



การจำแนกโรคใบข้าวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO
RICE LEAF DISEASES CLASSIFICATION USING YOLO CONVOLUTIONAL
NEURAL NETWORK

ธาวิน ภัคดีพรเลิศ

โปรแกรมวิศวกรรมคอมพิวเตอร์

คณะเทคโนโลยีดิจิทัล

มหาวิทยาลัยราชภัฏเชียงใหม่

ปีการศึกษา 2566

© ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏเชียงใหม่

การจำแนกโรคใบขาวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO

ธาวิน ภัคดีพรเลิศ

ปริญญาตรี สาขาวิชาวิศวกรรมคอมพิวเตอร์

โปรแกรมวิชาวิศวกรรมคอมพิวเตอร์

คณะเทคโนโลยีดิจิทัล มหาวิทยาลัยราชภัฏเชียงราย

2566

© ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏเชียงราย



ใบรับรองงานวิจัย
คณะเทคโนโลยีดิจิทัล มหาวิทยาลัยราชภัฏเชียงราย

เรื่อง การจำแนกโรคใบขาวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO
โดย นายธาวิณ ภัคดีพรเลิศ

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ 2566

.....รองคณบดีฝ่ายวิชาการ
(ผู้ช่วยศาสตราจารย์ อนุสรณ์ ใจแก้ว)

คณะกรรมการสอบงานวิจัย

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กฤตกรณ์ ศรีวันนา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.มยุร ไยบัวเทศ)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ธนาวุฒิ ธนวานิชย์)

.....กรรมการ
(อาจารย์ อธิคม ศิริ)

กิตติกรรมประกาศ

การพัฒนาการจำแนกโรคใบขาวโดยใช้โครมโซ่สายประสาทเทียมสำเร็จลุล่วงไปด้วยดีไม่ได้ หากไม่ได้รับความกรุณาจากบุคคลหลาย ๆ ท่าน ข้าพเจ้าขอขอบคุณบุคคลสำคัญดังที่จะกล่าวถึงดังต่อไปนี้เป็นอย่างยิ่งที่ช่วยให้คำแนะนำแก้ไขปัญหา ตลอดจนช่วยให้ข้าพเจ้าทำการพัฒนาการจำแนกโรคใบขาวโดยใช้โครมโซ่สายประสาทเทียม ซึ่งได้แก่ บิดามารดา ที่ให้การสนับสนุน ตลอดจนอาจารย์ ผู้ช่วยศาสตราจารย์ ดร.กฤตกรณ์ ศรีวันนา อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ภูมิพงษ์ ดวงตั้ง ผู้ช่วยศาสตราจารย์ ดร.ธนาวุฒิ ธนวานิชย์ ผู้ช่วยศาสตราจารย์ กมล บุญล้อม ผู้ช่วยศาสตราจารย์ ดร.มยุร ไยบัวเทศ และ อาจารย์อริศม ศิริ ที่คอยสั่งสอนวิชาเพื่อนำไปใช้งานการทำงาน ให้คำปรึกษาและคำแนะนำในการพัฒนาการจำแนกโรคใบขาวโดยใช้โครมโซ่สายประสาทเทียม ตั้งแต่เริ่มต้นจนเสร็จสิ้นกระบวนการทำงาน

สำหรับคุณงามความดีที่เกิดจากการพัฒนาการจำแนกโรคใบขาวโดยใช้โครมโซ่สายประสาทเทียม ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีต่อข้าพเจ้า ช่วยเหลือและแนะนำสิ่งต่าง ๆ ให้กับ ข้าพเจ้า

นายธาวิน ภัคดิพรเลิศ

20 ตุลาคม 2566

ชื่องานวิจัย	การจำแนกโรคใบข้าวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO
ชื่อผู้วิจัย	นายธาวิน ภัคดิพรเลิศ
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2566
อาจารย์ที่ปรึกษา	อาจารย์ ดร.กฤตกรณ์ ศรีวันนา

บทคัดย่อ

เนื่องจาก ข้าว เป็น อาหาร หลัก ที่ คน ไทย บริโภค และ เป็น พืช เศรษฐกิจ ที่ สำคัญ ของ ประเทศไทย หนึ่งในปัจจัยที่ทำให้ข้าวมีคุณภาพลดลงคือโรคข้าว งานวิจัยนี้มีจุดประสงค์เพื่อจำแนกโรคของใบข้าวจากภาพถ่ายใบข้าว โดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO ข้อมูลที่ใช้ศึกษาเป็นภาพโรคข้าว 3 โรค คือโรคจุดน้ำตาล โรคแมลงดำหนาม และโรคใบไหม้ การศึกษานี้จะทำการเปรียบเทียบประสิทธิภาพโมเดล YOLO ทั้งหมด 4 โมเดล ประกอบด้วย YOLOv5s, YOLOv5m, YOLOv5l, และYOLOv5x โดยเปรียบเทียบประสิทธิภาพด้านความแม่นยำและประสิทธิภาพด้านเวลา ซึ่งแต่ละแต่ละโมเดลได้ทำการ เรียนรู้ 200 รอบ จากการศึกษาและผลการทดลองพบว่า โมเดล YOLOv5l มีประสิทธิภาพมากที่สุด โดยมีความแม่นยำ 71.96% และใช้เวลาในการเรียนรู้ชุดข้อมูลสำหรับสร้างโมเดลในการจำแนกโรคอยู่ที่ 3 ชั่วโมง 42 นาที

คำสำคัญ: โรคข้าว, YOLO, โรคจุดน้ำตาล, โรคแมลงดำหนาม, โรคใบไหม้

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อ	ข
สารบัญตาราง	จ
สารบัญภาพ	ฉ
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	1
1.3 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย	2
1.4 ขอบเขตของการวิจัย	2
2 แนวคิด ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง	3
2.1 ทฤษฎีและแนวคิด	3
2.2 งานวิจัยที่เกี่ยวข้อง	8
3 วิธีดำเนินการวิจัย	11
3.1 เครื่องมือที่ใช้ในงานวิจัย	12
3.2 กรอบแนวคิดภาพรวมของระบบ	13
3.3 การรวบรวมข้อมูล	14
3.4 การเตรียมข้อมูลสำหรับการสร้างโมเดลการพยากรณ์	15
3.5 การสร้างโมเดลการทำนาย	16
3.6 การทดสอบการจำแนกโรคข้าว	19
3.7 การออกแบบการทดลอง	20

สารบัญ (ต่อ)

	หน้า
บทที่	
4 ผลการดำเนินงานวิจัย	22
4.1 ผลการทดลองประสิทธิภาพด้านเวลาของโมเดลทำนาย	22
4.2 ผลการทดลองประสิทธิภาพด้านความแม่นยำของโมเดลทำนายโดยใช้ตาราง ConfusionMatrix	23
5 สรุปและอภิปรายผล	26
5.1 สรุปผลและอภิปรายผล	26
5.2 ข้อจำกัด	27
5.3 ข้อเสนอแนะและงานวิจัยในอนาคต	28
บรรณานุกรม	30
ภาคผนวก	31
ก โค้ดของการสร้างโมเดลจำแนกภาพโรคใบขาว	32
ประวัติย่อของผู้วิจัย	40

สารบัญตาราง

ตารางที่		หน้า
3.1	การตั้งค่าพารามิเตอร์ให้โมเดล YOLO	18
4.1	ผลเวลาการสร้างโมเดลทำนายของ YOLO	22

สารบัญภาพ

ภาพที่	หน้า	
3.1	แผนผังการดำเนินงานวิจัยการจำแนกโรคข้าว	12
3.2	แผนผังแสดงภาพรวมของการจำแนกโรคข้าว	14
3.3	การสร้างกรอบป้ายชื่อในการจำแนกโรค	15
3.4	ภาพชุดข้อมูล	16
3.5	การแบ่งชุดข้อมูล	17
3.6	การตรวจจับวัตถุแบบหนึ่งขั้นตอน	18
3.7	ตัวภาพผลลัพธ์การจำแนกโรค	19
3.8	ConfusionMatrix	21
4.1	ตาราง ConfusionMatrixYOLOv5s	23
4.2	ตาราง ConfusionMatrixYOLOv5m	24
4.3	ตาราง ConfusionMatrixYOLOv5l	24
4.4	ตาราง ConfusionMatrixYOLOv5x	25
4.5	กราฟความแม่นยำในการจำแนกโรคของ โมเดลYOLOv5	25
5.1	กราฟแสดงความแม่นยำเฉลี่ยในการจำแนกโรคของโมเดลทำนายYOLOv5	27

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันนี้ข้าวเป็นอาหารหลักที่มนุษย์บริโภคและยังเป็นพืชเศรษฐกิจของประเทศไทย อีกทั้งยังเป็นผลผลิตที่ส่งออกเป็นอันดับต้นๆของประเทศไทย และโรคจากใบข้าวก็เป็นหนึ่งในปัจจัยหลักที่ทำให้ผลผลิตและคุณภาพลดลง ซึ่งถ้าสามารถรู้ถึงโรคที่กำลังเป็นก็จะสามารถทราบได้ถึงสาเหตุหรือวิธีรักษาโรคได้ ก็เป็นการช่วยให้เกษตรกรได้ทำการรักษาได้ทันเวลาที่ และยังช่วยให้ผลผลิตที่ได้มีคุณภาพและเพิ่มปริมาณผลผลิต

งานวิจัยในปัจจุบันที่เกี่ยวกับการจำแนกที่โรคจากใบข้าว คุณเตชินท์ วรสิทธิ์และคณะ ได้นำเสนอไว้ในงานวิจัยการประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกสำหรับการรู้จำโรคขอบใบแห้ง และโรคไหม้ของใบข้าว และงานวิจัยของคุณณัฐภัค ลาวัลย์วงศ์ การจำแนกโรคอ้อยที่พบมากบนพันธุ์ขอนแก่น 3 โดยใช้ระบบอัจฉริยะสำหรับจำแนกประเภทและลักษณะอาการของโรค อ้อยจะให้ความสำคัญในการวิเคราะห์โรคที่ส่งผลกระทบต่ออ้อย โดยระบบมีความสามารถในการจำแนกชนิดของโรค

งานวิจัยนี้ได้มีการนำภาพมาจัดทำชุดที่ใช้สำหรับให้ระบบเรียนรู้ โดยโรคที่ให้ระบบเรียนรู้ได้แก่ โรคใบจุดสีน้ำตาล โรคแมลงดำหนาม และโรคใบไหม้ โรคเหล่านี้เป็นตัวอย่างโรคที่ขอบสร้างเสียหายในพื้นที่ของผู้จัดจำ งานวิจัยในครั้งนี้สามารถตรวจจับโรคในใบข้าวได้ ตามเป้าหมายตามที่ให้ระบบได้เรียนรู้

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อการจำแนกโรคของข้าวจากภาพถ่าย
2. เพื่อการจำแนกโรคข้าวที่มีลักษณะใกล้เคียงกัน
3. เพื่อประเมินประสิทธิภาพของโมเดล YOLO

1.3 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1. ได้โมเดลที่ใช้สำหรับการจำแนกโรคของข้าวจากภาพถ่าย
2. ช่วยลดความเสียหายจากการเกิดโรค
3. ได้ทราบประสิทธิภาพของโมเดล YOLO

1.4 ขอบเขตของการวิจัย

1. เป็นระบบที่ตรวจจับ 3 โรค ได้แก่ โรคใบจุดสีน้ำตาล, โรคแมลงดำหนาม, โรคใบไหม้
2. ใช้จำแนกโรคจากภาพใบข้าว

บทที่ 2

แนวคิด ทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง

งานวิจัยเรื่องนี้ผู้วิจัยได้ศึกษาค้นคว้า และงานวิจัยที่เกี่ยวข้องเพื่อนำมาใช้สำหรับการกำหนดกรอบความคิด หลักการ ทฤษฎี เครื่องมือ เช่น YOLO, Google Colab การรวบรวมข้อมูล การวิเคราะห์และการอภิปรายผลการศึกษา ซึ่งประกอบไปด้วยเนื้อหา ดังต่อไปนี้

2.1 ทฤษฎีและแนวคิด

2.1.1 ปัจจัยของโรคที่มีผลต่อการเจริญเติบโตของข้าว

1. คุณภาพข้าว โรคข้าวสามารถมีผลต่อคุณภาพข้าวได้โดยตรง เช่น การทำให้เส้นใยข้าวสั้นลง การทำให้ข้าวเปื่อยนุ่มลง หรือการทำให้เกิดจุลินทรีย์ที่ทำให้เกิดกลิ่นหรือรสชาติที่ไม่พึงประสงค์ในข้าว เช่น โรคข้าวเปื่อย หรือโรคข้าวแบคทีเรีย

2. ผลผลิตข้าว โรคข้าวสามารถทำให้ผลผลิตข้าวลดลงได้ เนื่องจากการที่พืชถูกทำลายโดยโรคนั้น จะส่งผลให้ข้าวไม่สามารถเจริญเติบโตได้ตามปกติ ทำให้มีการเสียหายในการสร้างเมล็ดพันธุ์ นอกจากนี้ สภาวะโรคที่รุนแรงยิ่งขึ้นอาจทำให้เกิดการเสียหายต่อพืชที่สูงขึ้น

3. ความเสียหายเศรษฐกิจ การระบาดของโรคข้าวอาจมีผลกระทบต่อเศรษฐกิจของประเทศหรือพื้นที่ที่ปลูกข้าว เนื่องจากการลดผลผลิตข้าวและนัยสำคัญ อีกทั้งยังมีค่าใช้จ่ายในการควบคุมโรค รวมถึงค่าเสียหายจากการสูญเสียวัสดุและแรงงานในกระบวนการผลิต

4. การกระจายเชื้อโรค โรคข้าวอาจกระจายไปสู่แปลงข้าวอื่น ๆ และกระทบต่อสภาพแวดล้อมทางการเกษตรโดยรอบ ซึ่งอาจส่งผลให้เกิดการระบาดของโรคได้อีกมาก

5. ความเสียหายสำหรับเกษตรกร โรคข้าวอาจส่งผลให้เกษตรกรสูญเสียรายได้ และเสียหายในการผลิตข้าว เนื่องจากต้องใช้งบประมาณในการควบคุมโรค รวมถึงการลดผลผลิตที่เกิดจากการระบาดของโรค

2.1.1.1 ปัจจัยด้านสิ่งแวดล้อม

1. ปัจจัยทางดิน สภาพดิน คุณภาพและโครงสร้างของดินมีผลต่อการเจริญเติบโตของข้าว เช่น ดินร่วนหรือดินเหนียวสามารถเก็บน้ำและธาตุอาหารได้ดีกว่า ดินทราย ดินหน้าดินที่ชุ่มชื้นก็สามารถสร้างสภาพที่เหมาะสมสำหรับการเจริญเติบโตของข้าวได้

2. ที่ยึดเหนี่ยว ดิน พืชจะเจริญเติบโตและให้ผลผลิตได้ดี ต้องมีที่ยึดเหนี่ยวที่แข็งแรง เพื่อให้ลำต้นทรงอยู่ได้ในลักษณะที่เหมาะสมที่สุด ซึ่งจะทำให้ส่วนต่างๆ ทำหน้าที่ในการเจริญเติบโตได้อย่างเต็มที่ pH หรือความเป็นกรด ต่างในดิน เกี่ยวข้องกับปัจจัยเรื่องดินและน้ำเป็นหลัก ซึ่งมีผลต่อการเจริญเติบโตของพืช

3. อุณหภูมิที่เหมาะสมจะส่งเสริมให้พืชมีการเจริญเติบโตและพัฒนาการที่ดี อุณหภูมิ 25 - 33 °C เหมาะสำหรับข้าวแตกกอ หากอุณหภูมิต่ำเกินไปหรือสูงเกินไป ต่ำกว่า 15 องศาเซลเซียส สูงกว่า 35 องศาเซลเซียส จะมีผลต่อการงอกของเมล็ด การยึดของใบ การแตกกอ การสร้างดอกอ่อน การผสมเกสร เป็นต้น เช่น พบว่าอุณหภูมิที่สูงเกินไปและต่ำเกินไป ช่วงที่มีการออกดอกจะทำให้ดอกข้าวเป็นหมัน ซึ่งจะส่งผลทำให้ได้ผลผลิตต่ำกว่าปกติ อุณหภูมิที่เหมาะสมตามช่วงระยะการเจริญเติบโต มีดังนี้ เมล็ดงอก 18 - 40 °C ต้นอ่อน 25 - 30 °C ราก 25 - 28 °C ใบขยายตัว 31 °C ผสมพันธุ์ 30 - 33 °C และเมล็ดสุกแก่ 20 - 29 °C

4. อากาศ ในการเจริญเติบโตและพัฒนาการของพืช ต้องการพลังงานที่ได้มาจากการหายใจ จึงต้องมีอากาศอย่างเพียงพอ เพื่อให้การหายใจเกิดขึ้นได้อย่างเต็มที่นอกจากนี้ พืชยังต้องการก๊าซคาร์บอนไดออกไซด์เพื่อใช้ในการสังเคราะห์แสงด้วย

5. น้ำ เป็นส่วนประกอบที่สำคัญของสิ่งมีชีวิต น้ำทำหน้าที่ในการช่วยดูดแร่ธาตุอาหาร (Nutrients) ลำเลียงอาหาร (Photosynthates) ไปยังส่วนต่างๆ และช่วยในการลดอุณหภูมิภายในต้นพืช เกี่ยวข้องทั้ง คุณภาพน้ำ ปริมาณน้ำ ระดับน้ำ ซึ่งจะเกี่ยวข้องต่อเนื่องกับดิน ธาตุอาหาร ความชื้นในดินหรือน้ำ

6. อื่นๆ จากปัจจัยทั้งห้าข้อ ก็จะมีผลต่อเนื่องมาในเรื่องการจัดการฟาร์มหรือวิธีการ/ระบบการปลูกของเกษตรกร เช่น ระยะการปลูก วิธีการปลูก จำนวนต้นที่ปักดำตามช่วงที่ปลูกข้าว ความลึกในการปักดำ ขั้นตอนในการปลูก การดูแลรักษาแต่ละระยะ เป็นต้น

2.1.2 ภาษาที่ใช้ในการพัฒนาระบบ

2.1.2.1 ภาษาไพทอน Python (Python programming language) เป็นภาษาโปรแกรมระดับสูงแบบอินเทอร์พรีเตอร์ (Interpreter) ที่ถูกสร้างขึ้นโดย กิโด ฟานรอสซัม (Guido van Rossum) ในปี พ.ศ. 2533 และปัจจุบันมีการดูแลโดย มูลนิธิซอฟต์แวร์ไพทอน (Python Software Foundation - PSF) ภาษา Python สามารถใช้งานได้บนระบบปฏิบัติการทุกประเภท เช่น Unix, Linux, Windows NT, Windows 2000, Windows 95/98/ME, และ OS/X และได้รับการนำมาใช้ในหลายสาขางานโดยมีจุดเด่นหลายอย่างดังนี้

1. ภาษาสคริปต์ Python เป็นภาษาสคริปต์ที่ไม่ต้องคอมไพล์ก่อนการรัน ทำให้ง่ายต่อการเริ่มต้นและทดสอบโปรแกรม นิยมใช้ในงานด้านการดูแลระบบ (System administration) และมีการสนับสนุนภาษา Python ในระบบปฏิบัติการ Linux และ Unix อีกด้วย นอกจากนี้ยังสามารถเรียกใช้งาน Python ในระบบ Windows ผ่าน Windows Script Host ได้เช่นกัน

2. พัฒนา Web Application Python ถูกนำมาใช้ในการพัฒนาเว็บแอปพลิเคชันอย่างแพร่หลาย มี Framework สำหรับพัฒนาเว็บของ Python ที่ได้รับความนิยมอย่างมากคือ Django และ Flask

3. โครงสร้างแบบไม่ซับซ้อน โครงสร้างของ Python ไม่ซับซ้อน และใช้สัญลักษณ์แบ่งบล็อกของโปรแกรมโดยการย่อหน้า ทำให้โค้ด Python อ่านง่ายและลดจำนวนบรรทัดของโค้ด

4. การสนับสนุน Docstring Python สนับสนุนการเขียน Docstring ซึ่งเป็นข้อความสั้นๆ ที่ใช้ในการอธิบายการทำงานของฟังก์ชัน, คลาส, และโมดูล เพื่อให้โปรแกรมเมอร์เข้าใจโค้ดได้อย่างชัดเจน

5. ภาษาทาว (Glue Language) Python เป็นภาษาทาวที่สามารถเรียกใช้ภาษาโปรแกรมอื่นๆ ได้หลายภาษา ทำให้เหมาะในการประสานงานโปรแกรมที่เขียนด้วยภาษาต่างกัน

6. ไลบรารี Python มีชุดไลบรารีมาตรฐานที่มีให้ตั้งแต่ตอนติดตั้งอินเทอร์พรีเตอร์ ทำให้โปรแกรมเมอร์สามารถลดภาระในการเขียนโค้ดที่ซ้ำ ๆ เช่นการแสดงผลข้อมูลออกสู่หน้าจอและการรับค่า

7. ความเร็ว Python ทำงานได้อย่างเร็วและมีประสิทธิภาพเมื่อเทียบกับภาษาสคริปต์อื่น ๆ เช่น PHP, JSP, และ ASP Python เป็นภาษาโปรแกรมที่มีความยืดหยุ่น

และเหมาะสำหรับโปรแกรมเมอร์ทุกคนในหลายสาขางาน เริ่มต้นใช้งาน Python ไม่ยากเลย และมีชุดเครื่องมือที่มากมายที่ช่วยในการพัฒนาโปรแกรมอย่างมีประสิทธิภาพและรวดเร็ว

2.1.3 เครื่องมือที่ใช้ในการพัฒนา

2.1.3.1 YOLO (You Only Look Once) โมเดลการตรวจจับวัตถุแบบ YOLO เป็นวิธีการตรวจจับวัตถุที่มีความเร็วและมีประสิทธิภาพสูงในการตรวจจับวัตถุในรูปภาพหรือวิดีโอ โมเดล YOLO สามารถตรวจจับและระบุวัตถุที่ปรากฏในภาพได้พร้อมกันหลายวัตถุในหนึ่งกรอบสี่เหลี่ยม (bounding box) โดยไม่จำเป็นต้องทำขั้นตอนหลายขั้นตอนเหมือนวิธีการอื่น ๆ เช่น Sliding Window หรือ Region Proposal Networks (RPNs) ซึ่งทำให้ YOLO มีความเร็วในการทำนายที่สูงกว่าและเหมาะสำหรับการใช้งานแบบ real-time หรือในระบบที่ต้องการประมวลผลแบบเรียลไทม์อย่างรวดเร็ว เช่น รถยนต์ติดตามเชิงกลยุทธ์ (autonomous vehicles) หรือการตรวจจับวัตถุในวิดีโอสด (live video streaming)

ลักษณะหลักของ YOLO คือการส่งค่าขอ (inference) ในรูปภาพเดียวผ่านโมเดล เพื่อทำการตรวจจับวัตถุและระบุคลาสของวัตถุพร้อมกัน โดยทั้งกระบวนการตรวจจับและระบุคลาสจะเป็นการทำนายที่เกิดขึ้นในขั้นเดียว เรื่องนี้ต่างจากวิธีการอื่นที่อาจจะมีขั้นตอนต่าง ๆ แยกกัน ซึ่งทำให้ YOLO มีความเร็วสูง

อัลกอริทึมที่ใช้ใน YOLO เป็นอัลกอริทึมที่มีชื่อเรียกว่า "Darknet" ซึ่งเป็นอัลกอริทึมที่ถูกพัฒนาขึ้นโดยผู้สร้าง YOLO เอง อย่างไรก็ตาม การที่เรียก Darknet ว่า "อัลกอริทึม" นั้นเป็นเรื่องที่ค่อนข้างลำบาก เนื่องจาก Darknet แท้จริงเป็นเครื่องมือสำหรับการทดลองพัฒนาโมเดล Deep Learning ที่เขียนด้วยภาษา C และ CUDA (สำหรับคำนวณ GPU) ที่รวมอัลกอริทึมต่าง ๆ ที่เกี่ยวข้องกับการเรียนรู้เชิงลึก Darknet นั้นไม่ใช่เพียงแค่อัลกอริทึมเดียว แต่เป็นเครื่องมือที่ใช้สำหรับการสร้างและฝึกโมเดล Deep Learning ต่าง ๆ รวมถึงโมเดลการตรวจจับวัตถุ YOLO ด้วย Darknet ยังรวมเป็นต้นเรื่อง โดยมี Darknet ที่เป็นโครงการที่ใหญ่กว่า และใช้สำหรับการทดลองพัฒนาและทดสอบแนวคิดในการประมวลผลรูปภาพด้วยโมเดล Deep Learning ต่าง ๆ

2.1.3.2 Google Colab Notebooks เป็นแพลตฟอร์มที่ให้บริการในการเขียนและเรียกใช้งานโน้ตบุ๊ก (Notebooks) แบบออนไลน์โดยใช้เครื่องมือและพื้นที่ที่เกิดจาก Google ที่ไม่ต้องติดตั้งหรือกำหนดค่าใด ๆ มาก่อน มันถูกออกแบบมาเพื่อเป็นส่วนหนึ่งในนักพัฒนาและนักเรียนที่ต้องการทดลอง พัฒนา หรือแบ่งปันโค้ดของพวกเขาโดยไม่จำเป็นต้องมีเครื่องมือหรือโปรแกรมที่ซับซ้อนในการตั้งค่าหรือบำรุงรักษาสภาพแวดล้อมการพัฒนา

โน้ตบุ๊กใน Google Colab ใช้เพื่อเขียนและเรียกใช้งานโค้ด Python ได้แบบออนไลน์ โดยใช้สภาพแวดล้อมที่มีอินเทอร์เน็ตที่พร้อมสำหรับการเรียกใช้งานหน่วยประมวลผลกราฟิก (GPU) และหน่วยประมวลผลเทนเซอร์ (TPU) ซึ่งช่วยให้ความเร็วในการประมวลผลข้อมูลที่มากขึ้น โน้ตบุ๊กใน Colab สามารถใช้ในการทำงานที่เกี่ยวข้องกับการเรียนรู้เชิงลึก (Deep Learning) การวิเคราะห์ข้อมูล การสร้างแผนภูมิ และงานความเชี่ยวชาญทางวิทยาการข้อมูลอื่น ๆ

2.1.3.3 Roboflow Roboflow เป็นแพลตฟอร์มที่มุ่งเน้นการทำให้กระบวนการสร้างและปรับปรุงโมเดลการเรียนรู้เชิงลึก (Deep Learning) สำหรับการประมวลผลภาพและการมองเห็นที่ง่ายและมีประสิทธิภาพ โดยใช้ข้อมูลที่อยู่ในรูปภาพเพื่อสร้างโมเดลที่สามารถระบุวัตถุหรือคุณสมบัติต่าง ๆ ในภาพได้ คุณสมบัติและฟังก์ชันหลักของ Roboflow ประกอบด้วย

1. การเตรียมข้อมูล Roboflow ช่วยในกระบวนการเตรียมข้อมูลภาพ โดยรองรับหลายรูปแบบไฟล์ภาพ รวมถึงการแปลงรูปภาพเป็นรูปแบบที่เหมาะสมสำหรับ

2. การฝึกโมเดล เช่นการปรับขนาดรูปภาพ การตัดต่อภาพ การเพิ่มสีพื้นหลัง เป็นต้น

3. การสร้างและปรับปรุงโมเดล Roboflow มีเครื่องมือสำหรับสร้างและปรับปรุงโมเดลการเรียนรู้เชิงลึก รวมถึงการเลือกแบบจำลองที่เหมาะสมสำหรับงานของคุณ เช่น YOLO, TensorFlow, Fastai, และอื่น ๆ

4. การสร้างและจัดการกับข้อมูลเทรนนิ่ง คุณสามารถแบ่งข้อมูลเป็นชุดเทรนนิ่ง (training sets) และชุดทดสอบ (testing sets) สำหรับการประเมินและการทดสอบความสามารถของโมเดล

5. การตรวจสอบและประเมินโมเดล Roboflow ช่วยในกระบวนการตรวจสอบและประเมินความแม่นยำของโมเดลในขั้นตอนการทดสอบ โดยแสดงผลลัพธ์ในรูปแบบกราฟและตาราง

6. การใช้งานง่ายและอินเทอร์เฟซกราฟิก Roboflow มีอินเทอร์เฟซกราฟิกที่ใช้งานง่าย ทำให้ผู้ใช้สามารถเข้าถึงฟังก์ชันและการตั้งค่าต่าง ๆ ได้โดยไม่ยากลำบาก

7. การแบ่งปันและการทำงานร่วมกับทีม สามารถแบ่งปันโมเดลและโค้ดของคุณกับทีมหรือผู้อื่นได้ ทำให้งานร่วมกันเป็นเรื่องสะดวกและมีประสิทธิภาพ รวมถึงคุณสมบัติอื่น ๆ อีกมากมาย โดย Roboflow เป็นเครื่องมือที่ช่วยให้นักพัฒนาและนักวิจัยสามารถ สร้าง และ ปรับปรุง โมเดล การ เรียน รู้ เชิง ลึก สำหรับ การ ประมวล ผล ภาพ อย่าง มี ประสิทธิภาพและสะดวกในการใช้งาน

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 การประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกสำหรับการรู้จำโรคขอบใบแห้งและโรคไหม้ของข้าว (เตชินท์ วรสิทธิ์ et al., 2022)

ข้าวเป็นพืชเศรษฐกิจที่สำคัญของประเทศไทย ปัจจัยที่ทำให้ข้าวมีคุณภาพและผลผลิตลดลงส่วนหนึ่งมาจากการแพร่ระบาดของโรค โดยโรคที่เป็นสาเหตุอันดับต้นๆ ได้แก่ โรคขอบใบแห้ง หากมีการวินิจฉัยโรคข้าวได้ถูกต้องและรวดเร็วก็จะสามารถลดความเสียหายของผลผลิตได้ ผลการทดสอบพบว่าเทคนิคการเรียนรู้เชิงลึก YOLOv3 มีประสิทธิภาพสูงสุดในการรู้จำโรคทั้งสอง ซึ่งมีค่าเฉลี่ยความถูกต้องในการจำแนกของโรคขอบใบแห้งและโรคไหม้อยู่ที่ 90.33% และ 86.46%

2.2.2 การพัฒนาขั้นตอนวิธีสำหรับจำแนกโรคอ้อยที่พบมากในประเทศไทยสำหรับการจำแนกโรคอ้อยที่พบมากบนพันธุ์ขอนแก่น 3 (ณัฐภักดิ์ ลาวัลย์วงศ์, 2021)

การจำแนกโรคอ้อยที่พบมากบนพันธุ์ขอนแก่น 3 ซึ่งเป็นพันธุ์อ้อยที่ได้รับความนิยมมากที่สุดในประเทศไทย จะให้ความสำคัญในการวิเคราะห์โรคที่ส่งผลกระทบต่ออ้อย เช่น โรคที่เกิดจากแบคทีเรีย เชื้อรา และแมลง จากการตรวจหาและระบุโรคที่เกิดขึ้นในอ้อยซึ่งต้องการผู้เชี่ยวชาญที่มีประสบการณ์และเวลา จึงมีแนวคิดในการสร้างระบบที่ช่วยในการจำแนกโรคอ้อยที่มีความแม่นยำอย่างอัตโนมัติ ซึ่งส่งผลให้การวินิจฉัยโรคทำได้รวดเร็วมากขึ้น เป็นระบบอัจฉริยะ สำหรับ จำแนก ประเภท และ ลักษณะ อาการ ของ โรค อ้อย (Intelligent System Diagnosis Sugarcane Diseases with Deep Convolutional Neural Network) เป็นระบบที่ให้ผู้ใช้งานอัปโหลดรูปภาพใบอ้อย สามารถบ่งบอกโรคอ้อยและระบุสาเหตุพร้อมวิธีการป้องกันหรือควบคุม โดยระบบมีความสามารถในการจำแนกชนิดของโรคอ้อย 5 คลาส ได้แก่ โรคเส้นกลางใบแดง (Red Rot) โรคราสนิม (Rust) โรคใบจุดวงแหวน (Ring Spot) โรคใบขาว (White Leaf) และใบสมบูรณ์ (Normal) ที่มีค่า Mean Average Precision (mAP) สูงถึง 0.8681 หรือร้อยละ 86.81

2.2.3 คัดแยกขยะรีไซเคิลด้วยการประมวลผลภาพ(ธนโชติ ภาชนะนัย, จักรกริช ปานเงิน, กรรณิการ์ คนงาม, วรชัย ศรีสมุดคำ, & วาสนา วงศ์ษา, 2022)

ให้สามารถควบคุมการคัดแยกขยะรีไซเคิล 4 ประเภทได้แก่ กระป๋อง ขวดแก้ว กล่องกระดาษ และขวดพลาสติก พร้อมทำการแจ้งเตือนผ่านแอปพลิเคชันไลน์เมื่อถึงขยะใกล้เต็ม

ระบบ ประกอบด้วย กล้อง IMILAB Webcam รับ ภาพ วัตถุ แบบ เร็ยล ไทม์ เชื่อม ต่อ กับ Raspberry Pi4 Model B เพื่อทำการประมวลผลการจำแนกขยะด้วยโมเดลของ YOLOv5 ที่ผ่านการฝึกสอนด้วยชุดข้อมูลภาพ 1,321 ภาพ 250 รอบ มีค่า mAP50% 0.995, Recall 0.995, Precision 0.999 และค่า F-measure 0.997 โดยเชื่อมต่อกับ Module L298N ในการควบคุมตำแหน่งการหมุนของสเต็ปมอเตอร์ไปยังถังขยะเอาต์พุตและควบคุมเซอร์โวมอเตอร์ในการเทขยะลงถังขยะแต่ละประเภท ผลการทดสอบประสิทธิภาพการทำงานพบว่าสามารถคัดแยกขยะได้ถูกต้องตามประเภทคิดเป็น 98.33% ผลลัพธ์ค่าช่วงความเชื่อมั่น (Confidence interval) อยู่ในช่วง 0.50 - 0.97 ผลทดสอบประสิทธิภาพการทำงานของฮาร์ดแวร์ในการแยกขยะไปยังถังขยะเอาต์พุตแต่ละประเภทด้วยสเต็ปมอเตอร์และเซอร์โวมอเตอร์ และการแจ้งเตือนเมื่อถังขยะใกล้เต็มถึงผ่านแอปพลิเคชันไลน์สามารถทำงานได้ถูกต้อง 100%

2.2.4 การตรวจหาช่องจอดรถแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียมYOLO(จันทนะชาติ et al., n.d.)

ระบบการตรวจหาช่องจอดรถแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียม YOLO โดยนำอัลกอริทึม Tiny YOLO มาประยุกต์ การตรวจหารถยนต์จากภาพต่อเนื่องโดยการจับภาพทีละเฟรมและคำนวณหาวัตถุทั้งหมดบนเฟรมภาพนิ่งแล้วนำวัตถุทั้งหมดมาทำนายด้วยโครงข่ายประสาทเทียมมาตรวจหารถยนต์จากภาพต่อเนื่องโดยการจับภาพทีละเฟรมและคำนวณหาวัตถุทั้งหมดบนเฟรมภาพนิ่งแล้วนำวัตถุทั้งหมดมาทำนายด้วยโครงข่ายประสาทเทียม คำนวณพื้นที่ทับซ้อนของวัตถุรถยนต์และพื้นที่ของแต่ละช่องจอดรถถ้าพื้นที่ทับซ้อนกันมากกว่าร้อยละ 50 แสดงว่าช่องจอดนั้นไม่ว่างถ้าน้อยกว่าแสดงว่าไม่ว่างผลการทดสอบระบบโดยใช้ภาพตัวอย่างจากโรงจอดรถจริงจำนวน 17 เฟรม โดยแต่ละเฟรมมีรถยนต์เข้าจอดในแต่ละช่องจอดไม่เหมือนกัน พบว่า ระบบสามารถตรวจหาช่องจอดรถและแสดงผลมีค่าร้อยละความถูกต้อง 67.13 และมีค่าเฉลี่ยความคลาดเคลื่อนร้อยละสัมบูรณ์ 32.9

2.2.5 การสร้างระบบตรวจนับบุคคลแบบเวลาจริงราคาประหยัดบนRaspberry Pi โดยประยุกต์อัลกอริทึม Tiny YOLO V3(Punyato & Sidahao, 2019)

การสร้างระบบตรวจนับบุคคลแบบเวลาจริงราคาประหยัดบนRaspberry Pi ด้วยการประยุกต์ใช้อัลกอริทึมTiny YOLO V3ซึ่งเป็นอัลกอริทึมการตรวจจับวัตถุ ที่มีความเร็วและความถูกต้องแม่นยำสูง สถาปัตยกรรมของTiny YOLO v3 มีความซับซ้อนน้อยกว่าสถาปัตยกรรมYOLO แบบสมบูรณ์มาก และเนื่องจากอัลกอริทึมTiny YOLO มีโมเดลพื้นฐานที่ถูกฝึก (train) สำหรับใช้ในการตรวจจับวัตถุทั่วไปที่สามารถพบได้ในชีวิตประจำวันเช่น คน

สุนัขแมว โตะ แก้ว รถ ได้นำโมเดลดังกล่าวมาประยุกต์ใช้ในการรู้จำแล้วนับจำนวนบุคคลโดยกำหนดขอบเขตของสภาวะแวดล้อมจะเป็นบริเวณเฉพาะที่คนเดินผ่านเท่านั้น

2.2.6 การตรวจจับสิ่งกีดขวางสำหรับแก้อั้วรถเซ็นไฟฟ้าด้วยคอมพิวเตอร์วิทัศน์(พัฒน จิตรศิลป์ & เพ็ญพิชชา, 2021)

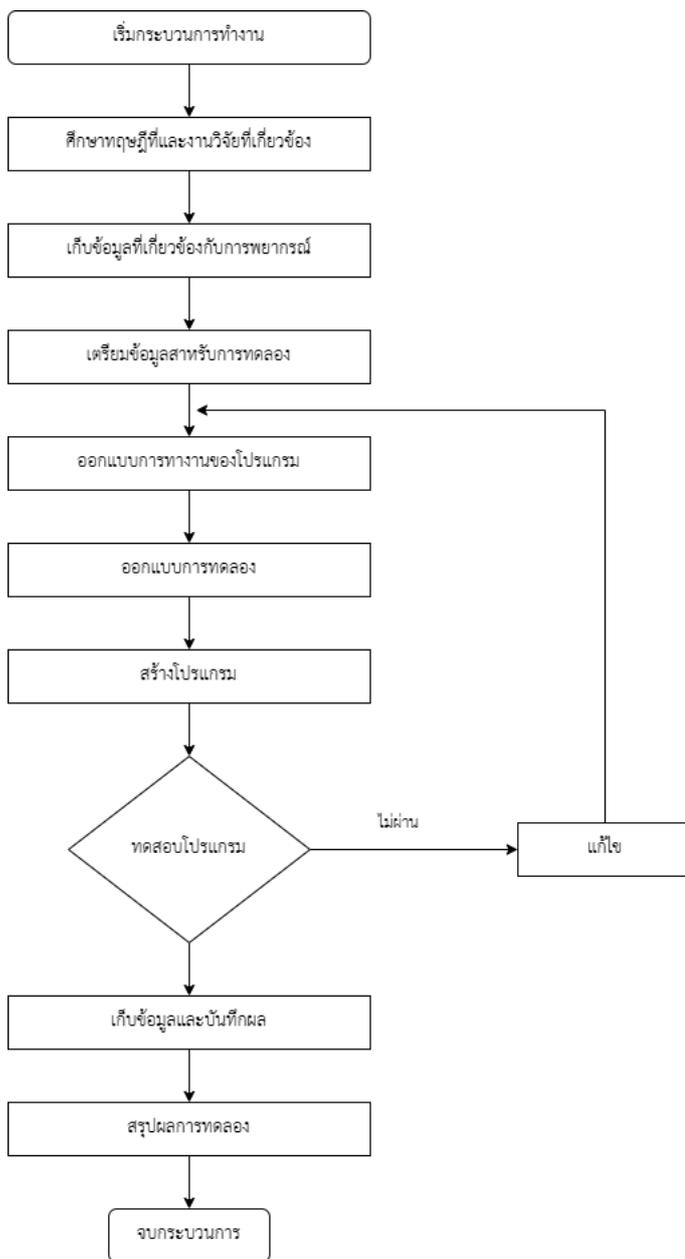
วรวิทธิ์ ได้นำเสนอบทความ การตรวจจับสิ่งกีดขวางสำหรับแก้อั้วรถเซ็นไฟฟ้าด้วยคอมพิวเตอร์วิทัศน์ งานวิจัยนี้นำเสนอระบบตรวจจับสิ่งกีดขวางขณะถอยหลังสำหรับแก้อั้วรถเซ็นไฟฟ้า เพื่ออำนวยความสะดวกให้แก่ผู้พิการ และใช้อัลกอริทึม YOLOv3 ในการตรวจจับสิ่งกีดขวางประเภทวัตถุ ในงานวิจัยนี้จึงได้มีการพัฒนาอัลกอริทึมที่ใช้ในการตรวจจับสิ่งกีดขวางที่เป็นเสา กำแพง หรือ ประตู เพิ่มเติมขึ้นมาด้วยวิธีการตรวจจับขอบภาพ ระบบจึงมีการเลือกใช้ อัลกอริทึมในการตรวจจับสิ่งกีดขวางระหว่างวัตถุกับเส้นขอบ ซึ่งผลการทดลองพบว่า ระบบสามารถเลือกใช้อัลกอริทึมในการตรวจจับสิ่งกีดขวางได้ถูกต้องถึง 80% และ ทำการแจ้งเตือนก่อนชนได้แม่นยำสูงถึง 90%

บทที่ 3

วิธีดำเนินการวิจัย

การดำเนินงานวิจัยผู้วิจัยได้ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องที่มีความจำเป็นต่องานวิจัย โดยนำแนวคิด และปัญหา มาพัฒนาสร้างการจำแนกโรคข้าวให้มีประสิทธิภาพและการทำงานที่แม่นยำมากขึ้น โดยใช้เทคนิคโครงข่ายประสาทเทียม โดยเริ่มต้นจากการศึกษา ที่มีอัลกอริทึมเหมาะสมกับการสร้างโมเดลโดยใช้เทคนิคโครงข่ายประสาทเทียม ผู้วิจัยได้นำ YOLO (You Only Look Once) มาใช้ ในการออกแบบและการสร้างการจำแนกโรคข้าว และภาษาที่ใช้ในการเขียนโปรแกรมเช่น Python ซึ่งจำเป็นต่องานวิจัยอย่างมาก ในการพัฒนาระบบนี้ผู้วิจัยได้ใช้ Visual Studio Code IDE บนระบบปฏิบัติการ Windows และใช้ roboflow ในการเก็บข้อมูลโรค และใช้ Google Colab (Google Colaboratory) ในการให้เทรนเพื่อให้ได้โมเดลที่ใช้ในการพยากรณ์ให้มีประสิทธิภาพและรวดเร็ว เพื่อนำมาพยากรณ์โดยใช้เทคนิคโครงข่ายประสาทเทียม และสร้างระบบตรวจจับโรคข้าว

ในการดำเนินงานวิจัยนี้ ผู้วิจัยได้สร้าง โปรแกรม สำหรับ การ ทำงาน เพื่อให้ได้ประสิทธิภาพ สูงสุดในการดำเนินงานวิจัย ซึ่งผู้วิจัยได้แบ่งขั้นตอน เพื่อศึกษาหาวิธีการที่เหมาะสมกับการจำแนกโรคข้าว โดยมีขั้นตอน แสดงในแผนผังการดำเนินงานวิจัยดังภาพ ที่ 3.1 ซึ่งแสดงแผนผังการดำเนินงานวิจัยการจำแนกจับโรคข้าว โดยใช้เทคนิคโครงข่ายประสาทเทียม จากแผนผังการดำเนินงานวิจัยผู้วิจัยได้เริ่มจากการศึกษาเรื่องแบบจำลองโครงข่ายประสาทเทียม ประกอบกับการศึกษางานวิจัยที่เกี่ยวข้องอื่น ๆ ทำให้ผู้วิจัยทราบว่าแบบจำลองโครงข่ายประสาทเทียมสามารถจำแนกและตรวจจับโรคของข้าวได้ ผู้วิจัยได้สร้าง ชุดข้อมูล (Dataset) ที่ใช้ในการพยากรณ์ โดยใช้ข้อมูลภาพจาก Web kaggle และใช้ Web Application Roboflow สำหรับใช้ในการ Label ภาพและกำหนด Class ของโรคข้าวที่ไว้ให้ระบบตรวจจับและจำแนกโรคได้ และRoboflowยังแบ่งชุดข้อมูลไว้ใช้ในการทดลอง ในการทดสอบโมเดลผู้วิจัยเลือกใช้ การทดสอบโมเดล ที่แยกชุดข้อมูลไว้มาทดสอบ โดยนำโมเดลของYolov5 มาทดสอบและใช้ในการเลือกโมเดลสำหรับใช้งานในระบบ โดยผู้วิจัยได้ใช้ Visual Studio Code ในการพัฒนาโปรแกรมเพื่อทดสอบและเลือกใช้โมเดลสำหรับพัฒนาระบบ โดยเลือก



ภาพที่ 3.1 แผนผังการดำเนินงานวิจัยการจำแนกโรคข้าว

3.1 เครื่องมือที่ใช้ในงานวิจัย

การตรวจจับโรคของข้าว อาศัยเครื่องมือและซอฟต์แวร์ดังต่อไปนี้

3.1.1 ซอฟต์แวร์ และเครื่องมือทางสถิติ

- Python ใช้เป็นภาษาโปรแกรมในการพัฒนา
- Visual Studio Code ใช้สำหรับเป็นเครื่องมือ IDE ที่ช่วยเขียนภาษา Python
- Roboflow ใช้ตัดป้ายชื่อโรคแล้วแบ่งชุดข้อมูล ใน Dataset
- GoogleColab เป็นแพลตฟอร์มที่ให้บริการรันโค้ดPythonบนคลาวด์(cloud)
- YOLO(You Only Look Once) โมเดลการตรวจจับวัตถุแบบความเร็วและมี

ประสิทธิภาพสูงในการตรวจจับวัตถุในรูปภาพหรือวิดีโอ

3.1.2 ฮาร์ดแวร์

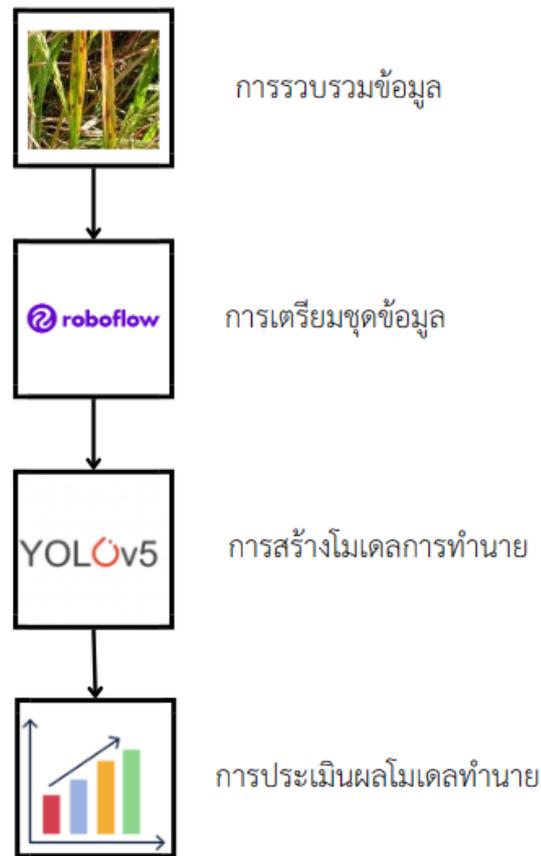
- เครื่องคอมพิวเตอร์

3.2 กรอบแนวคิดภาพรวมของระบบ

กรอบแนวความคิดภาพรวมของการจำแนกโรคข้าว จากการที่ผู้วิจัยได้วางแผนการดำเนินงานผู้วิจัยได้ออกแบบภาพรวมของกรจำแนกโรคข้าว โดยมีกรอบแนวคิดในการออกแบบ ดังนี้

จากภาพที่ 3.2 ระบบโดยรวมของการจำแนกโรคข้าว มีหลักการทำงานที่ส่วนคือ

1. รวบรวมภาพโรคข้าวมาจัดเป็นกลุ่มและแยกโรคโดยใช้ภาพที่มีความชัดที่สุด
2. จัดเตรียมภาพแล้วอัปโหลดขึ้นRoboflow เพื่อจะเตรียมสร้างกรอบ และ ตัดป้ายชื่อโรคของข้าว แบ่งชุดข้อมูลเป็น 3 ส่วนคือ ชุดเรียนรู้ ชุดช่วยเรียนรู้ และชุดทดสอบ
3. นำชุดข้อมูล มาเทรนให้โมเดลYOLO เรียนรู้ เพื่อให้ได้โมเดลทำนาย
4. นำโมเดลทำนาย มาจำแนกโรคข้าวเปรียบเทียบประสิทธิภาพด้านความแม่นยำ และประสิทธิภาพด้านเวลา



ภาพที่ 3.2 แผนผังแสดงภาพรวมของการจำแนกโรคข้าว

3.3 การรวบรวมข้อมูล

การเลือกรูปที่ใช้ในการทำชุดข้อมูล ในขั้นตอนนี้ได้เลือกรูป ที่รวบรวมจากภาพของใบข้าว จาก เว็บไซต์ Kaggle ได้ ภาพ ทั้งหมด 1200 ภาพ โดย แบ่ง เป็น รูป ของ โรค จุด ใบน้ำตาล(BrownSpot) 400 ภาพ โรค ใบไหม้(LeafBlast) 400 ภาพ และ โรค แมลงดำหนาม(Hispa) 400 ภาพ โดยที่การเลือกภาพนั้นเราจะเลือกภาพที่ต่างกันและชัดให้มากที่สุด เพื่อระบบมีการทำงานที่มีประสิทธิภาพ

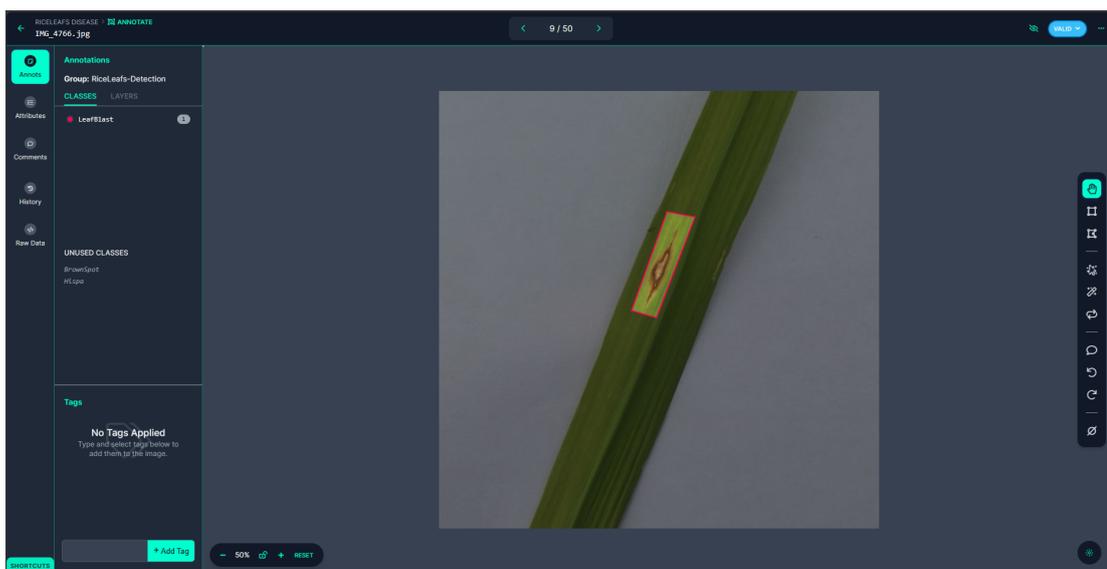
3.4 การเตรียมข้อมูลสำหรับการสร้างโมเดลการพยากรณ์

การสร้างชุดข้อมูลโดยการนำภาพที่เลือกมาจำนวน1200ภาพ ใส่ป้ายชื่อโรคบนภาพใบข้าวโดยใช้แพลตฟอร์มโรโบโพล์ ซึ่งกำหนดป้ายชื่อคือโรคใบจุดสีน้ำตาล,โรคแมลงดำหนาม และโรคใบไหม้

3.4.1 การเตรียมชุดข้อมูล

เป็นการสร้างชุดข้อมูลที่ให้โมเดลทำนายเรียนรู้แบ่งเป็นสองขั้นตอนดังนี้

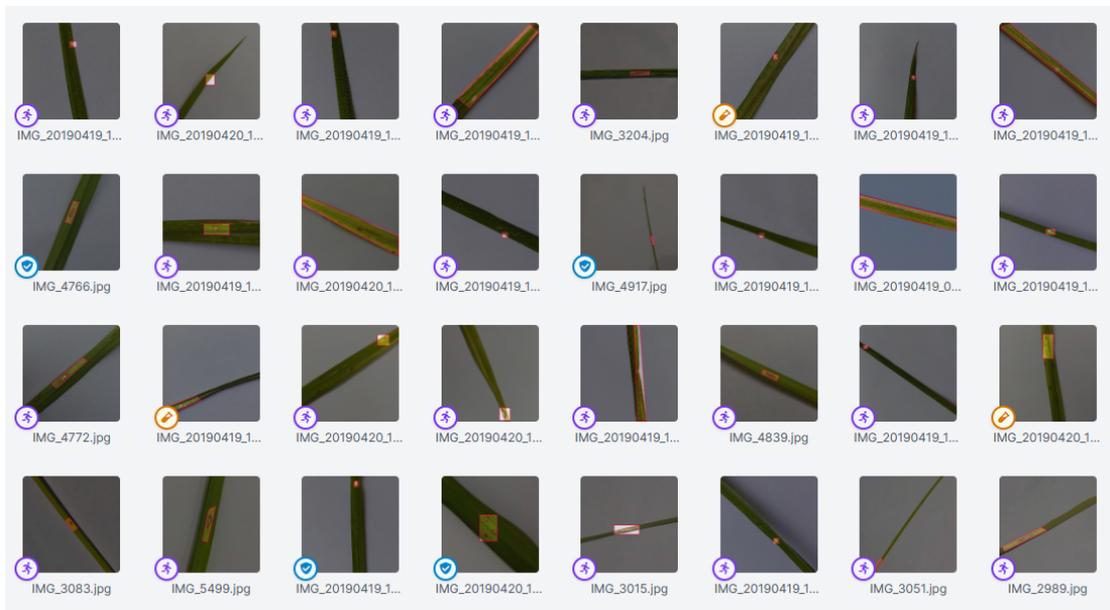
3.4.1.1 สร้างชุดข้อมูลโดยใช้แพลตฟอร์มโรโบโพล์ ผู้วิจัยได้ใช้เครื่องมือบนแพลตฟอร์มโรโบโพล์ ในการสร้างกรอบติดป้ายชื่อโรคบนภาพ โดยที่สร้างกรอบป้ายชื่อบนภาพทั้งหมดในชุดข้อมูล ดังแสดงภาพที่ 3.3



ภาพที่ 3.3 การสร้างกรอบป้ายชื่อในการจำแนกโรค

3.4.1.2 แบ่งชุดข้อมูลโดยใช้แพลตฟอร์มโรโบโพล์ เมื่อได้ชุดข้อมูลแล้ว ตัวแพลตฟอร์มโรโบโพล์ จะมีฟังก์ชันในการเพิ่มภาพโดยการกลับภาพ หมุนซ้ายหมุนขวา เพื่อเพิ่มประสิทธิภาพของชุดข้อมูล ดังแสดงตัวอย่างภาพข้อมูลภาพที่ 3.4 และทำการแบ่งชุด

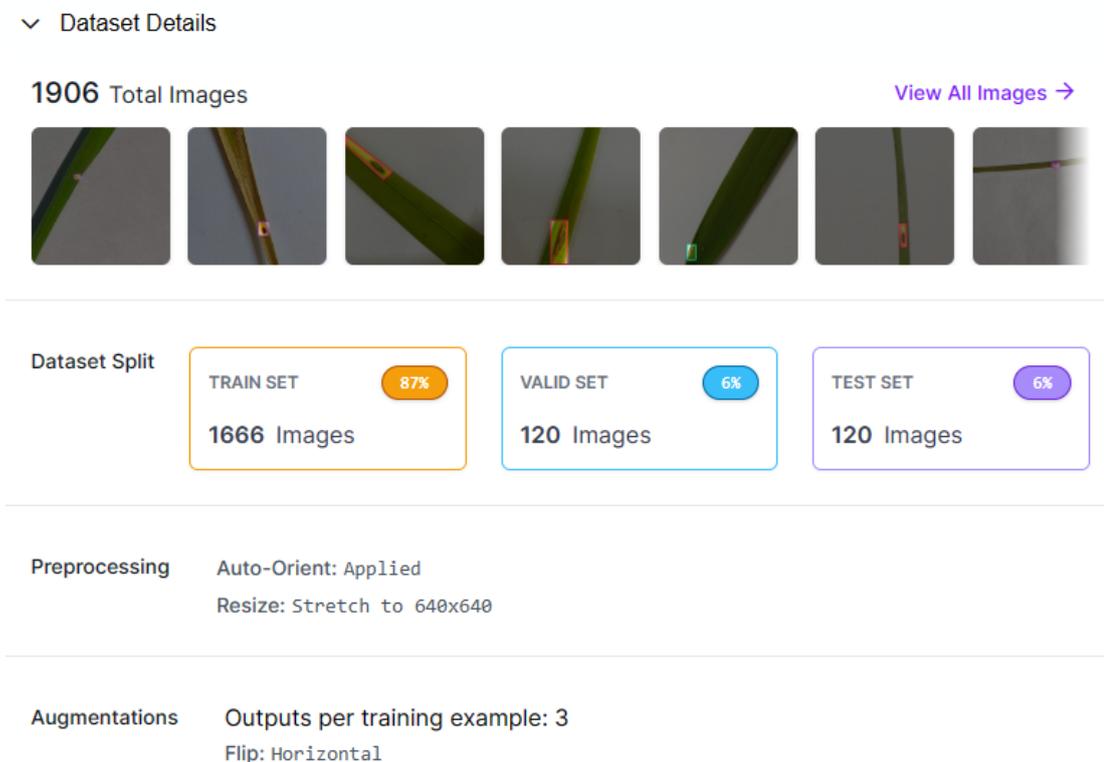
ข้อมูลออกเป็นสามส่วน คือชุดเรียนเป็นจำนวน 1666 ภาพ, ชุดช่วยเรียนรู้เป็นจำนวน 120 ภาพ และชุดทดสอบ เป็นจำนวน 120 ภาพ ดังแสดงภาพที่ 3.5



ภาพที่ 3.4 ภาพชุดข้อมูล

3.5 การสร้างโมเดลการทำนาย

การสร้างโมเดลการทำนาย ผู้วิจัย โมเดลYOLOv5 มาสร้างโมเดลทำนาย โดยใช้ YOLOv5s, YOLOv5m, YOLOv5l และYOLOv5x นำเอาชุดข้อมูลเรียนรู้ และชุดช่วยเรียนรู้ ดังที่กล่าวใน 3.4.1.2 ให้แต่ละโมเดลกำหนดการตั้งค่าพารามิเตอร์ (Parameter) ให้แต่ละโมเดลทำนายโดยมีขนาดภาพ, batch, epochs ดังแสดงในตารางที่ 3.1 หลังจากการเรียนรู้ทุกโมเดลแล้ว ให้แต่ละโมเดลของYOLOv5 บันทึกและใช้โมเดลทำนายรอบที่มีความถูกต้องและผลลัพธ์ที่ดีที่สุด



ภาพที่ 3.5 การแบ่งชุดข้อมูล

3.5.1 การตรวจจับวัตถุแบบหนึ่งขั้นตอนของYOLO

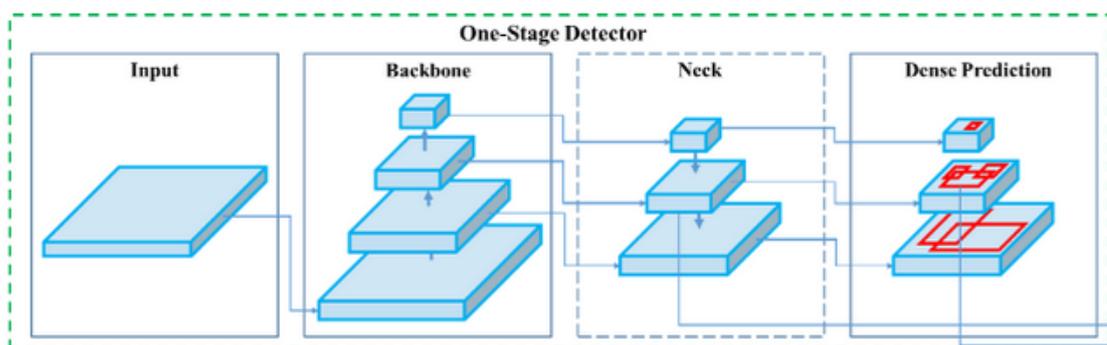
YOLO คือโมเดลที่ใช้วิธีการตรวจจับวัตถุโดยใช้ขั้นตอนเดียว ดังแสดงภาพที่ 3.6 โดยมีลักษณะดังนี้

1. Backbone ส่วนนี้ใช้เครือข่ายที่ถูกละเอียดเพื่อสกัดคุณลักษณะของภาพ เพื่อให้เกิดการลดขนาดของภาพและเพิ่มคุณลักษณะของภาพ เพื่อให้โมเดลเข้าใจภาพได้อย่างมีประสิทธิภาพ
2. Neck ส่วนนี้ใช้ในการสกัดปิรามิดคุณลักษณะ (feature pyramids) เพื่อช่วยให้โมเดลสามารถทำนายวัตถุขนาดต่าง ๆ ได้อย่างแม่นยำ นั่นคือการทำนายวัตถุที่มีขนาดต่างกันภายในภาพ
3. Head ส่วนนี้ใช้ในการทำขั้นตอนสุดท้าย โดยการนำกล่องสแกน (anchor boxes) มาใช้บนแผนผังคุณลักษณะ (feature maps) และสร้างผลลัพธ์สุดท้ายที่ประกอบด้วยคลาส (classes), คะแนนความเป็นวัตถุ (objectness scores) และกล่องสี่เหลี่ยมรอบวัตถุ

ตารางที่ 3.1 การตั้งค่าพารามิเตอร์ให้โมเดล YOLO

model	Image Size	Batch Size	Number of Epochs	Model depth multiple	Lyer channel multiple
YOLOv5s	640x640	16	200	0.33	0.5
YOLOv5m	640x640	16	200	0.67	0.75
YOLOv5l	640x640	16	200	1	1
YOLOv5x	640x640	16	200	1.33	1.25

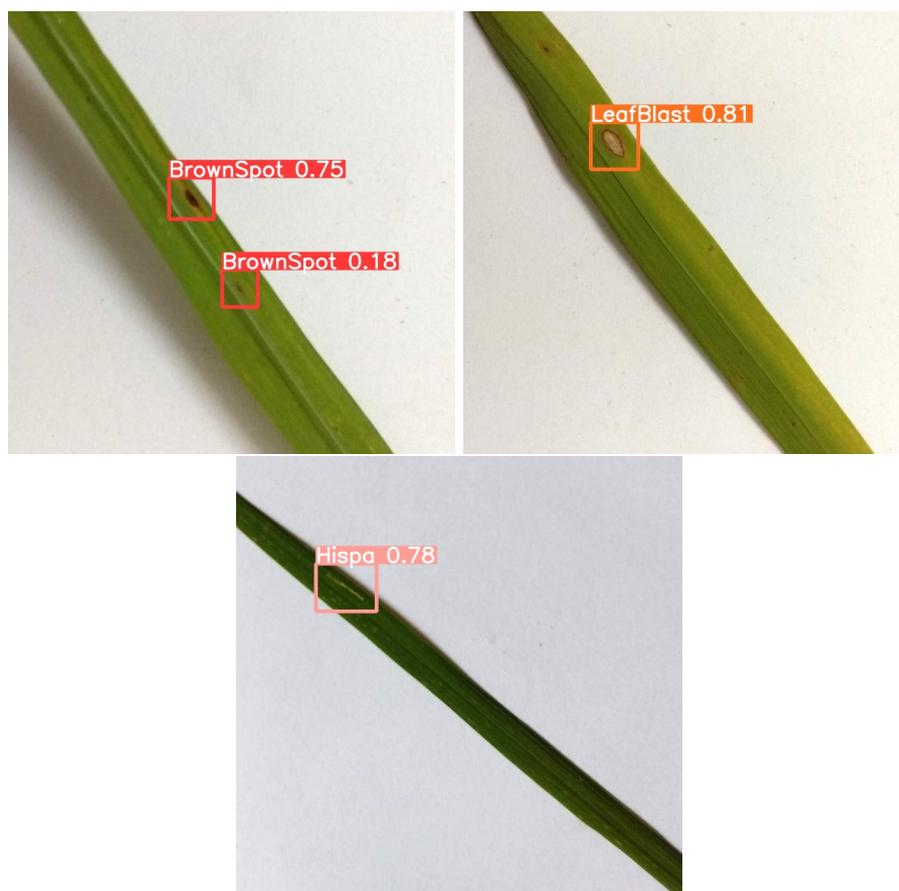
(bounding boxes) เพื่อระบุว่าแต่ละส่วนของภาพมีวัตถุอะไรบ้างและระบุตำแหน่งของวัตถุในภาพได้อย่างแม่นยำ



ภาพที่ 3.6 การตรวจจับวัตถุแบบหนึ่งขั้นตอน
ที่มา: <https://iq.opengenus.org/yolov5/>

3.6 การทดสอบการจำแนกโรคข้าว

ผู้วิจัยได้ทดสอบจำแนกโรคข้าวของแต่ละโมเดลที่สร้างเสร็จแล้ว ถ้าหากว่าภาพใบข้าวมีโรค YOLO จะสร้างกรอบสี่เหลี่ยมไว้ตรงจุดที่พบโรคบนภาพ โดยที่จะบอกชื่อโรคและความแม่นยำ ดังแสดงในภาพที่ 3.7



ภาพที่ 3.7 ตัวอย่างผลลัพธ์การจำแนกโรค

3.7 การออกแบบการทดลอง

จากการที่ผู้วิจัยได้ทำการสร้างโมเดลและทดสอบการจำแนกโรค ผู้วิจัยได้นำตาราง ConfusionMatrix ดังกล่าวในภาพที่ 3.8 มาใช้ในการทดสอบโมเดลทำนาย เมื่อได้ตาราง ConfusionMatrix ใช้สมการที่ 3.1 ในการหาค่าความแม่นยำ ส่วนของการทดลองผู้วิจัยได้ออกแบบการทดลองออกเป็นสองส่วนคือ การทดลองโมเดลประสิทธิภาพด้านความแม่นยำและประสิทธิภาพด้านเวลา

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

โดยที่

TP คือจำนวนที่ถูก กรณีที่ระบบทำนายว่าเหตุการณ์หรือข้อมูลเป็นจริง และตรงกับความจริงที่ว่าเป็นจริง ในกรณีนี้ระบบทำนายถูกต้อง

TN คือจำนวนที่ถูก กรณีที่ระบบทำนายว่าเหตุการณ์หรือข้อมูลไม่เป็นจริง และตรงกับความจริงที่ว่าเป็นจริง ในกรณีนี้ระบบทำนายถูกต้อง

FP คือจำนวนที่ผิด กรณีที่ระบบทำนายว่าเหตุการณ์หรือข้อมูลเป็นจริง แต่ความจริงแล้วไม่เป็นจริง ในกรณีนี้ระบบทำนายผิด

FN คือเป็นจำนวนที่ผิด กรณีที่ระบบทำนายว่าเหตุการณ์หรือข้อมูลไม่เป็นจริง แต่ความจริงแล้วเป็นจริง ในกรณีนี้ระบบทำนายผิด

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

ภาพที่ 3.8 ConfusionMatrix

ที่มา: <https://www.shorturl.asia/4dXLQ>

บทที่ 4

ผลการดำเนินงานวิจัย

จากการออกแบบและพัฒนาการจำแนกโรคใบขาวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO จะกล่าวถึงผลการทดลองและบันทึกผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของโมเดล โดยมีแนวคิดในการนำโครงข่ายประสาทเทียมคอนโวลูชัน YOLO ได้แก่ YOLOv5s, YOLOv5m, YOLOv5l และYOLOv5x มาใช้ในการสร้างโมเดลและเปรียบเทียบประสิทธิภาพ โดยใช้ตาราง ConfusionMatrix และชุดข้อมูลทดสอบ ดังที่กล่าวใน 3.4.1.2 เพื่อได้โมเดลทำนายที่ดีที่สุดในการใช้งาน ผลการทดลองเปรียบเทียบประสิทธิภาพเป็นสองส่วนได้แก่ ส่วนที่หนึ่ง ประสิทธิภาพด้านความแม่นยำ ส่วนที่สอง ประสิทธิภาพด้านเวลา โดยหาโมเดลทำนายที่มีความแม่นยำมากที่สุดและตั้งค่าพารามิเตอร์ ดังแสดงตารางที่ 3.1

4.1 ผลการทดลองประสิทธิภาพด้านเวลาของโมเดลทำนาย

ผลการทดลองเวลาในการสร้างโมเดลทำนายของYOLOv5 ได้แก่ YOLOv5s, YOLOv5m, YOLOv5l และYOLOv5x โดยจะเก็บเวลาในการเรียนรู้ของแต่ละโมเดลทำนาย ตั้งแต่เริ่มจนกระทั่งสิ้นสุดในการสร้างโมเดลทำนายของแต่ละตัว ดังที่แสดงตาราง

ตารางที่ 4.1 ผลเวลาการสร้างโมเดลทำนายของ YOLO

ชื่อโมเดลทำนาย	เวลาที่ใช้ในการสร้างโมเดล
YOLOv5s	1 ชั่วโมง 33 นาที
YOLOv5m	2 ชั่วโมง 17 นาที
YOLOv5l	3 ชั่วโมง 42 นาที
YOLOv5x	6 ชั่วโมง 29 นาที

4.2 ผล การ ทดลอง ประสิทธิภาพ ด้าน ความ แม่นยำ ของ โมเดล ทำนาย โดยใช้ ตาราง ConfusionMatrix

ผล การ ทดลอง โมเดล โดยใช้ เทคนิค โครง ConfusionMatrix สำหรับ การ หา ประสิทธิภาพความแม่นยำในการจำแนกโรคใบข้าวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO ดังแสดงภาพตาราง ConfusionMatrix ที่ 4.1 - 4.4 ได้สรุปประสิทธิภาพด้านความแม่นยำและประสิทธิภาพเวลาของแต่ละโมเดลไว้ดังนี้ โมเดลYOLOv5s ได้ค่าความแม่นยำของโรคใบจุดสีน้ำตาล 71.96%,โรคแมลงดำหนาม 56.96% , โรคใบไหม้ 75% โมเดลYOLOv5m ได้ค่าความแม่นยำ ของโรคใบจุดสีน้ำตาล 71.96%,โรคแมลงดำหนาม 60% , โรคใบไหม้ 67.68% YOLOv5l ได้ค่าความแม่นยำ ของโรคใบจุดสีน้ำตาล 71.96%,โรคแมลงดำหนาม 66.96% , โรคใบไหม้ 66.96% โมเดลYOLOv5x ได้ค่าความแม่นยำ ของโรคใบจุดสีน้ำตาล 81.96%,โรคแมลงดำหนาม 56.96% , โรคใบไหม้ 67.68%

Predicted	โรคใบจุดสีน้ำตาล	29	1	10
	โรคแมลงดำหนาม	8	23	9
	โรคใบไหม้	10	0	30
		โรคใบจุดสีน้ำตาล	โรคแมลงดำหนาม	โรคใบไหม้
		True Positives		

ภาพที่ 4.1 ตาราง ConfusionMatrixYOLOv5s

Predicted	โรคใบจุดสีน้ำตาล	29	3	8
	โรคแมลงดำหนาม	6	24	10
	โรคใบไหม้	7	4	29
		โรคใบจุดสีน้ำตาล	โรคแมลงดำหนาม	โรคใบไหม้

True Positives

ภาพที่ 4.2 ตาราง ConfusionMatrixYOLOv5m

Predicted	โรคใบจุดสีน้ำตาล	33	0	7
	โรคแมลงดำหนาม	4	27	9
	โรคใบไหม้	11	2	27
		โรคใบจุดสีน้ำตาล	โรคแมลงดำหนาม	โรคใบไหม้

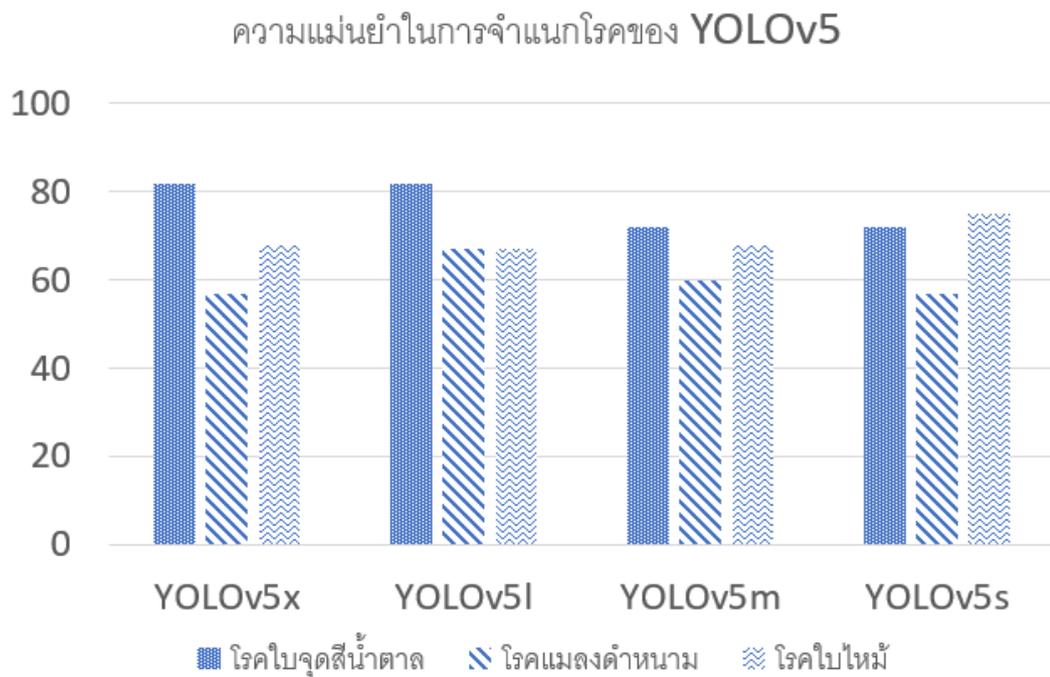
True Positives

ภาพที่ 4.3 ตาราง ConfusionMatrixYOLOv5l

Predicted	โรคใบจุดสีน้ำตาล	33	0	7
	โรคแมลงดำหนาม	4	27	9
	โรคใบไหม้	11	2	27
		โรคใบจุดสีน้ำตาล	โรคแมลงดำหนาม	โรคใบไหม้

True Positives

ภาพที่ 4.4 ตาราง ConfusionMatrixYOLOv5x



ภาพที่ 4.5 กราฟความแม่นยำในการจำแนกโรคของ โมเดลYOLOv5

บทที่ 5

สรุปและอภิปรายผล

5.1 สรุปผลและอภิปรายผล

จากการทดลองงานวิจัยการจำแนกโรคใบขาวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO ผู้วิจัยได้ศึกษาโรคของข้าว 3 โรคได้แก่ โรคใบจุดสีน้ำตาล, โรคแมลงดำหนามและโรคใบไหม้ ได้เตรียมภาพชุดข้อมูลเป็นจำนวน 1200 ภาพ ทำการเตรียมชุดเพื่อการฝึกและเรียนรู้ของโมเดล YOLOv5 เพื่อวัดประสิทธิภาพด้านความแม่นยำและประสิทธิภาพด้านเวลา ผู้วิจัยได้ทดลองกับโมเดลของYOLOv5 ได้แก่ YOLOv5s, YOLOv5m, YOLOv5l, และYOLOv5x ผลการทดลองจะแสดงให้เห็นว่าประสิทธิภาพด้านความแม่นยำกับประสิทธิภาพด้านเวลาของแต่ละโมเดล โดยที่ YOLOv5s ได้ค่าความแม่นยำ 67.98% ใช้เวลาในการสร้างโมเดลทำนาย 1 ชั่วโมง 33 นาที YOLOv5m ได้ค่าความแม่นยำ 66.55% ใช้เวลาในการสร้างโมเดลทำนาย 2 ชั่วโมง 17 นาที YOLOv5l ได้ค่าความแม่นยำ 71.96% ใช้เวลาในการสร้างโมเดลทำนาย 3 ชั่วโมง 42 นาที YOLOv5x ได้ค่าความแม่นยำ 68.87% ใช้เวลาในการสร้างโมเดลทำนาย 6 ชั่วโมง 29 นาที ดังแสดงภาพที่ 5.1 สรุปผลการทดลองสามารถบอกได้ว่า การเรียนรู้โมเดลของYOLOv5 สามารถจำแนกโรคข้าวจากใบขาวได้อย่างมีประสิทธิภาพ โดยที่มีค่าความแม่นยำมากที่สุด



ภาพที่ 5.1 กราฟแสดงความแม่นยำเฉลี่ยในการจำแนกโรคของโมเดลทำนายYOLOv5

5.2 ข้อจำกัด

1. ในงานวิจัยนี้ ผู้วิจัยได้ใช้ค่าเริ่มต้นของอัลกอริทึมของYOLOv5 สี่ตัว ได้แก่ YOLOv5s, YOLOv5m, YOLOv5l และYOLOv5x จากตารางที่ 3.1 โดยไม่ได้ปรับโหมดของอัลกอริทึม (Node) และฮิดเดนเลเยอร์ (Hidden layers)
2. การจำแนกโรคข้าว สามารถจำแนกได้เฉพาะโรคที่ให้โมเดลทำนายเรียนรู้ไว้ในชุดข้อมูล ได้แก่ โรคใบจุดสีน้ำตาล โรคแมลงดำหนาม โรคใบไหม้
3. การจำแนกโรคข้าว บางโรคมีลักษณะคล้ายกันจึงทำความยากในการจำแนกโรคลดลง เช่น โรคใบจุดสีน้ำตาล โรคใบไหม้

5.3 ข้อเสนอแนะและงานวิจัยในอนาคต

จากการทดลองการจำแนกโรคใบขาวโดยใช้โครงข่ายประสาทเทียมคอนโวลูชัน YOLO มีแนวทางในการพัฒนาให้สามารถนำไปใช้ประโยชน์ได้อย่างมีประสิทธิภาพมากขึ้น โดยมีข้อเสนอแนะและงานวิจัยในอนาคตดังนี้

1. พัฒนาการจำแนกชนิดโรคให้ขาวได้มากขึ้นกว่าเดิม เช่น โรคใบขีดสีน้ำตาล โรคขอบใบแห้ง
2. พัฒนา ปรับปรุง เทคนิค วิธีการและแนวคิดสำหรับเพิ่มประสิทธิภาพ เช่น เพิ่มเทคนิคการสร้างโมเดลโดยการปรับโหนด และฮิดเดนเลเยอร์ ให้เหมาะสมมากขึ้น

บรรณานุกรม

บรรณานุกรม

- Punyato, P., & Sidahao, N. (2019). การสร้างระบบตรวจนับบุคคลแบบเวลาจริงราคาประหยัดบนraspberry pi โดยประยุกต์อัลกอริทึม tiny yolo v3. *Engineering Transactions: A Research Publication of Mahanakorn University of Technology*, 22(2), 72–78.
- จันทนะชาติ, ชนาธิป, ดาวรีรัมย์, สวรรคพฐ, จันทร์นวล, & เก่ง. (n.d.). การตรวจหาช่องจอดรถแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียมyolo.
- ณัฐภัค ลาวัลย์วงศ์. (2021). การพัฒนาขั้นตอนวิธีสำหรับจำแนกโรคอ้อยที่พบบ่อยในประเทศไทย.
- ธนโชติ ภาชนะนัย, จักรกริช ปานเงิน, กรรณิการ์ คนงาม, วรชัย ศรีสมุดคำ, & วาสนา วงศ์ษา. (2022). เครื่องคัดแยกขยะรีไซเคิลด้วยการประมวลผลภาพ. *วารสารวิชาการเทคโนโลยีอุตสาหกรรมและวิศวกรรมมหาวิทยาลัยราชภัฏพิบูลสงคราม*, 4(2), 242–253.
- พัฒนจิตรศิลป์, & เพ็ญพิชชา. (2021). การตรวจจับสิ่งกีดขวางสำหรับเก้าอี้รถเข็นไฟฟ้าด้วยคอมพิวเตอร์วิทัศน์.
- เตชินท์ วรสิทธิ์, อีระ ภัทราพรนันท์, อีระยุทธ ตูจันดา, วสิน สินธุภิญโญ, กรรณทิพย์ กิรดิรัตน์ พฤกษ์, พิชญกาญจน์ เต็มนิรันรัตน์, อภิชน กิจวิมลรัตน์, จินตนา อันอาดมงาม, & สุจินต์ ภัทรภูวดล. (2022). การประยุกต์ใช้เทคนิคการเรียนรู้เชิงลึกสำหรับการรู้จำโรคขอบใบแห้งและโรคไหม้ของข้าว.

ภาคผนวก

ภาคผนวก ก

โค้ดของการสร้างโมเดลจำแนกภาพโรคใบขาว

ก.1 การดาวน์โหลด YOLO และติดตั้งไลบรารีสำหรับการใช้งาน YOLO

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt # install dependencies
import torch
import os
import utils
display = utils.notebook_init()
from IPython.display import Image, clear_output # to display images
```

ภาพที่ ก.1 ดาวน์โหลด YOLO และติดตั้งไลบรารีสำหรับการใช้งาน YOLO

จากภาพที่ ก.1 ดาวน์โหลด YOLOv5 จากเว็บไซต์ git และติดตั้งไลบรารีต่างๆสำหรับการใช้งาน YOLO ในการจำแนกโรคข้าวด้วยโครงข่ายประสาทเทียมคอนโวลูชัน YOLO

ก.2 โครงสร้างโมเดล

จากภาพที่ ก.2 ถึงภาพที่ ก.5 แสดงโครงสร้างของแต่ละโมเดล YOLOv5 ได้แก่ โครงสร้างYOLOv5s โครงสร้างYOLOv5m โครงสร้างYOLOv5l และโครงสร้างYOLOv5x

ก.3 การเทรนโมเดล

จากภาพที่ ก.6 เรียกใช้ไฟล์ train.py และตั้งค่าขนาดภาพ BatchSize epochs dataset จากนั้นเลือกโมเดลของ yolo สำหรับการเทรนและสร้างโมเดลการจำแนกโรคข้าว

ก.4 การทดสอบโมเดล

จากภาพที่ ก.7 เรียกใช้ไฟล์ Detect.py เลือกไฟล์ best.py ซึ่งเป็นตัวที่ดีที่สุดของโมเดลนั้นๆ ทดสอบการจำแนกโรคจากชุดข้อมูลทดสอบ จะแสดงการจำแนกโรคข้าวดังภาพที่ ก.8 บันทึกผลการทดลองและประเมินประสิทธิภาพแต่ละโมเดล

```

models > ! yolov5s.yaml
 1 # YOLOv5 🚀 by Ultralytics, AGPL-3.0 license
 2
 3 # Parameters
 4 nc: 80 # number of classes
 5 depth_multiple: 0.33 # model depth multiple
 6 width_multiple: 0.50 # layer channel multiple
 7 anchors:
 8   - [10,13, 16,30, 33,23] # P3/8
 9   - [30,61, 62,45, 59,119] # P4/16
10   - [116,90, 156,198, 373,326] # P5/32
11
12 # YOLOv5 v6.0 backbone
13 backbone:
14   # [from, number, module, args]
15   [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
16     [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
17     [-1, 3, C3, [128]],
18     [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
19     [-1, 6, C3, [256]],
20     [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
21     [-1, 9, C3, [512]],
22     [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
23     [-1, 3, C3, [1024]],
24     [-1, 1, SPPF, [1024, 5]], # 9
25   ]
26
27 # YOLOv5 v6.0 head
28 head:
29   [[-1, 1, Conv, [512, 1, 1]],
30     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
31     [[-1, 6], 1, Concat, [1]], # cat backbone P4
32     [-1, 3, C3, [512, False]], # 13
33
34     [-1, 1, Conv, [256, 1, 1]],
35     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
36     [[-1, 4], 1, Concat, [1]], # cat backbone P3
37     [-1, 3, C3, [256, False]], # 17 (P3/8-small)
38
39     [-1, 1, Conv, [256, 3, 2]],
40     [[-1, 14], 1, Concat, [1]], # cat head P4
41     [-1, 3, C3, [512, False]], # 20 (P4/16-medium)
42
43     [-1, 1, Conv, [512, 3, 2]],
44     [[-1, 10], 1, Concat, [1]], # cat head P5
45     [-1, 3, C3, [1024, False]], # 23 (P5/32-large)
46
47     [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
48   ]

```

ภาพที่ ก.2 โครงสร้างของโมเดลYOLOv5s

```

models > ! yolov5m.yaml
 1 # YOLOv5 🚀 by Ultralytics, AGPL-3.0 license
 2
 3 # Parameters
 4 nc: 80 # number of classes
 5 depth_multiple: 0.67 # model depth multiple
 6 width_multiple: 0.75 # layer channel multiple
 7 anchors:
 8   - [10,13, 16,30, 33,23] # P3/8
 9   - [30,61, 62,45, 59,119] # P4/16
10   - [116,90, 156,198, 373,326] # P5/32
11
12 # YOLOv5 v6.0 backbone
13 backbone:
14   # [from, number, module, args]
15   [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
16     [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
17     [-1, 3, C3, [128]],
18     [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
19     [-1, 6, C3, [256]],
20     [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
21     [-1, 9, C3, [512]],
22     [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
23     [-1, 3, C3, [1024]],
24     [-1, 1, SPPF, [1024, 5]], # 9
25   ]
26
27 # YOLOv5 v6.0 head
28 head:
29   [[-1, 1, Conv, [512, 1, 1]],
30     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
31     [[-1, 6], 1, Concat, [1]], # cat backbone P4
32     [-1, 3, C3, [512, False]], # 13
33
34     [-1, 1, Conv, [256, 1, 1]],
35     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
36     [[-1, 4], 1, Concat, [1]], # cat backbone P3
37     [-1, 3, C3, [256, False]], # 17 (P3/8-small)
38
39     [-1, 1, Conv, [256, 3, 2]],
40     [[-1, 14], 1, Concat, [1]], # cat head P4
41     [-1, 3, C3, [512, False]], # 20 (P4/16-medium)
42
43     [-1, 1, Conv, [512, 3, 2]],
44     [[-1, 10], 1, Concat, [1]], # cat head P5
45     [-1, 3, C3, [1024, False]], # 23 (P5/32-large)
46
47     [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
48   ]

```

ภาพที่ ก.3 โครงสร้างของโมเดลYOLOv5m

```

models > ! yolov5l.yaml
 1 # YOLOv5 🚀 by Ultralytics, AGPL-3.0 license
 2
 3 # Parameters
 4 nc: 80 # number of classes
 5 depth_multiple: 1.0 # model depth multiple
 6 width_multiple: 1.0 # layer channel multiple
 7 anchors:
 8   - [10,13, 16,30, 33,23] # P3/8
 9   - [30,61, 62,45, 59,119] # P4/16
10   - [116,90, 156,198, 373,326] # P5/32
11
12 # YOLOv5 v6.0 backbone
13 backbone:
14   # [from, number, module, args]
15   [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
16     [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
17     [-1, 3, C3, [128]],
18     [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
19     [-1, 6, C3, [256]],
20     [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
21     [-1, 9, C3, [512]],
22     [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
23     [-1, 3, C3, [1024]],
24     [-1, 1, SPPF, [1024, 5]], # 9
25   ]
26
27 # YOLOv5 v6.0 head
28 head:
29   [[-1, 1, Conv, [512, 1, 1]],
30     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
31     [[-1, 6], 1, Concat, [1]], # cat backbone P4
32     [-1, 3, C3, [512, False]], # 13
33
34     [-1, 1, Conv, [256, 1, 1]],
35     [-1, 1, nn.Upsample, [None, 2, 'nearest']],
36     [[-1, 4], 1, Concat, [1]], # cat backbone P3
37     [-1, 3, C3, [256, False]], # 17 (P3/8-small)
38
39     [-1, 1, Conv, [256, 3, 2]],
40     [[-1, 14], 1, Concat, [1]], # cat head P4
41     [-1, 3, C3, [512, False]], # 20 (P4/16-medium)
42
43     [-1, 1, Conv, [512, 3, 2]],
44     [[-1, 10], 1, Concat, [1]], # cat head P5
45     [-1, 3, C3, [1024, False]], # 23 (P5/32-large)
46
47     [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
48   ]

```

ภาพที่ ก.4 โครงสร้างของโมเดลYOLOv5l

```

models > ! yolo5x.yaml
1 # YOLOv5 🚀 by Ultralytics, AGPL-3.0 license
2
3 # Parameters
4 nc: 80 # number of classes
5 depth_multiple: 1.33 # model depth multiple
6 width_multiple: 1.25 # layer channel multiple
7 anchors:
8   - [10,13, 16,30, 33,23] # P3/8
9   - [30,61, 62,45, 59,119] # P4/16
10  - [116,90, 156,198, 373,326] # P5/32
11
12 # YOLOv5 v6.0 backbone
13 backbone:
14   # [from, number, module, args]
15   [[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
16    [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
17    [-1, 3, C3, [128]],
18    [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
19    [-1, 6, C3, [256]],
20    [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
21    [-1, 9, C3, [512]],
22    [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
23    [-1, 3, C3, [1024]],
24    [-1, 1, SPPF, [1024, 5]], # 9
25   ]
26
27 # YOLOv5 v6.0 head
28 head:
29   [[-1, 1, Conv, [512, 1, 1]],
30    [-1, 1, nn.Upsample, [None, 2, 'nearest']],
31    [-1, 6, 1, Concat, [1]], # cat backbone P4
32    [-1, 3, C3, [512, False]], # 13
33
34    [-1, 1, Conv, [256, 1, 1]],
35    [-1, 1, nn.Upsample, [None, 2, 'nearest']],
36    [-1, 4, 1, Concat, [1]], # cat backbone P3
37    [-1, 3, C3, [256, False]], # 17 (P3/8-small)
38
39    [-1, 1, Conv, [256, 3, 2]],
40    [-1, 14, 1, Concat, [1]], # cat head P4
41    [-1, 3, C3, [512, False]], # 20 (P4/16-medium)
42
43    [-1, 1, Conv, [512, 3, 2]],
44    [-1, 10, 1, Concat, [1]], # cat head P5
45    [-1, 3, C3, [1024, False]], # 23 (P5/32-large)
46
47    [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
48   ]

```

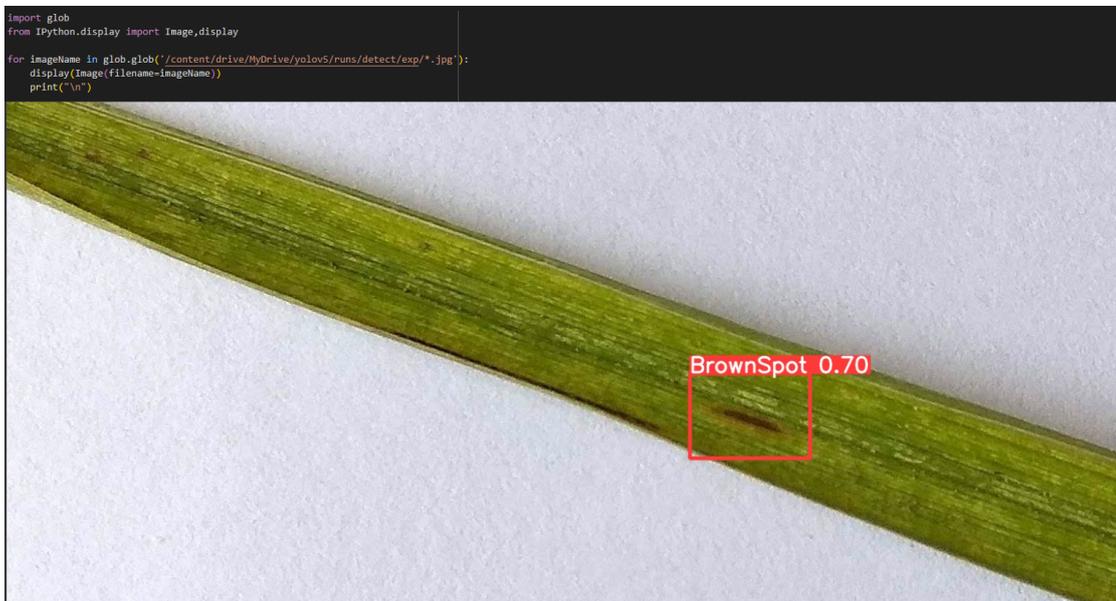
ภาพที่ ก.5 โครงสร้างของโมเดลYOLOv5x

```
!python train.py --img 640 --batch 16 --epochs 200 --data /content/drive/MyDrive/dataset/data.yaml --weights yolov5s.pt
```

ภาพที่ ก.6 การเรียกใช้ไฟล์ train.py สำหรับในการเทรนโมเดลการจำแนกโรคข้าว

```
!python detect.py --weights /content/drive/MyDrive/yolov5/runs/train/YOLOv5L/weights/best.pt --source /content/drive/MyDrive/test
```

ภาพที่ ก.7 การเรียกใช้ไฟล์ Detect.py สำหรับการทดสอบการจำแนกโรคข้าว



ภาพที่ ก.8 โรคข้าวที่ YOLO สามารถตรวจจับและจำแนกได้จากภาพ

ประวัติย่อของผู้วิจัย

ประวัติย่อของผู้วิจัย



ชื่อ สกุล	นาย ธาวิน ภัคดีพรเลิศ
วัน เดือน ปีเกิด	7 มิถุนายน 2543
สถานที่เกิด	220/1 หมู่ 9 ตำบลแม่สลองใน อำเภอมะป้าหลวง จังหวัด เชียงราย
เบอร์โทร	065-927-2419
ประวัติการศึกษา	
2556	ระดับ 1-6 มัธยมศึกษา จบ การ ศึกษา จาก โรงเรียน แม่จัน วิทยาคม อำเภอมะจัน จังหวัดเชียงราย