้อมูลอื่น ๆ

ตารางที่ 4.2 ตารางผลทดสอบการจำแนกชนิดโรคของใบองุ่น

		จำนวน	CNN_7layers	CNN_9layers	MobileNet	VGG19
Original	Black_rot	198	70.00	67.50	<b>97.50</b>	85.00
	Black_Measles	200	82.50	85.00	<b>90.00</b>	<b>95.00</b>
	Leaf_blight	197	<b>97.50</b>	<b>92.50</b>	22.50	<b>100.00</b>
	Healthy	200	<b>92.50</b>	<b>95.00</b>	<b>100.00</b>	<b>97.50</b>
Global	Black_rot	1,782	67.50	65.00	<b>100.00</b>	<b>92.50</b>
	Black_Measles	1,800	<b>92.50</b>	85.00	75.00	<b>97.50</b>
	Leaf_blight	1,773	<b>100.00</b>	<b>100.00</b>	20.00	<b>100.00</b>
	Healthy	1,800	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
LoBIA	Black_rot	841	77.50	77.50	<b>100.00</b>	<b>90.00</b>
	Black_Measles	570	<b>90.00</b>	<b>95.00</b>	87.50	<b>95.00</b>
	Leaf_blight	832	<b>97.50</b>	<b>97.50</b>	50.00	<b>100.00</b>
	Healthy	200	<b>92.50</b>	85.00	85.00	<b>97.50</b>
LoBIAGlobal	Black_rot	5,342	75.00	77.50	<b>100.00</b>	<b>100.00</b>
	Black_Measles	3,160	<b>95.00</b>	87.50	82.50	<b>95.00</b>
	Leaf_blight	5,277	<b>97.50</b>	<b>100.00</b>	67.50	<b>100.00</b>
	Healthy	1,800	<b>100.00</b>	<b>90.00</b>	<b>100.00</b>	<b>100.00</b>

จากตารางที่ 4.2 เป็นตารางแสดงผลค่าเฉลี่ยจากการคำนวณค่าความถูกต้องในการจัดกลุ่มข้อมูลของโรคบนใบองุ่นในแต่ละกลุ่มข้อมูลที่นำมาฝึกสอนเพื่อนำมาแสดงให้เห็นว่าโมเดลที่ถูกฝึกสอนด้วยกลุ่มฝึกสอนแต่ละกลุ่มสามารถจำแนกชนิดของโรคได้กี่เปอร์เซ็นต์ในของแต่ละชนิดโรค โดยจะเห็นว่าผลลัพธ์ในกลุ่มข้อมูลชุดฝึกสอน Original + Global นั้นให้ผลในการจำแนกข้อมูลได้ค่อนข้างดีในบางกลุ่มโรคเท่านั้น โดยเมื่อนำข้อมูลผลลัพธ์ที่ได้นั้นมาทำการเปรียบเทียบกับผลลัพธ์ที่ได้จากกลุ่มข้อมูลชุดฝึกสอน Original + LoBIA และ Original + LoBIAGlobal โดยทั้งสองกลุ่มนี้ให้ผลเฉลี่ยในการจำแนกข้อมูลที่มีความใกล้เคียงกันในแต่ละโรค

ตารางที่ 4.3 ตารางผลทดสอบการจำแนกชนิดโรคของใบมันฝรั่ง

		จำนวน	CNN_7layers	CNN_9layers	MobileNet	VGG19
Original	Early_blight	150	87.50	77.50	<b>92.50</b>	<b>97.50</b>
	Late_blight	150	<b>92.50</b>	<b>97.50</b>	20.00	<b>100.00</b>
	Healthy	150	<b>100.00</b>	<b>97.50</b>	<b>100.00</b>	<b>90.00</b>
Global	Early_blight	1,350	<b>97.50</b>	<b>95.00</b>	<b>95.00</b>	<b>95.00</b>
	Late_blight	1,350	<b>97.50</b>	<b>97.50</b>	35.00	<b>95.00</b>
	Healthy	1,350	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>
LoBIA	Early_blight	405	<b>97.50</b>	<b>97.50</b>	<b>97.50</b>	<b>100.00</b>
	Late_blight	440	<b>97.50</b>	<b>97.50</b>	70.00	<b>97.50</b>
	Healthy	150	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>92.50</b>
LoBIAGlobal	Early_blight	2,190	<b>92.50</b>	<b>97.500</b>	<b>97.50</b>	<b>97.50</b>
	Late_blight	2,470	<b>95.00</b>	<b>97.500</b>	70.00	87.50
	Healthy	1,350	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>	<b>100.00</b>

จากตารางที่ 4.3 เป็นตารางแสดงผลค่าเฉลี่ยจากการคำนวณค่าความถูกต้องในการจัดกลุ่มข้อมูลของโรคบนใบมันฝรั่งในแต่ละกลุ่มข้อมูลที่นำมาฝึกสอนเพื่อนำมาแสดงให้เห็นว่าโมเดลที่ถูกฝึกสอนด้วยกลุ่มฝึกสอนแต่ละกลุ่มสามารถจำแนกชนิดของโรคได้ก็เปอร์เซ็นต์ในของแต่ละชนิดโรค โดยจะเห็นได้ว่าผลลัพธ์ในกลุ่มข้อมูลชุดฝึกสอน Original + Global นั้นให้ผลในการจำแนกข้อมูลได้ค่อนข้างดีในบางกลุ่มโรคเท่านั้น โดยเมื่อนำข้อมูลผลลัพธ์ที่ได้นั้นมาทำการเปรียบเทียบกับผลลัพธ์ที่ได้จากกลุ่มข้อมูลชุดฝึกสอน Original + LoBIA และ Original + LoBIAGlobal โดยทั้งสองกลุ่มนี้ให้ผลเฉลี่ยในการจำแนกข้อมูลที่มีความใกล้เคียงกันในแต่ละโรค

ตารางที่ 4.4 ตารางผลทดสอบของแต่ละชนิดของชุดข้อมูลฝึกสอน

		จำนวน	CNN_7layers	CNN_9layers	MobileNet	VGG19
Potato	Original	450	93.33	90.83	70.83	95.83
	Global	4,050	<b>98.33</b>	97.50	76.67	<b>96.67</b>
	LoBIA	995	<b>98.33</b>	<b>98.33</b>	<b>89.17</b>	<b>96.67</b>
	LoBIAGlobal	6,010	95.83	<b>98.33</b>	<b>89.17</b>	95.00
Grape	Original	795	85.63	85.00	77.50	94.38
	Global	7,155	90.00	87.50	73.75	97.50
	LoBIA	2,443	89.38	<b>88.75</b>	80.63	95.63
	LoBIAGlobal	15,579	<b>91.88</b>	<b>88.75</b>	<b>87.50</b>	<b>98.75</b>

จากตารางที่ 4.4 จะเห็นได้ว่าค่าเฉลี่ยของการจำแนกโรคของแต่ละโมเดลนั้นให้ผลที่ออกมาค่อนข้างดีและเป็นไปตามหลักการตามที่ต้องการของงานวิจัยโดยชุดข้อมูลของใบมันฝรั่งสามารถจำแนกประเภทของโรคได้ดีกับโมเดลที่ฝึกสอนด้วยชุดฝึกสอน Original + LoBIA กับทุกโมเดล และกับใบองุ่น สามารถจำแนกประเภทของโรคได้ดีกับโมเดลที่ฝึกสอนด้วยชุดฝึกสอน Original + LoBIAGlobal กับทุกโมเดล

#### 4.5 ผลที่เกิดจากการใช้ Local-based image augmentation ที่ทำให้โมเดลมีความสามารถในการจำแนกดีขึ้น



รูปที่ 4.81 ภาพตัวอย่างโรค Black measles ที่ทำนายผิด

จากรูปที่ 4.81 เป็นรูปของโรค Black measles ของใบองุ่นซึ่งเป็นโรคที่นำมาใช้ทดสอบการทำนายของโมเดลทั้ง 4 และปรากฏว่าเกือบทุกโมเดลที่ฝึกสอนโดยชุดข้อมูลฝึกสอน



Original+Global นั้นไม่สามารถที่จะทำนายชนิดของโรคจากภาพทดสอบนี้ได้ถูกต้อง โดยได้ถูกทำนายเป็นโรคชนิด Black rot และ Leaf blight ตามลำดับ แต่เมื่อฝึกสอนด้วยชุดฝึกสอน Original+LoBIA แล้วนั้น โมเดลก็สามารถที่จะทำนายโรคจากภาพทดสอบนี้ได้



รูปที่ 4.82 ภาพตัวอย่างโรค Black rot ที่ทำนายผิด

จากรูปที่ 4.82 เป็นรูปของโรค Black rot ของใบองุ่นซึ่งเป็นโรคที่นำมาใช้ทดสอบการทำนายของโมเดลทั้ง 4 และปรากฏว่าเกือบทุกโมเดลที่ฝึกสอนโดยชุดข้อมูลฝึกสอน Original+Global นั้นไม่สามารถที่จะทำนายชนิดของโรคจากภาพทดสอบนี้ได้ถูกต้อง โดยได้ถูกทำนายเป็นโรคชนิด Leaf blight และ Black measles ตามลำดับ แต่เมื่อฝึกสอนด้วยชุดฝึกสอน Original+LoBIA แล้วนั้น โมเดลก็สามารถที่จะทำนายโรคจากภาพทดสอบนี้ได้



รูปที่ 4.83 ภาพตัวอย่างโรค Leaf blight ที่ทำนายผิด

จากรูปที่ 4.83 เป็นรูปของโรค Leaf blight ของใบองุ่นซึ่งเป็นโรคที่นำมาใช้ทดสอบการทำนายของโมเดลทั้ง 4 และปรากฏว่าเกือบทุกโมเดลที่ฝึกสอนโดยชุดข้อมูลฝึกสอน Original+Global นั้นไม่สามารถที่จะทำนายชนิดของโรคจากภาพทดสอบนี้ได้ถูกต้อง โดยได้ถูกทำนายเป็นโรคชนิด Black rot แต่เมื่อฝึกสอนด้วยชุดฝึกสอน Original+LoBIA แล้วนั้น โมเดลก็สามารถที่จะทำนายโรคจากภาพทดสอบนี้ได้



รูปที่ 4.84 ภาพตัวอย่างโรค Early blight ที่ทำนายผิด

จากรูปที่ 4.84 เป็นรูปของโรค Early blight ของใบมันฝรั่งซึ่งเป็นโรคที่นำมาใช้ทดสอบการทำนายของโมเดลทั้ง 4 และปรากฏว่าเกือบทุกโมเดลที่ฝึกสอนโดยชุดข้อมูลฝึกสอน Original+Global นั้นไม่สามารถที่จะทำนายชนิดของโรคจากภาพทดสอบนี้ได้ถูกต้อง โดยได้ถูกทำนายเป็นโรคชนิด Late Blight แต่เมื่อฝึกสอนด้วยชุดฝึกสอน Original+LoBIA แล้วนั้น โมเดลก็สามารถที่จะทำนายโรคจากภาพทดสอบนี้ได้



รูปที่ 4.85 ภาพตัวอย่างโรค Late blight ที่ทำนายผิด

จากรูปที่ 4.85 เป็นรูปของโรค Late blight ของใบมันฝรั่งซึ่งเป็นโรคที่นำมาใช้ทดสอบการทำนายของโมเดลทั้ง 4 และปรากฏว่าเกือบทุกโมเดลที่ฝึกสอนโดยชุดข้อมูลฝึกสอน Original+Global นั้นไม่สามารถที่จะทำนายชนิดของโรคจากภาพทดสอบนี้ได้ถูกต้อง โดยได้ถูกทำนายเป็นโรคชนิด Healthy แต่เมื่อฝึกสอนด้วยชุดฝึกสอน Original+LoBIA แล้วนั้น โมเดลก็สามารถที่จะทำนายโรคจากภาพทดสอบนี้ได้



## บทที่ 5

### สรุปและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

วิทยานิพนธ์นี้นำเสนอการพัฒนาโปรแกรมการเรียนรู้เชิงลึกสำหรับจำแนกชนิดของโรคพืช มีวัตถุประสงค์เพื่อทำการศึกษาระเบียบวิธีการออกแบบโครงสร้างของโมเดลที่นำมาใช้ในการแบ่งส่วนและจำแนกชนิดของโรค โดยการนำความรู้การทำงานของอัลกอริทึมต่าง ๆ ทางด้านการประมวลผลภาพมาช่วยเพื่อให้สามารถจำแนกโรคได้แม่นยำยิ่งขึ้น และสามารถทำได้เป็นจำนวนมากแทนการวิเคราะห์ด้วยตาของผู้สังเกต ซึ่งในการออกแบบระบบการจำแนกชนิดของโรคนั้น จะทำการแบ่งชนิดและบ่งบอกชนิดของโรคจากลักษณะเด่นของชนิดโรคนั้น ๆ เพื่อช่วยในการพยากรณ์โรคที่เกิดขึ้นให้กับงานทางการเกษตรเพื่อให้สามารถแก้ไขได้ตรงจุด ในงานวิจัยได้ทำการทดลองกับใบพืช 2 ชนิด ได้แก่ 1. ใบองุ่น ประกอบไปด้วยกลุ่มโรค 4 กลุ่มโรค คือ Black rot, Black measles, Leaf blight และ Healthy 2. ใบมันฝรั่ง ประกอบไปด้วยกลุ่มโรค 3 กลุ่มโรค คือ Early blight, Late blight และ Healthy

ในวิทยานิพนธ์นี้ได้มีการแบ่งระเบียบวิธีการทำงานของโปรแกรมออกเป็น 2 ส่วน ได้แก่ 1. ส่วนของการจัดกลุ่มและแยกส่วนบริเวณที่เป็นโรคและใบ 2. ส่วนของการวิเคราะห์จำแนกชนิดของโรคพืช

ในส่วนของการออกแบบส่วนแรก การออกแบบการจัดกลุ่มและแยกส่วนบริเวณที่เป็นโรคและใบ โดยในส่วนนี้เป็นส่วนที่นำเอาเทคนิควิธีการจัดกลุ่มของฟิกเซลสี SLIC superpixel มาใช้แบ่งส่วนบริเวณโรคกับใบอย่างละ 300 ภาพ เพื่อนำมาทำเป็นชุดข้อมูลฝึกสอนให้กับโมเดล CNN เพื่อให้สามารถแบ่งส่วนที่เป็นโรคได้แม่นยำยิ่งขึ้น ได้ทำการทดลองกับภาพทดสอบที่จะนำไปใช้ในขั้นตอนต่อไปโดยทำการปรับเปลี่ยนจำนวนของซูเปอร์ฟิกเซล  $K$  ให้มีจำนวนที่แตกต่างกัน  $K = 500, 600, 700, 800$  เพื่อศึกษาความแม่นยำในการแบ่งส่วนบริเวณโรค จากการทดสอบได้ว่าเมื่อทำการปรับค่า  $K = 800$  ก็จะทำให้จำนวนของซูเปอร์ฟิกเซลมีจำนวนมากขึ้นและทำให้โมเดลสามารถจำแนกส่วนบริเวณที่เป็นโรคได้ถูกต้องโดยให้ค่าความถูกต้องสูงสุดที่ 94.20%

ในส่วนที่สองเป็นส่วนของการสร้างโมเดลสำหรับวิเคราะห์จำแนกชนิดของโรค ในส่วนนี้ได้ใช้ผลลัพธ์ของการแบ่งส่วนบริเวณโรคจากขั้นตอนแรกมาใช้โดยเลือกบริเวณพื้นที่ที่เป็นโรคจำนวน 3 พื้นที่ที่ใหญ่ที่สุดมาใช้ โดยจะทำการนำเอาพื้นที่บริเวณโรคที่เลือกมาใช้กระจายตัวบนใบไม้เพื่อให้เกิดความหลากหลายของข้อมูลแบบเชิงพื้นที่และนำข้อมูลที่ได้ไปใช้ฝึกสอนโมเดลทั้ง 4 โมเดล ในการ

ฝึกสอนโมเดลได้จัดกลุ่มชุดข้อมูลฝึกสอนออกเป็น 4 ชุดข้อมูลฝึกสอน ได้แก่ 1. Original 2. Original + Global 3. Original + LoBIA และ 4. Original + LoBIAGlobal เพื่อนำผลลัพธ์ที่ได้มาเปรียบเทียบค่าความแม่นยำในการจำแนกชนิดของโรคกับชุดภาพทดสอบภาพใบองุ่นจำนวน 160 ภาพ และภาพใบมันฝรั่งจำนวน 120 ภาพ จากผลการทดลองเห็นได้ว่า ผลของการจำแนกจากชุดข้อมูลของ Original + LoBIAGlobal กับทุกโมเดลให้ผลลัพธ์ของการจำแนกเฉลี่ยที่ค่อนข้างสูง และมีความถูกต้องที่ใกล้เคียงกันในทุกชนิดของโรค อาจเนื่องมาจากเป็นกลุ่มข้อมูลที่มีจำนวนภาพฝึกสอนที่เยอะที่สุดของทั้งสี่กลุ่ม โดยมีจำนวน 6,010 ภาพของใบมันฝรั่ง และ 15,579 ภาพของใบองุ่น จึงทำให้มีความหลากหลายของข้อมูลที่มากกว่า และในผลของการจำแนกจากชุดข้อมูลของ Original + Global ให้ผลลัพธ์ที่สูงเช่นกันด้วยจำนวนภาพฝึกสอน 4,050 ภาพของใบมันฝรั่ง และ 7,155 ภาพของใบองุ่น แต่อาจมีบางชนิดของโรคในทุกโมเดลที่ให้ผลของการจำแนกที่น้อยกว่าโรคอื่น และในผลของการจำแนกจากชุดข้อมูลของ Original + LoBIA ให้ผลลัพธ์ที่ค่อนข้างใกล้เคียงกับ ชุดข้อมูลของ Original + Global แต่ด้วยจำนวนภาพฝึกสอนที่น้อยกว่าที่ 995 ภาพของใบมันฝรั่ง และ 2,443 ภาพของใบองุ่น ทำให้ใช้เวลาในการฝึกสอนโมเดลน้อยกว่าชุดข้อมูลแบบ Original + LoBIAGlobal และ Original + Global ค่อนข้างมากและสูญเสียทรัพยากรของเครื่องที่ไม่จำเป็นน้อยกว่าชุดข้อมูลฝึกสอนข้างต้น ซึ่งเป็นวิธีการที่นำเสนอในวิทยานิพนธ์นี้

## 5.2 ข้อเสนอแนะ

1. โปรแกรมจำแนกชนิดของโรคพืชที่ออกแบบนี้เป็นเพียงโปรแกรมต้นแบบของการจำแนกโรคพืช เนื่องจากมีทรัพยากรเครื่องที่จำกัดจึงไม่สามารถนำข้อมูลจำนวนมากมาใช้งานได้
2. รูปแบบของการเพิ่มข้อมูลภาพฝึกสอนแบบเชิงพื้นที่ที่นำเสนอ สามารถนำไปประยุกต์ใช้ได้กับโมเดลการเรียนรู้เชิงลึกโมเดลอื่น ๆ ได้
3. ข้อจำกัดในการทำ Local-based image augmentation นั้นจะต้องคำนึงถึงหลักความเป็นจริงของการเกิดโรคบนใบไม้ โดยในการปรับค่าเปอร์เซ็นต์ของการกระจายตัวของข้อมูลลงไปเพิ่มนั้นจึงไม่อาจปรับค่าให้สูงขึ้นในปริมาณมาก ๆ ได้ เนื่องจากอาจจะเป็นการเพิ่มบริเวณข้อมูลของการเกิดโรคให้มีพื้นที่มากเกินไปโดยในความเป็นจริงอาจเกิดขึ้นได้ยากหรืออาจจะเกิดการจำแนกที่ผิดพลาดกลายเป็นโรคอื่น

## 5.3 แนวทางการพัฒนาต่อยอด

1. ต่อยอดทำการฝึกสอนโมเดลกับข้อมูลฝึกสอนที่มากขึ้นให้สามารถจำแนกโรคได้หลากหลายชนิดมาก

2. พัฒนาต่อยอดโปรแกรมให้สามารถจำแนกโรคพืชได้หลายชนิดโรคในเวลาเดียวกัน หรือสามารถวิเคราะห์โรคพืชแบบ Real time ด้วยกล้องวิดีโอผ่านโทรศัพท์มือถือ
3. นำไปต่อยอดเพื่อให้สามารถใช้งานได้ในแอปพลิเคชันบน Smart Phone



## รายการอ้างอิง

1. รศ.ดร.ปริญญา สงวนวสีตย์, *Artificial Intelligence with Machine Learning, AI สร้างได้ด้วยแมชชีนเลิร์นนิง*. Python ed. Vol. 1. 2562, นนทบุรี: ไอทีซี พรีเมียร์. 356.
2. อรพิน ประวัตติบริสุทธิ์, *Python สำหรับ Data Science Data Visualization และ Machine Learning*. 2564, กรุงเทพฯ: โปรวิชั่น. 400.
3. รศ.ดร.จตุรภัทร วาฤทธิ, *องค์ความรู้ระบบ MJU Smart Farm and Solutions*. 2561, เชียงใหม่: หจก.วนิดาการพิมพ์. 192.
4. Agarwal, M., S.K. Gupta, and K.K. Biswas, *Grape Disease Identification Using Convolution Neural Network*, in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*. 2019, IEEE: Phuket, Thailand. p. 224-229.
5. Kumar, A. and M. Vani, *Image Based Tomato Leaf Disease Detection*, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2019, IEEE: Kanpur, India. p. 1-6.
6. Beron, E.P., A.F.C. Salcedo, and A. Bejarano, *Detection of foliar diseases using image processing techniques*. *Revista Ceres*, 2020. 67(2): p. 100-110.
7. Zhou, G., et al., *Rapid Detection of Rice Disease Based on FCM-KM and Faster R-CNN Fusion*. *IEEE Access*, 2019. 7: p. 143190-143206.
8. Arya, S. and R. Singh, *A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf*, in *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. 2019, IEEE: Ghaziabad, India. p. 1-6.
9. Widiyanto, S., R. Fitrianto, and D.T. Wardani, *Implementation of Convolutional Neural Network Method for Classification of Diseases in Tomato Leaves*, in *2019 Fourth International Conference on Informatics and Computing (ICIC)*. 2019, IEEE: Semarang, Indonesia. p. 1-5.
10. Deepak, A.H., et al., *Disease Detection in Tomato plants and Remote Monitoring of agricultural parameters*, in *2019 11th International Conference on Advanced Computing (ICoAC)*. 2019, IEEE: Chennai, India. p. 28-33.
11. Hasan, M.Z., et al., *Recognition of Jute Diseases by Leaf Image Classification*

- using Convolutional Neural Network, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2019, IEEE: Kanpur, India. p. 1-5.
12. Jiang, P., et al., *Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks*. *IEEE Access*, 2019. 7: p. 59069-59080.
  13. Sheikh, M.H., et al., *Detection of Maize and Peach Leaf diseases using Image Processing*, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2019, IEEE: Kanpur, India. p. 1-7.
  14. Yang, M., Z. Li, and J. Li, *Super-pixel based single image haze removal*, in *Chinese Control and Decision Conference (CCDC)*. 2016, IEEE: Yinchuan, China. p. 1965-1969.
  15. Zhang, S., Z. You, and X. Wu, *Plant disease leaf image segmentation based on superpixel clustering and EM algorithm*. *Neural Computing and Applications*, 2017. 31: p. 1225-1232.
  16. Bhattarai, S., *New Plant Diseases Dataset*. 2019.
  17. Liu, D., *Plant Diseases(6:2:2)*. 2020.
  18. Khirade, S.D. and A.B. Patil. *Plant Disease Detection Using Image Processing*. in *International Conference on Computing Communication Control and Automation*. 2015. Pune, India: IEEE.
  19. Gonzales, R.C., R.E. Woods, and S.L. Eddins, *Digital Image Processing Using MATLAB*. 2 ed. 2011: McGraw-Hill Education (Asia). 756.
  20. Kurtz, O., et al., *Innovative Tin Electrolyte combining high technical standards with significant cost saving potentials*, in *2012 35th IEEE/CPMT International Electronics Manufacturing Technology Conference (IEMT)*. 2012, IEEE: Ipoh, Malaysia. p. 1-15.
  21. HunterLab, *CIE L\*a\*b\* Color Scale*. 1996, Hunter Associates Laboratory, Inc.: Virginia. p. 4.
  22. Achanta, R., et al., *SLIC Superpixels*. 2010, School of Computer and Communication Sciences (IC): *École Polytechnique Fédérale de Lausanne (EPFL)*.

- p. 15.
23. Achanta, R., et al., *SLIC Superpixels Compared to State-of-the-art Superpixel Methods*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 34(11): p. 2274-2282.
  24. Huiying, D. and J. Tengguang, *Rapid detection of obstacle based on super pixels*, in *The 27th Chinese Control and Decision Conference (2015 CCDC)*. 2015, IEEE: Qingdao, China. p. 4993-4995.
  25. ดร.สรรพฤทธิ์ มฤคมัต, การรู้จำรูปแบบและการเรียนรู้ของเครื่อง *Pattern Recognition and Machine Learning*. Vol. 2. 2562, ปทุมธานี: สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ. 206.
  26. Zejmo, M., et al., *Nuclei Recognition Using Convolutional Neural Network and Hough Transform*. *Advanced Solutions in Diagnostics and Fault Tolerant Control* ed. *I.C.o.D.o.P.a. Systems*. Vol. 635. 2018, Advances in Intelligent Systems and Computing: Springer, Cham.
  27. Elhassouny, A. and F. Smarandache, *Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks*, in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*. 2019, IEEE: Agadir, Morocco. p. 1-4.
  28. Sardogan, M., A. Tuncer, and Y. Ozen, *Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm*, in *2018 3rd International Conference on Computer Science and Engineering (UBMK)*. 2018, IEEE: Sarajevo, Bosnia and Herzegovina. p. 382-385.
  29. Sharma, P., P. Hans, and S.C. Gupta, *Classification of Plant Leaf Diseases Using Machine Learning and Image Preprocessing Techniques*, in *International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. 2020, IEEE: Noida, India. p. 480-484.
  30. Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*. *CoRR*, 2015. abs/1409.1556: p. 1-14.
  31. Agarwal, M., et al., *FCNN-LDA: A Faster Convolution Neural Network model for Leaf Disease identification on Apple's leaf dataset*, in *2019 12th International Conference on Information & Communication Technology and System (ICTS)*.



- 2019, IEEE: Surabaya, Indonesia. p. 246-251.
32. Bhatt, P., S. Sarangi, and S. Pappula, *Comparison of CNN Models for Application in Crop Health Assessment with Participatory Sensing*, in *IEEE Global Humanitarian Technology Conference (GHTC)*. 2017, IEEE: San Jose, CA, USA. p. 1-7.
  33. Suryawati, E., et al., *Deep Structured Convolutional Neural Network for Tomato Diseases Detection*, in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. 2018, IEEE: Yogyakarta, Indonesia. p. 385-390.
  34. Zheng, Z., S. Pan, and Y. Zhang, *Fruit Tree Disease Recognition Based on Convolutional Neural Networks*, in *2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS)*. 2019, IEEE: Shenyang, China. p. 118-122.
  35. Ferguson, M., et al., *Automatic localization of casting defects with convolutional neural networks*, in *2017 IEEE International Conference on Big Data (Big Data)*. 2017, IEEE: Boston, MA, USA. p. 1726-1735.
  36. Combinido, J.S., J.R. Mendoza, and J. Aborot, *A Convolutional Neural Network Approach for Estimating Tropical Cyclone Intensity Using Satellite-based Infrared Images*, in *2018 24th International Conference on Pattern Recognition (ICPR)*. 2018, IEEE: Beijing, China. p. 1474-1480.
  37. G.Suresh, V.Gnanaprakash, and R. Santhiya, *Performance Analysis of Different CNN Architecture with Different Optimisers for Plant Disease Classification*, in *International Conference on Advanced Computing & Communication Systems (ICACCS)*. 2019, IEEE: Coimbatore, India. p. 916-921.
  38. Howard, A.G., et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. *ArXiv*, 2017. abs/1704.04861: p. 1-9.
  39. Kamrul, M.H., P. Paul, and M. Rahman, *Machine Vision Based Rice Disease Recognition by Deep Learning*, in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*. 2019, IEEE: Dhaka, Bangladesh. p. 1-6.



40. Deepak, A.H., et al., *Disease Detection in Tomato plants and Remote Monitoring of agricultural parameters*, in *International Conference on Advanced Computing (ICoAC)*. 2020, IEEE: Chennai, India. p. 28-33.
41. Hasan, M.J., et al., *Rice Disease Identification and Classification by Integrating Support Vector Machine with Deep Convolutional Neural Network*, in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. 2019, IEEE: Dhaka, Bangladesh. p. 1-6.
42. Howlader, M.R., et al., *Automatic Recognition of Guava Leaf Diseases using Deep Convolution Neural Network*, in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. 2019, IEEE: Cox'sBazar, Bangladesh. p. 1-5.
43. Khirade;, S.D. and A.B. Patil, *Plant Disease Detection Using Image Processing*, in *2015 International Conference on Computing Communication Control and Automation*. 2015, IEEE: Pune, India. p. 768-771.





ภาคผนวก



ภาคผนวก ก  
ผลงานที่ได้รับการตีพิมพ์



The 2021 18th International Conference on Electrical Engineering/Electronics,  
Computer, Telecommunications and Information Technology

On May 19-22, 2021 (Virtual Conference Platform)



# Certificate of Attendance

for the paper

“Plant Leaf Disease Classification using Local-Based Image  
Augmentation and Convolutional Neural Network”

authored by

Jirasak Wongbongkotpaisan and Sophon Phumeechanya

(Assoc. Prof. Dr. Yuttana Kumsuwan)  
General Chair

(Assoc. Prof. Dr. Nipon Theera-Umpon)  
Technical Program Chair



# ECTI-CON 2021 | Chiang Mai

2021 18th International Conference on Electrical Engineering/Electronics,  
Computer, Telecommunications and Information Technology

## Smart Electrical Systems & Technology

**MAY 19-22, 2021**  
Virtual Conference Hosted by  
Department of Electrical Engineering,  
Faculty of Engineering, Chiang Mai University

*Proceedings & Program Book*



ECTI Association        ECTI-CON COOPERATION

SPONSORS & EXHIBITORS    



# Plant Leaf Disease Classification using Local-Based Image Augmentation and Convolutional Neural Network

Jirasak Wongbongkotpaisan  
 Department of Electrical Engineering  
 Faculty of Engineering and Industrial Technology  
 Silpakorn University  
 Nakorn Pathom, Thailand  
 wongbongkot\_j@su.ac.th

Sopon Phumeechanya  
 Department of Electrical Engineering  
 Faculty of Engineering and Industrial Technology  
 Silpakorn University  
 Nakorn Pathom, Thailand  
 phumeechanya\_s@su.ac.th

**Abstract**— A local-based image augmentation method to cooperate with convolutional neural networks for plant leaf disease classification is proposed. There are three processes of the diseased classification such as: 1) the local-based image augmentation method, 2) the global-based image augmentation methods and 3) convolutional neural network (CNN) models. First process, the local-based image augmentation method is designed to distribute disease regions on the plant leaves for increasing the number of images in the datasets. The proposed method focuses on only the regions on the leaves which is called the local-based approach. Second process, the general global-based image augmentation methods are used to increase variety of the images regarding pose, brightness, blur and noise. Third process, the simple CNN model is proposed for plant leaf disease classification using the augmented images. The CNN model is proposed in this research. The VGG19 and MobileNet are using for comparison and study the impact of the proposed method in the first process including. We test on two kinds of plants: potato and grape with three CNN models: our CNN model, VGG19 and MobileNet. The accuracy rate of potato and grape leaf for our CNN, VGG19 and MobileNet model are 93.78%, 95.56%, 77.33%, 93.90%, 96.30%, and 80.00%, respectively. From the result can be concluded, that the proposed local-based image augmentation method can increase the accuracy rates of all models.

**Keywords**— *image augmentation, convolutional neural network, plant leaf disease classification, deep learning, image processing*

## I. INTRODUCTION

Image dataset is essential for classification task with convolutional neural networks in order to train and test the models [1,2]. If there is a lot of image in the dataset, it could improve the performance of the models and avoid overfitting. In case of image lacking, image augmentation techniques are employed to add the number of images in the datasets [3,4].

Previously several researches, image augmentation techniques are widely used for plant leaf disease classification. The original images in the datasets are variously augmented using different techniques including image enhancement, image filtering, geometric transformation, and adding noise [5-8]. However, these techniques perform globally with the overall region of the image. They do not focus only on the region of plant leaf in the image. These techniques are called the global-based image augmentation methods. Besides, it is

not covered in the creation of a variety of disease distribution on the leaves.

We present a local-based image augmentation method to multiply the number of images in the datasets. First, Our method acts to spread the disease throughout the leaves of the plant. Second, the locally augmented images become the inputs of the global-based image augmentation step. Then, we offer a simple CNN model for the classification of plant diseases on the leaves. Our local-based image augmentation method is tested on two plant datasets of potato and grape. Finally, the impact of our local-based method is studied. The augmented images are used for training the three models via the CNN model, VGG19 [9], and MobileNet [10].

## II. RELATED WORK

Image augmentation is widely used to be a pre-processing step for plant disease classification in case of a shortage of images, which are not enough to train models. It also contributes to the relief of overfitting.

There are many works, that take into account the image augmentation for increasing the number of images in the datasets. For instance, Jiang et al. [3] proposed an approach for plant disease classification on apple leaves using VGG-INCEP model with 12 types of image augmentation regarding brightness, contrast, flip, sharpness, Gaussian, and rotation. Hasan et al. [11] applied convolutional neural network for jute disease classification. The original images were augmented in 6 styles by flipping, rotating, and scaling. Widiyanto et al. [4] used only one technique of image augmentation for tomato leaf disease classification. The original images were rotated both right and left sides to 20 degrees. Sheikh et al. [12] applied rotating, shifting, shearing, flipping and zooming in image augmentation step. This work classified the diseases on maize and peach leaves using convolutional neural network. Arya et al. [1] proposed the plant disease classification on potato and mango leaves using several image augmentation methods, affine transformation, perspective transformation and intensity transformation. The augmented images were used for training convolutional neural network and AlexNet models.

Regarding the above works, the image augmentation methods did not cover in spreading of diseases on the plant

leaves. These methods are performed globally with the overall region of the image. Hence, we called them the global-based image augmentation methods. In this paper, we propose a local-based image augmentation method to generate a variety of disease distribution on the plant leaves for improvement the efficiency of disease classification.

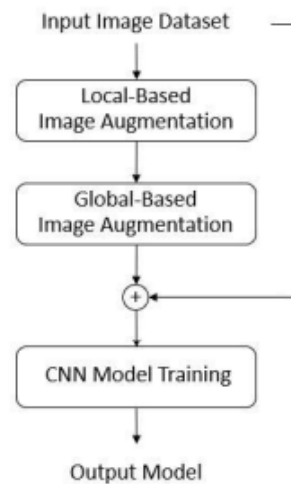


Fig. 1. Our proposed training framework.

### III. THE PROPOSED METHOD

The overall process of the proposed plant leaf disease classification is divided into three parts: A) local-based image augmentation, B) global-based image augmentation, and C) convolutional neural network model. Our proposed training framework is shown in Fig. 1. Our local-based image

augmentation method is designed to increase the number of images in the dataset by spreading disease regions on the leaf area. Then, the locally augmented images are forwarded to the global-based image augmentation step to multiply the number of images in a variety of pose, brightness, blur, and noise. Lastly, a simple CNN model is designed to classify plant leaf diseases using the augmented images in the process of training and testing.

#### A. Local-based image augmentation

Fig. 2 represents the overview of our local-based image augmentation method. The original RGB image is converted to  $L^*a^*b^*$  color space. Then, the  $L^*a^*b^*$  image is split into  $L^*$  channel,  $a^*$  channel, and  $b^*$  channel. The  $a^*$  channel is used to segment the disease regions on the leaf because this channel reacts to the color range of the diseases (brown tone). The  $b^*$  channel is used to find the leaf region in the image because this channel corresponds to the color range of the leaf (green tone). Next, the  $b^*$  channel is converted from grayscale to binary image by Otsu's thresholding and then all holes in the leaf region are filled. The AND operator is used to crop only the leaf region on the  $a^*$  channel. After that, the disease regions on the leaf region are detected by Otsu's thresholding. The biggest disease region (BDR) on the leaf region is selected to use in the disease distribution step. Meanwhile the disease distribution rate (%DDR) is calculated to be a criterion for spreading the disease region on the leaf. The %DDR is the ratio of all disease regions to the leaf region. Afterward, the AND operator is used to crop only the biggest disease region on the original RGB image. The final step is the disease distribution (DD). In this step, the biggest disease region is copied and pasted randomly on the leaf region of the original RGB image to create various patterns of disease distribution. We set the %DDR to 5%, 10%, 15%, and 20%, respectively, for creating the diversity of output images. The number of pasting is depended on the %DDR of the leaf so the number of output images might be less than or equal to four images. The examples of output images are illustrated in Fig. 3.

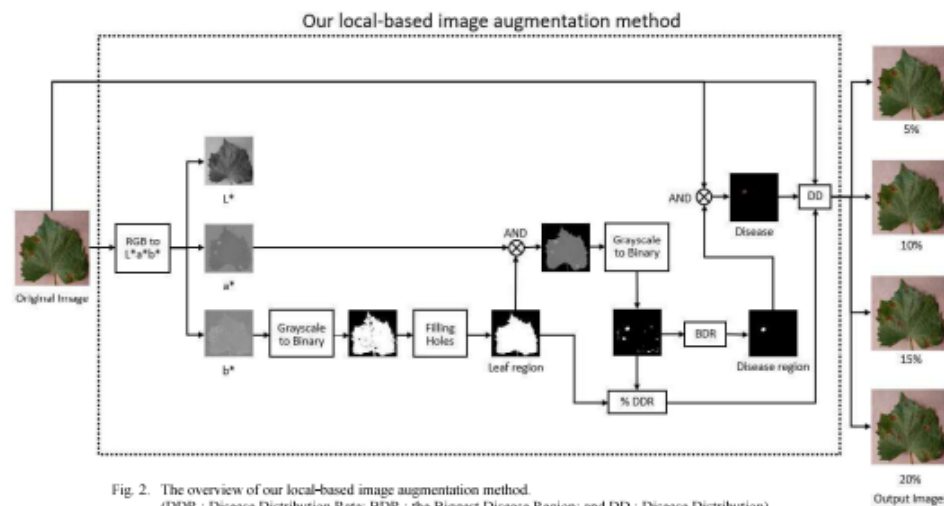


Fig. 2. The overview of our local-based image augmentation method. (DDR : Disease Distribution Rate; BDR : the Biggest Disease Region; and DD : Disease Distribution)



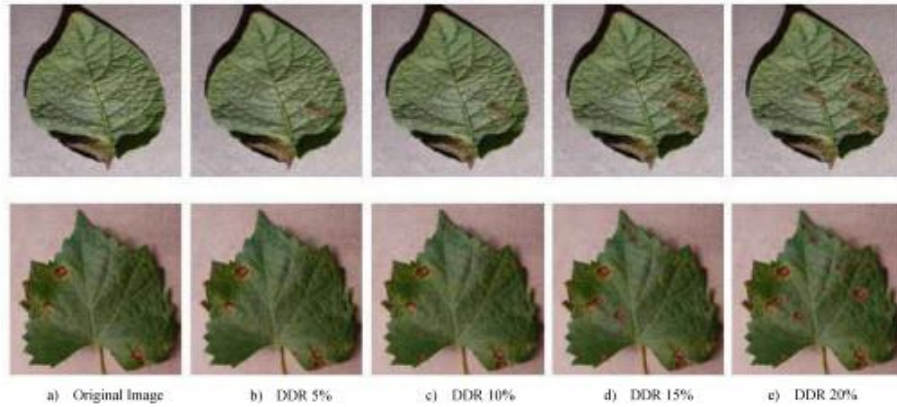


Fig. 3. The examples of output images from our local-based image augmentation step: a) original image of potato leaf (first row) and grape leaf (second row), b) DDR 5%, c) DDR 10%, d) DDR 15%, and e) DDR 20%.

**B. Global-based image augmentation**

In this section, the general global-based image augmentation methods are used to increase the variety of the images regarding pose, brightness, blur, and noise. The output images from the local-based image augmentation step are multiplied to six styles including horizontal flip, 45-degree rotation, -135-degree rotation, brightness reduction, blur, and Gaussian noise addition. The examples of the six output images are shown in Fig. 4.

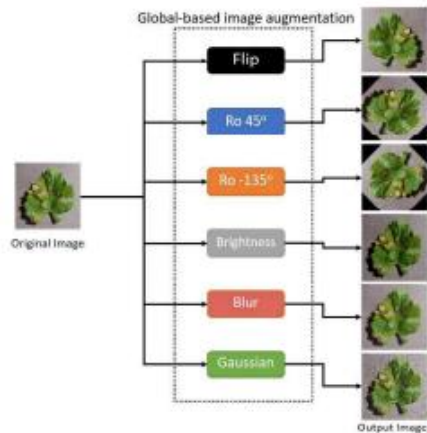


Fig. 4. The example of output images from the global-based image augmentation step.

**C. Convolutional neural network model**

To study the impact of our local-based image augmentation method, we propose a simple convolutional neural network (CNN) model for plant leaf disease classification as follows in Fig. 5. Our CNN structure is

included the convolution layers, ReLu, max pooling, flatten, and fully connected.



Fig. 5. Our convolutional neural network (CNN) model.

**IV. EXPERIMENTAL RESULTS**

In the experiments, we test the proposed method on two plant leaf disease datasets from Kaggle.com [8,13,14]: 1) potato dataset [15] with three classes (early blight, late blight and healthy) and 2) grape dataset [16] with four classes (black rot, black measles, leaf blight and healthy). Some images in potato and grape datasets are shown in Fig. 6. Besides, our simple CNN model in Fig.5, we also use VGG19 and MobileNet to study the impact of our local-based image augmentation method. The numbers of parameters and depths for each model are shown in TABLE I.



Fig. 6. Some images in potato (first row) and grape (second row) datasets.

We set the datasets for training the three models to three cases. Case I is the original images only. Case II is the original images combined with the augmented images, which derive

from only the global-based image augmentation process. Case III is the original images combined with the augmented images, which derive from both our local-based and the global-based image augmentation steps. The number of training images for each case is shown in TABLE II. For programming, we use python on a laptop with Intel® Core™ i5-9300H 4 Core @ CPU 2.40 GHz, GPU NVIDIA GeForce GTX 1650, and DDR4-2666 RAM 32 GB.

TABLE I. THE NUMBER OF PARAMETERS AND DEPTHS OF THE THREE MODELS

Models	Parameters	Depths
CNN in Fig.6	1,998,404	10
VGG19	21,078,606	25
MobileNet	4,807,374	90

TABLE II. THE NUMBER OF TRAINING IMAGES

Plants	No. of classes	Diseases	Number of training images		
			Case I: Original images	Case II: Original images +Global	Case III: Original images +(Local+Global)
Potato [15]	3	Early blight	150	1,050	906
		Late blight	150	1,050	954
		Healthy	150	1,050	1,050
		<b>Total</b>	<b>450</b>	<b>3,150</b>	<b>2,910</b>
Grape [16]	4	Black rot	200	1,400	3,602
		Black Measles	200	1,400	2,504
		Leaf blight	200	1,400	1,190
		Healthy	700	1,400	1,400
<b>Total</b>	<b>800</b>	<b>5,600</b>	<b>8,696</b>		

TABLE III. ACCURACY RATES ON TESTING IMAGES

Plants	Models	Accuracy Rates (%)		
		Case I: Original images	Case II: Original images +Global	Case III: Original images +(Local+Global)
Potato	CNN in Fig.6	93.56	93.56	93.78
	VGG19	90.00	94.89	95.56
	MobileNet	68.22	71.11	77.33
Grape	CNN in Fig.6	91.10	93.20	93.90
	VGG19	95.11	96.00	96.30
	MobileNet	70.98	78.00	80.00

In testing the trained models, we prepare 90 potato leaf images that consist of 30 early blight images, 30 late blight images, and 30 healthy images. Meanwhile, we prepare 200 grape leaf images that include 50 black rot images, 50 black measles images, 50 leaf blight images, and 50 healthy images.

The training accuracy, the validation accuracy, the training loss and the validation loss of our CNN model which is trained by using the training images in Case I and Case II are shown in Fig. 7 and Fig. 8. Meanwhile, the results of using Case III are shown in Fig. 9. We can be seen the graphs of the training accuracy and the validation accuracy of using Case III are smoother than using Case I and Case II. Furthermore, the accuracy rates on the testing images as shown in TABLE III. We can also see that the three models (our CNN model, VGG19 and MobileNet) which are trained via Case III provide the highest accuracy rates when compared to the results of using Case I and Case II in both the potato and the grape datasets. These are the impact of our local-based image

augmentation method which can improve the capability of the models for plant leaf disease classification.

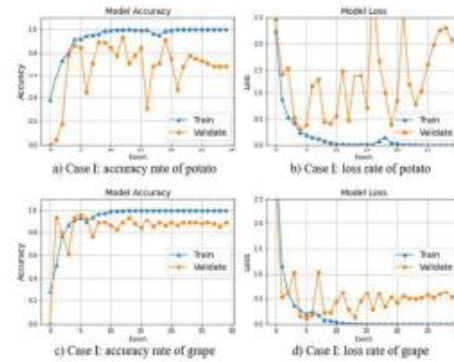


Fig. 7. Accuracy rates and Loss rates of case I.

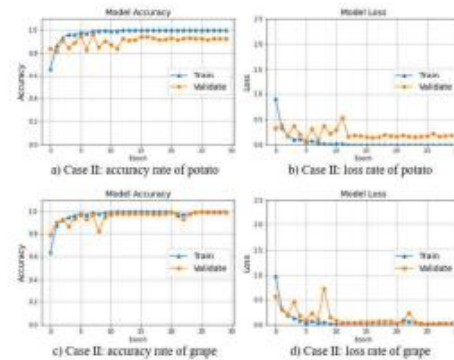


Fig. 8. Accuracy rates and Loss rates of case II.

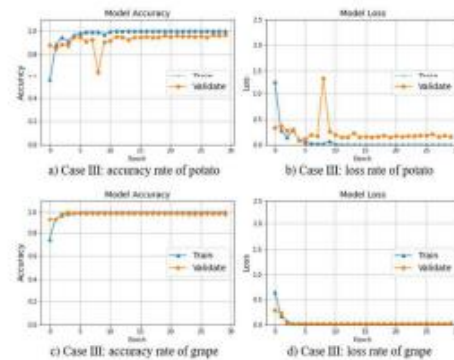


Fig. 9. Accuracy rates and Loss rates of case III.

## V. CONCLUSIONS

A local-based image augmentation method has been proposed in this paper. Our local-based approach is designed to cooperate with the general global-based image augmentation methods and convolutional neural network models for plant leaf disease classification. The original images are augmented by creating the output images, which have a variety of disease distribution on the plant leaf region. The locally augmented images will be forwarded to the global-based image augmentation step to multiply the number of images in a variety of pose, brightness, blur, and noise. Eventually, all augmented images are employed to train the three models, which consist of our simple CNN model, VGG19, and MobileNet. In our experiments, the proposed method can improve the accuracy rates of the three models in both potato and grape datasets.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the contribution of Department of Electrical Engineering, Faculty of Engineering and Industrial Technology, Silpakorn University Scholarship for funding this research. The authors would also like to thank Assistant Professor Dr. Rapepan Kaewon and Dr. Orrathai Watcharakitchakorn for their time and advice.

## REFERENCES

- [1] S. Arya, and R. Singh, "A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf," International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), GHAZIABAD, India, pp. 08977648, September 2019.
- [2] A. Elhassouny, and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks," International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, pp. 08807737, July 2019.
- [3] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," Advanced Optical Imaging for Extreme Environments, vol. 7, pp. 59069-59080, May 2019.
- [4] S. Widiyanto, R. Fitrianto, and D. T. Wardani, "Implementation of Convolutional Neural Network Method for Classification of Diseases in Tomato Leaves," Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, pp. 08985909, October 2019.
- [5] A. H. Deepak, A. Gupta, M. Choudhary, and S. Meghana, "Disease Detection in Tomato plants and Remote Monitoring of agricultural parameters," 11th International Conference on Advanced Computing (ICoAC), Chennai, India, pp. 09087312, December 2019.
- [6] Z. Zheng, S. Pan, and Y. Zhang, "Fruit Tree Disease Recognition Based on Convolutional Neural Networks," IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Shenyang, China, pp. 08982677, October 2019.
- [7] A. Kumar, and M. Vani, "Image Based Tomato Leaf Disease Detection," 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 08944692, July 2019.
- [8] M. Agarwal, R. K. Kaliyar, G. Singal, and S. Kr. Gupta, "FCNN-LDA: A Faster Convolution Neural Network model for Leaf Disease identification on Apple's leaf dataset," 12th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, pp. 08850964, July 2019.
- [9] K. Simonyan, and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Journal of International Conference on Learning Representations 2015 (ICLR 2015), San Diego, CA, USA., May 2015.
- [10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNet: Efficient Convolutional Neural Networks for Mobile Vision Applications," Google Inc., 2017.
- [11] Md. Z. Hasan, Md. S. Ahamed, A. Rakshit, and K. M. Zubair Hasan, "Recognition of Jute Diseases by Leaf Image Classification using Convolutional Neural Network," 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 08944907, July 2019.
- [12] Md. H. Sheikh, T. T. Mim, Md. S. Reza, A. S. Azad Rabby, and S. A. Hossain, "Detection of Maize and Peach Leaf diseases using Image Processing," 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, pp. 08944692, July 2019.
- [13] M. Agarwal, S. Kr. Gupta, and K. K. Biswas, "Grape Disease Identification Using Convolution Neural Network," 23rd International Computer Science and Engineering Conference (ICSEC), Phuket, Thailand, pp. 08974752, October-November 2019.
- [14] A. Adedjoja, P. A. Owolawi, and T. Mapayi, "Deep Learning Based on NASNet for Plant Disease Recognition Using Leave Images," 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), Surabaya, Indonesia, pp. 08851029, August 2019.
- [15] D. Liu, "Plant Diseases(6:2:2)," 2020. [online]. Available: <https://www.kaggle.com/daoliu/plant-diseases>
- [16] S. Bhattarai, "New Plant Diseases Dataset," 2019. [online]. Available: <https://www.kaggle.com/vipocool/new-plant-diseases-dataset>





# การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 44

The 44<sup>th</sup> Electrical Engineering Conference (EECON-44)

ขอแสดงความยินดีที่ขอแสดงความ

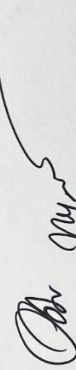
ยินดีที่ วงษ์มณฑิลา และ ไชยภรณ์ ผู้มีจรรยา

ได้เข้าร่วมนำเสนอผลงานเรื่อง

การตรวจประเมินโรตารีเฟรมมอเตอร์เฟรมคู่และเครื่องจ่ายกระแสไฟฟ้าแบบคอนโวลูชัน

ระหว่างวันที่ 17-19 พฤศจิกายน พ.ศ. 2564

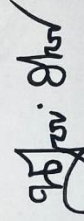
รหัสกิจกรรม 301-03-2025-00/2111-001 (คะแนนหน่วยงานการเรียนรู้ที่ได้รับ 15 PDU)



รองศาสตราจารย์ ดร. นุชิต ไชยประสิทธิ์  
นายกสมาคมวิชาการทางวิศวกรรมไฟฟ้า (ประเทศไทย)



ดร. นุชิต ไชยประสิทธิ์  
อธิการบดีมหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร



รองศาสตราจารย์ ดร. นุชิต ไชยประสิทธิ์  
ประธานคณะกรรมการจัดงานประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 44





**EECON-44**  
Electrical Engineering Conference



## การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 44

The 44<sup>th</sup> Electrical Engineering Conference (EECON-44)  
17-19 พฤศจิกายน 2564  
ณ โรงแรม ที อิมเพรสชั่น อำเภอเมืองน่าน จังหวัดน่าน

- ไฟฟ้ากำลัง (PW)
- ไฟโตนิคส์ (PH)
- คอมพิวเตอร์และเทคโนโลยีสารสนเทศ (CP)
- งานวิจัยที่เกี่ยวข้องกับวิศวกรรมไฟฟ้า (GN)
- วิศวกรรมชีวการแพทย์ (BE)
- ไฟฟ้าสื่อสาร (CM)
- การประมวลผลสัญญาณดิจิทัล (DS)
  - ระบบควบคุมและการวัดคุม (CT)
  - อิเล็กทรอนิกส์กำลัง (PE)
  - พลังงานหมุนเวียน (RE)
  - อิเล็กทรอนิกส์ (EL)









### การตรวจจับบริเวณโรคพืชบนใบโดยใช้เทคนิคซูเปอร์พิกเซลและโครงข่ายประสาทแบบคอนโวลูชัน Leaf Disease Region Detection using Super-pixel Technique and Convolutional Neural Network

จิรศักดิ์ วงษ์ภักขไพศาล<sup>1</sup> และ โสภณ สุมิขรธยา

<sup>1</sup>ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร wongkajit\_j@sil.ac.th, phonsorn.s@sil.ac.th

#### บทคัดย่อ

ในปัจจุบันได้มีการนำเอาความรู้ด้านปัญญาประดิษฐ์มาใช้งานวิจัยและงานในชีวิตประจำวันมากขึ้น การรู้จำ จำนวน และตรวจชนิดของใบพืชเป็นปัญญาประดิษฐ์ที่ได้รับความนิยมเป็นอย่างมากในการนำมาใช้งาน บทความนี้จึงนำเสนอการรู้จำและตรวจชนิดบริเวณโรคบนใบไม้โดยใช้โครงข่ายประสาทแบบคอนโวลูชันและซูเปอร์พิกเซล ในงานวิจัยที่ได้เป็นโปรแกรม โมเดลที่ทำการเรียนรู้การรู้จำโรคบนใบไม้และนำมาทดสอบกับชุดภาพทดสอบจำนวน 100 ภาพ โดยการทำซูเปอร์พิกเซลของ SLIC เพื่อจำแนกบริเวณที่เป็นโรคกับไม่เป็นโรคออกจากกัน จากการทดสอบโมเดลในงานวิจัยนี้แสดงให้เห็นว่าสามารถตรวจชนิดบริเวณที่เป็นโรคได้จากภาพทดสอบคอนโวลูชัน โมเดลและการทำซูเปอร์พิกเซล SLIC ซึ่งได้ความแม่นยำสูงสุด 0.703 ที่  $m = 800$

**คำสำคัญ:** ปัญญาประดิษฐ์ โครงข่ายประสาทแบบคอนโวลูชัน ซูเปอร์พิกเซลของ SLIC การรู้จำโรคบนใบไม้

#### Abstract

Recently, an artificial intelligence is increasingly used in research and everyday life. The recognition, classification, and detection are one of more popular artificial intelligence to use. This research presented recognition and region detection on leaf disease using convolutional neural network and SLIC super-pixel. Result of paper is the trained model of recognized disease and come to investigate on a dataset of 100 test images by SLIC super-pixel to diagnose diseased and non-diseased areas. According to the testing model in this research, the results were able to detect diseased areas by CNN model training and SLIC super-pixel, with an accuracy of 0.703 at  $m = 800$ .

**Keywords:** Artificial Intelligent, Convolutional Neural Network, SLIC super-pixel, Leaf Disease Recognition

#### 1. บทนำ

วิธีการรู้จำประเภทที่ถูกลำมาพัฒนาใช้งานในงานตรวจจับสิ่งของหรือวัตถุในรูปภาพ เทคนิคดังกล่าวได้มีการถูกพัฒนาและนำไปประยุกต์ใช้ร่วมกับงานทางด้านวิทยาศาสตร์ข้อมูล (Data science) โดยใช้โครงข่ายของกรเรียนรู้เชิงลึกอย่างง่าย (Deep learning) สำหรับใช้ฝึกสอนข้อมูลเพื่อสร้างโปรแกรมในการจำแนกบริเวณที่เป็นโรคพืชกับ

บริเวณที่ไม่เป็นโรค อย่างไรก็ตามการได้มาของชุดข้อมูลฝึกสอนนั้นเป็นส่วนที่สำคัญในการทำงานทางด้านนี้เพื่อช่วยให้การฝึกสอนของข้อมูลนั้นมีประสิทธิภาพมากขึ้น

ในการจัดการหรือปรับปรุงข้อมูลสำหรับฝึกสอนจะมีความแตกต่างกันและความเหมาะสมที่ต่างกันออกไปซึ่งขึ้นอยู่กับชนิดของข้อมูลและชนิดของงานที่ทำ การปรับปรุงรูปภาพที่เป็นวิธีการจัดการกับข้อมูลรูปแบบหนึ่งที่น่าสนใจนำมาใช้กับโครงสร้างการเรียนรู้เชิงลึกชนิดโครงข่ายประสาทแบบคอนโวลูชัน โดยมีงานวิจัยหลายงานที่นิยมทำการปรับปรุงข้อมูลรูปภาพแบบทั้งภาพหรือเพิ่มความแตกต่างของข้อมูลแบบทั้งภาพ เช่น ปรับความสว่าง (brightness) อนุภาพ (noise) กลิ่นภาพ (lign) หรือใส่สัญญาณรบกวนให้กับภาพ (noise) เป็นต้น [1-2] โดยผู้วิจัยเห็นว่า การปรับปรุงภาพลักษณะนี้เป็นไปได้ยากในการที่โปรแกรมจะสามารถจำแนกเฉพาะบริเวณที่เป็นโรคอย่างชัดเจนจึงได้มีการนำเสนอการทำซูเปอร์พิกเซลเข้ามาช่วยในการจำแนกเพื่อให้สามารถจำแนกบริเวณโรคได้ดียิ่งขึ้น เนื่องจากการจับกลุ่มของซูเปอร์พิกเซลนั้นพิจารณาจากพิกเซลที่เหมือนกันให้เป็นกลุ่มของพิกเซลเดียวกัน จึงทำให้สามารถจำแนกบริเวณที่เป็นโรคออกจากใบไม้ได้อย่างแม่นยำ

จากที่กล่าวมาข้างต้นผู้วิจัยได้ออกแบบโปรแกรมสำหรับการจำแนกบริเวณที่เป็นโรคด้วยการทำซูเปอร์พิกเซล [3] เพื่อสร้างข้อมูลสำหรับนำไปใช้ฝึกสอนมีขั้นตอนและโครงสร้างทางสถาปัตยกรรมของโมเดลที่ใช้สำหรับฝึกสอนเป็นโมเดลโครงข่ายประสาทแบบคอนโวลูชัน ซึ่งมีองค์ประกอบดังที่จะกล่าวในหัวข้อถัดไป และโปรแกรมที่ได้จากการฝึกสอนข้อมูลจะเป็นโปรแกรม .h5 ที่นำไปใช้สำหรับจำแนกบริเวณที่เป็นโรคด้วยการทดสอบกับชุดข้อมูลทดสอบและเปรียบเทียบค่าความถูกต้องกับงานวิจัยของ Sachin [4] ที่ทำเกี่ยวกับกรแบ่งส่วนโรคของพืชด้วยวิธีการจัดกลุ่มข้อมูลแบบ K-mean และเทียบกับชุดข้อมูลที่คำนวณมองเห็นด้วยการทำ F1-score หาค่าเฉลี่ย ( $\mu$ ) ค่าเบี่ยงเบนมาตรฐาน ( $S.D.$ ) ค่าต่ำสุด (min) และค่าสูงสุด (max) ตามลำดับ

#### 2. ทฤษฎีที่เกี่ยวข้อง

##### 2.1 ซูเปอร์พิกเซลของ SLIC

ซูเปอร์พิกเซล คือ ขอบเขตบริเวณพื้นที่ในภาพที่เกิดจากการรวมกันของกลุ่มพิกเซลสีในภาพที่อยู่ติดกันและมีพิกเซลที่มีลักษณะที่คล้ายคลึงกัน เช่น มีช่วงสีที่ใกล้เคียงกัน มีความสว่างเท่ากัน มีพื้นผิวที่เหมือนกัน และคุณสมบัติอื่น ๆ ที่ใกล้เคียงกันของพิกเซลบริเวณนั้น ๆ โดยในงานวิจัยนี้ได้ทำการแบ่งสัดส่วนพื้นที่บริเวณบนใบไม้ด้วยวิธี



การประชุมวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 44  
 The 44<sup>th</sup> Electrical Engineering Conference (EECON44)  
 วันที่ 17-19 พฤศจิกายน 2564 ณ โรงแรม ดิ อิมพารัส บาน จันทรมณี ภูเก็ต



ซูเปอร์พิกเซลชนิด SLIC (Simple Linear Iterative Clustering) ซึ่งเป็น ซูเปอร์พิกเซลชนิดหนึ่งที่มีเมตริกใช้งาน โดยเปลี่ยนเวกเตอร์สี 3 มิติ ของรูปภาพจาก  $V_{rgb} = [r, g, b]$  เป็น  $V_s = [L, a, b, x, y]$  ขนาด 5 มิติ โดยที่  $[L, a, b]$  เป็นเวกเตอร์ของพิกเซลสีในปริภูมิสี CIELAB และ  $[x, y]$  เป็น เวกเตอร์ตำแหน่งของพิกเซลต่าง ๆ [5]

การทำงานของ SLIC เริ่มโดยการเลือกจำนวน ซูเปอร์พิกเซล จำนวน  $K$  ขึ้น โดยชรั้งขอบเขตด้วยขั้วขึ้นมา หากจุดใดห่างจากสมการ  $C_k = [L_k, a_k, b_k, x_k, y_k]^T$  ที่  $k = [1, K]$  หระยะห่างของจุดถึงกลางระหว่าง ซูเปอร์พิกเซลได้  $S = \sqrt{N/K}$  ที่  $N$  คือจำนวนพิกเซลของรูปภาพ และ  $N/K$  คือขนาดพื้นที่ของซูเปอร์พิกเซล และทำการย้ายตำแหน่งจุด ที่กลางของซูเปอร์พิกเซลที่หาได้ใหม่ [6-8] จากสมการที่ (1) ถึง (3)

$$d_{L,a,b} = \sqrt{(L_k - L)^2 + (a_k - a)^2 + (b_k - b)^2} \quad (1)$$

$$d_{x,y} = \sqrt{(x_k - x)^2 + (y_k - y)^2} \quad (2)$$

$$D_k = d_{L,a,b} + \left(\frac{m}{S}\right) d_{x,y} \quad (3)$$

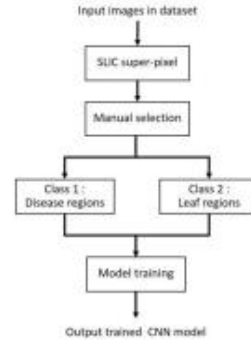
โดยที่  $d_{L,a,b}$  คือ ระยะทางแบบยูคลิดของปริภูมิสี CIELAB  
 $d_{x,y}$  คือ ระยะทางแบบยูคลิดบนระนาบ  $xy$   
 $D_k$  คือ ค่าความห่างที่หาค่าของซูเปอร์พิกเซลใหม่  
 $m$  คือ ค่าความหนาแน่นของซูเปอร์พิกเซล เป็นค่าคงที่

**2.2 โครงข่ายประสาทแบบคอนโวลูชัน**

โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network : CNN) เป็นรูปแบบการเรียนรู้เชิงลึกชนิดหนึ่งที่มีการสกัด คุณสมบัติเฉพาะของข้อมูลออกมา ถูกนิยมนำมาใช้งานในด้านการจำแนกรูปภาพ วัตถุ เป็นอีกรูปแบบพื้นฐานที่สามารถสร้างได้ง่าย โดยกรรมาเอา โครงสร้างพื้นฐานของโครงข่ายประสาทแบบ MLP (Multi-Layer Perceptron) ที่ประกอบด้วย ชั้นนำเข้า (input layer) ชั้นที่ซ่อนอยู่ (hidden layer) และชั้นนำออก (output layer) โดยเพิ่มการคำนวณของ คอนโวลูชันเข้าไปในชั้นที่ซ่อนอยู่ทำให้เกิดการสกัดคุณสมบัติที่มากขึ้นและเกิดการเตือนรู้ที่หลากหลายกว่าโครงข่ายประสาทแบบ MLP โดยในงานวิจัยนี้ได้ออกแบบโครงข่ายประสาทแบบ CNN ที่ประกอบด้วยชั้น การคอนโวลูชัน (convolution layer) ฟังก์ชันเชิงเส้นตรงที่ถูกปรับแก้ (rectified linear unit : ReLu) ชั้นตัวกรอง max pooling ชั้นการ drop out ชั้นการ flatten และชั้นการทำ fully connected เพื่อใช้ในการทดลอง [9-10]

**3. การดำเนินการและวิธีการทดลอง**

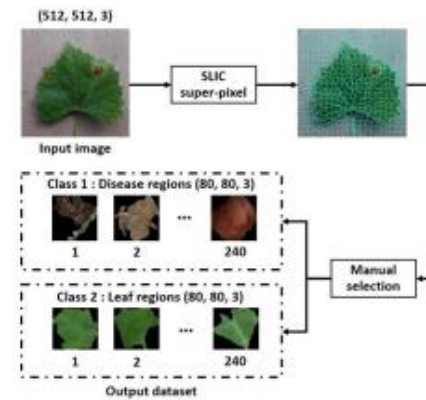
ในงานวิจัยนี้ได้ทำการทดลองบนคอมพิวเตอร์โน้ตบุ๊กที่ใช้หน่วยประมวลผลกลาง (CPU) Intel® Core™ i5-9300H 4 แกน มีความถี่ 2.40 GHz หน่วยประมวลผลกราฟิก NVIDIA GeForce GTX 1650 RAM DDR4-2666 32 GB โดยในบทความนี้มีขั้นตอนในการฝึกสอน CNN model ดังรูปที่ 1



รูปที่ 1 ขั้นตอนการฝึกสอน CNN model

**3.1 การแบ่งส่วนภาพด้วยวิธีซูเปอร์พิกเซลแบบ SLIC**

ในการจำแนกรูปภาพ โรคและใบออกจากกันสำหรับนำไปใช้ ฝึกสอนเพื่อสร้างโมเดลนั้น ในงานวิจัยได้ใช้ภาพที่เป็นภาพอินพุตและนำ ภาพเข้าสู่กระบวนการทำ SLIC ซูเปอร์พิกเซล เพื่อจำแนกพื้นที่ส่วน บริเวณที่เป็นโรคกับบริเวณที่เป็นใบออกจากกันโดยแสดงขอบเขต บริเวณของพื้นที่นั้น ๆ ที่ SLIC จำนวนออกมาได้ และทำการเลือกบริเวณ ที่เป็นโรคและใบด้วยตามคุณสมบัติของกระบวนการทำดังรูปที่ 2 จำนวน อย่างละ 240 ชุด เพื่อนำไปใช้ในกระบวนการฝึกสอนและสร้างโมเดล

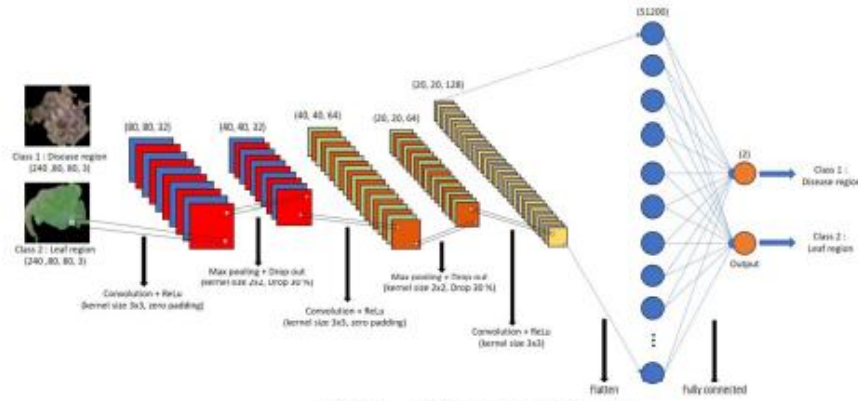


รูปที่ 2 ขั้นตอนการทำ super-pixel สำหรับสร้างชุดข้อมูลฝึกสอน

**3.2 การสร้างชุดข้อมูลฝึกสอนของบริเวณที่เป็นใบและโรค**

จากรูปที่ 2 หลังจากภาพใบไม้ได้ผ่านกระบวนการซูเปอร์พิกเซล มาแล้ว โดยในบทความนี้ได้ใช้บริเวณที่เป็นโรคจำนวน 240 ภาพ และ บริเวณที่เป็นใบอีก 240 ภาพ



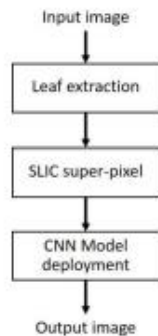


รูปที่ 3 CNN model ที่ใช้ในการจำแนกบริเวณที่เป็นโรคพืชบนใบ

**3.3 การฝึกสอน CNN model**

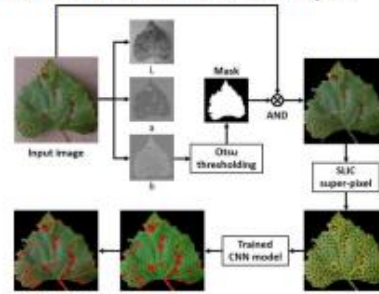
ในขั้นตอนการฝึกสอนและสร้างโมเดลจะนำภาพถ่ายที่ได้จากขั้นตอนก่อนหน้ามาใช้เป็นข้อมูลสำหรับการฝึกสอนของโมเดลที่ออกแบบ โดยในงานวิจัยนี้ได้ทำการออกแบบโมเดลโครงข่ายประสาทแบบคอนโวลูชันที่มีโครงสร้างประกอบด้วย ชั้นการทำคอนโวลูชัน + ฟังก์ชันเส้นตรงที่ถูกปรับแก้ 3 ชั้น (convolution layer + ReLU) ชั้นการทำ drop out 2 ชั้น ชั้นการทำ max pooling 2 ชั้น ชั้นการทำ flatten และ fully connected อย่างละ 1 ชั้น ดังรูปที่ 3 โดยเอาผลลัพธ์ของโมเดลที่ได้มีด้วยกัน 2 กลุ่ม (classes) 1. กลุ่มที่เป็นโรค 2. กลุ่มที่เป็นใบ

**3.4 การแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ได้ฝึกสอนมาแล้ว**



รูปที่ 4 ขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ CNN model ที่ผ่านการฝึกสอนมาแล้ว จากวิธีก่อนหน้าจะได้โมเดลไฟล์ .h5 จากการฝึกสอนข้อมูลสำหรับนำมาใช้จำแนกพื้นที่ ๆ เป็นบริเวณโรคกับใบโดยมีขั้นตอนการทำงาน

ตามรูปที่ 4 เริ่มจากการนำภาพสำหรับทดสอบโมเดลที่เตรียมไว้จำนวน 100 ภาพ มาทำการจำแนกส่วนที่เป็นใบออกจากพื้นหลังส่วนภาพที่ได้ไปเข้าสู่วิธีการประมวลผลภาพเพื่อฝึกและโมเดลโครงข่ายประสาทแบบคอนโวลูชันเพื่อจำแนกบริเวณที่เป็นโรค ดังที่กล่าวมาข้างต้นมีขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรคโดยใช้ผลลัพธ์ดังรูปที่ 5 :



รูปที่ 5 ขั้นตอนการแบ่งส่วนบริเวณที่เป็นโรค

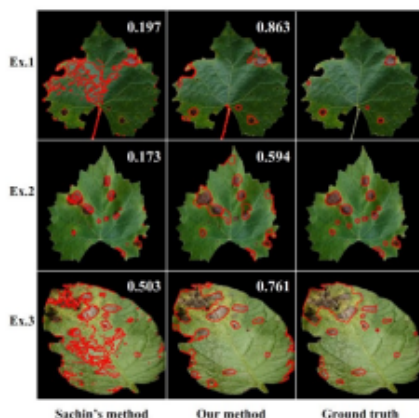
**4. ผลการทดลอง**

ในการทดลองนี้ได้ใช้ชุดข้อมูลภาพจากฐานข้อมูล Kaggle.com [11] ประกอบไปด้วยใบอ่อนจำนวน 58 ภาพ และใบมันฝรั่งจำนวน 42 ภาพ ผลการทดลองที่ได้แสดงดังรูปที่ 6 โดยในแถวของแต่ละแถวเป็นผลลัพธ์ที่ได้จากข้อมูลภาพเดียวกัน ส่วนใน column 1 เป็นผลลัพธ์ของการจำแนกโรคด้วยวิธีการของ Sachin column 2 เป็นผลลัพธ์ของการจำแนกโรคด้วยวิธีที่ผู้วิจัยนำเสนอ column 3 เป็นภาพสีแบบที่จำแนกด้วยมือ สำหรับนำมาเปรียบเทียบกันภาพของงานวิจัยที่นำเสนอทั้งสองงานวิจัย

การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 44  
 The 44<sup>th</sup> Electrical Engineering Conference (EECON44)  
 วันที่ 17-19 พฤศจิกายน 2564 ณ โรงแรม ดิ อิมพเรส น่าน อังคณิณีบ้าน จิตรวิวัฒนา



การวัดประสิทธิภาพของงานวิจัยในบทความนี้ ตัวชี้วัดที่ใช้เปรียบเทียบความถูกต้องของผลการจำแนกบริเวณที่เป็นโรคของวิธีที่นำเสนอกับวิธีการจากงานวิจัยของ Sachin และคณะ ได้แก่ F1-score [12] สามารถคำนวณได้ดังสมการที่ (4) และวิเคราะห์ค่า  $\mu$ , S.D., min และ max ของทั้งสองงานวิจัย ตามลำดับ ผลลัพธ์แสดงดังตารางที่ 1 โดยได้มีการทดลองปรับค่า m ซึ่งเป็นค่าแสดงถึงความหนาแน่นของจุดเปอร์ฟิเคชันในช่วง 300 ถึง 800 โดยการเลือกใช้ค่า m นั้นจะพิจารณาจากขนาดของบริเวณที่เป็นโรคนั้น



รูปที่ 6 ผลการแบ่งส่วนบริเวณที่เป็นโรค

$$F1\text{-score} = \frac{2TP}{2TP + FP + FN} \quad (4)$$

โดยที่ TP = True Positive, FP = False Positive, FN = False Negative

ตารางที่ 1 การเปรียบเทียบความถูกต้องในการแบ่งส่วนบริเวณที่เป็นโรคระหว่างวิธีของ Sachin เทียบกับวิธีการที่นำเสนอ

Methods	F1-score				
	$\mu$	S.D.	min	max	
Sachin's method	0.386	0.2083	0.0130	0.8187	
Our method	m				
	300	0.617	0.1712	0.0807	0.9265
	400	0.647	0.1607	0.0935	0.9351
	500	0.664	0.1492	0.1338	0.9414
	600	0.685	0.1385	0.1194	0.9407
	700	0.703	0.1322	0.1219	0.9394
800	0.703	0.1319	0.1498	0.9420	

### 5. สรุป

ในงานวิจัยนี้ได้ออกแบบและพัฒนาโปรแกรมการจำแนกบริเวณที่เป็นโรคนั้นโดยวิธีการแบ่งส่วนภาพด้วยจุดเปอร์ฟิเคชันเพื่อแบ่งบริเวณที่เป็นโรคออกจากส่วนใบเพื่อนำไปเป็นข้อมูลเพื่อนำไปฝึกสอน

ด้วยโมเดลโครงข่ายประสาทแบบคอนโวลูชัน จากการทดลองพบว่า การนำจุดเปอร์ฟิเคชันมาใช้สำหรับจำแนกบริเวณที่เป็นโรคโดยอ้างอิงบริเวณที่มีสีคล้ายกันเป็นพื้นที่เดียวกันและนำมาใช้ร่วมกับขั้นตอนการทดสอบพบว่าผลของการทำนายของโปรแกรมเมื่อนำมาเปรียบเทียบด้วยค่า F1-score ได้ค่าความถูกต้องเฉลี่ยสูงสุด 0.703 ที่ m = 800 เมื่อเทียบกับผลลัพธ์ที่ได้ในงานวิจัยของ Sachin กับชุดข้อมูลทดสอบเดียวกันได้ค่าความถูกต้องเฉลี่ยอยู่ที่ 0.386

### กิตติกรรมประกาศ

ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปากร สำหรับรับสนับสนุนงานวิจัยนี้

### เอกสารอ้างอิง

- [1] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," *Journal of IEEE Access*, vol. 7, no. 1, May, pp. 59069-59080, 2019.
- [2] A. H. Deepak, A. Gupta, M. Choudhary, and S. Meghana, "Disease Detection in Tomato plants and Remote Monitoring of agricultural parameters," In Proc. 11th International Conference on Advanced Computing (ICoAC), 2019, pp. 28-33.
- [3] S. Zhang, Z. You, and X. Wu, "Plant disease leaf image segmentation based on superpixel clustering and EM algorithm," *Journal of Neural Computing and Applications*, June, 2017.
- [4] S. D. Khirale, and A. B. Patil, "Plant Disease Detection Using Image Processing," In Proc. International Conference on Computing Communication Control and Automation (ICCCA), 2015, pp. 768-771.
- [5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Siostrunk, "SLIC Superpixels," in *Ecole Polytechnique Fédérale de Lausanne (EPFL) School of Computer and Communication Sciences (CC)*, [online document], 2010. Available: EPFL Technical Report 149300, <http://www.epfl.com> [Accessed: June 2010].
- [6] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Siostrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 31, Nov, pp. 2274-2282, 2012.
- [7] D. Haiying, and J. Tengguang, "Rapid detection of obstacle based on super pixels," In Proc. The 27th Chinese Control and Decision Conference (CCDC), 2015, pp. 4993-4995.
- [8] M. Yang, Z. Li, J. Liu, "Super-pixel Based Single Image Haze Removal," In Proc. The 28th Chinese Control and Decision Conference (CCDC), 2016, pp. 1965-1969.
- [9] A. Elhassoumy, and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks," presented at International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, July 2019.
- [10] P. Sharma, P. Hans, and S. C. Gupta, "Classification of Plant Leaf Diseases Using Machine Learning and Image Preprocessing Techniques," In Proc. 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 480-484.
- [11] S. Bhattacharjee, "New Plant Diseases Dataset," 2019. [online]. Available: <https://www.kaggle.com/viprassad/new-plant-diseases-dataset>
- [12] สมชาย สีนเทพ, จักรวรรดิ ดิ่ง, ชะพอส ศรีเจริญ, และ นิตติศักดิ์ หนูมาบัต, "การวิเคราะห์สายสัมพันธ์สายใยการควบคุมขนาดผลทวารด้วยวิธีการเรียนรู้ของเครื่องจักร," ในการประชุมวิชาการทางวิศวกรรมไฟฟ้า (EECON) ครั้งที่ 43: 2563, หน้า 141-144.

## ประวัติผู้เขียน

ชื่อ-สกุล	จรัสศักดิ์ วงษ์บงกชไพศาล
วัน เดือน ปี เกิด	18 มีนาคม 2540
สถานที่เกิด	ราชบุรี
วุฒิการศึกษา	สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น โรงเรียนเบญจมราชูทิศ ราชบุรี ปีการศึกษา 2554 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนสายธรรมจันทร์ ปีการศึกษา 2557 สำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และระบบคอมพิวเตอร์ มหาวิทยาลัยศิลปากร ปีการศึกษา 2561
ที่อยู่ปัจจุบัน	
ผลงานตีพิมพ์	Jirasak Wongbongkotpaisan and Sopon Phumeechanya, 2021, "Plant Leaf Disease Classification using Local-Based Image Augmentation and Convolutional Neural Network", International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1023-1027. จรัสศักดิ์ วงษ์บงกชไพศาล และ โสภณ ผู้มีจรรยา, "การตรวจจับบริเวณโรคพืชบนใบโดยใช้เทคนิคซูเปอร์พิกเซลและโครงข่ายประสาทแบบคอนโวลูชัน (Leaf Disease Region Detection using Super-pixel Technique and Convolutional Neural Network)," การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 44 (EECON44), 17-19 พฤศจิกายน 2564, จังหวัดน่าน, หน้า 465-468.