

การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผล  
การแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน



นายหัตถ์ มละสาร

การศึกษาค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต (เทคโนโลยีสารสนเทศและการสื่อสาร)  
สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมมาธิราช

พ.ศ. 2566

**Scratched Surface Detection of Metal Objects Using Processing Methods  
of Image Subdivision and Convolution Neural Network**



An Independent Study Submitted in Partial Fulfillment of the Requirements for  
the Degree of Master of Science (Information and Communication Technology)  
School of Science and Technology Sukhothai Thammathirat Open University

2023

หัวข้อการศึกษาค้นคว้าอิสระ	การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผล การแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน
ชื่อและนามสกุล	นายณัฏฐิ มละสาร
แขนงวิชา / วิชาเอก	วิทยาศาสตร์มหาบัณฑิต (เทคโนโลยีสารสนเทศและการสื่อสาร)
สาขาวิชา	วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช
อาจารย์ที่ปรึกษา	อาจารย์ ดร.เตชต์จัฐสินี เปี้ยซ้าย

การศึกษาค้นคว้าอิสระนี้ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรระดับปริญญาโท เมื่อวันที่ 28 สิงหาคม พ.ศ. 2566

คณะกรรมการสอบการศึกษาค้นคว้าอิสระ

..... ประธานกรรมการ  
(อาจารย์ ดร.เตชต์จัฐสินี เปี้ยซ้าย)  
..... กรรมการ  
(รองศาสตราจารย์นศพัชราณี ชินปัญญาชนะ)



.....  
(ผู้ช่วยศาสตราจารย์ ดร.ภูมิ เจือศิริภักดี)

ประธานกรรมการประจำสาขาวิชาวิทยาศาสตร์และเทคโนโลยี

ชื่อการศึกษาค้นคว้าอิสระ การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการ  
ประมวลผล การแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน  
ผู้ศึกษา นายณัฏฐิ มละสาร รหัสนักศึกษา 2649600257  
ปริญญา วิทยาศาสตร์มหาบัณฑิต  
อาจารย์ที่ปรึกษา อาจารย์ ดร.เดชรัฐสถิณี เพี้ยชัย ปีการศึกษา 2566

## บทคัดย่อ

วัตถุประสงค์การศึกษาค้นคว้าอิสระนี้คือ (1) เพื่อศึกษาการตรวจจับการขีดข่วนบนภาพพื้นผิวโลหะอะลูมิเนียมโดยใช้วิธีการแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน (2) เพื่อพัฒนาและประเมินประสิทธิภาพโมเดลการตรวจจับการขีดข่วนบนภาพพื้นผิวโลหะอะลูมิเนียมโดยใช้วิธีการแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน

การศึกษานี้ได้พัฒนาวิธีการตรวจจับรอยขีดข่วนบนผิวโลหะอะลูมิเนียมโดยใช้วิธีการแบ่งภาพอินพุตที่มีขนาดจำนวนพิกเซลและความละเอียดสูงออกเป็นภาพย่อย แต่ละภาพย่อยที่ถูกแบ่งจะถูกประมวลผลด้วยโครงข่ายประสาทแบบคอนโวลูชัน (CNN) เพื่อตรวจจับรอยขีดข่วน วิธีการแบ่งภาพสามารถแบ่งได้เป็น 4 ส่วน, 9 ส่วน, 16 ส่วน, 36 ส่วน และ 64 ส่วน การแบ่งภาพช่วยให้รายละเอียดในภาพยังคงอยู่ได้แม้ในการประมวลผลของ CNN ที่มีการปรับลดขนาดพิกเซลของภาพลง โดยวิธีการของ CNN ในการศึกษานี้ใช้โมเดล YOLOv8n, โมเดล YOLOv8s และโมเดล YOLOv8x สำหรับทำการประมวลผลภาพย่อยแต่ละภาพ

ผลการศึกษาพบว่า (1) วิธีการแบ่งภาพและ CNN สามารถตรวจจับรอยขีดข่วนได้เพิ่มขึ้นและดีกว่าการไม่แบ่งภาพ และ (2) วิธีการ CNN ของโมเดล YOLOv8s ได้รับผลที่เหมาะสมสำหรับวิธีการแบ่งภาพอินพุตเป็นภาพย่อยแบบ 4 ส่วน, 9 ส่วน, 16 ส่วน, 36 ส่วน และ 64 ส่วน การประเมินประสิทธิภาพวิธีการนี้ ได้ใช้ชุดข้อมูลภาพฝึกสอน 2,600 ภาพที่มีรอยขีดข่วนแบบต่าง ๆ บนภาพ และชุดข้อมูลภาพทดสอบ 7 ภาพที่มีรอยขีดข่วนรวม 48 รอย การตรวจจับรอยขีดข่วนของภาพทดสอบที่ไม่ได้ถูกแบ่งภาพสามารถตรวจจับรอยได้ 37 รอยหรือร้อยละ 77.08 และเมื่อแบ่งภาพทดสอบนี้เป็น 4 ส่วนสามารถตรวจจับรอยขีดข่วนที่เหลือเพิ่มได้อีก 10 รอยหรือเพิ่มเป็นร้อยละ 90.90 และเมื่อแบ่งภาพทดสอบออกเป็น 9 ส่วน, 16 ส่วน, 36 ส่วน และ 64 ส่วนเทียบกับภาพทดสอบที่ไม่ได้ถูกแบ่งภาพ วิธีการนี้สามารถตรวจจับรอยที่เหลือทั้งหมดได้ครบ 11 รอยหรือคิดเป็นร้อยละ 100

**คำสำคัญ** การตรวจจับรอยขีดข่วน การแบ่งภาพ โครงข่ายประสาทแบบคอนโวลูชัน

Independent Study title: “Scratched Surface Detection of Metal Objects Using Processing Methods of Image Subdivision and Convolution Neural Network”

Author: “Mr. NATTEE MALASARN”; ID: “2649600257”;

Degree: Masrter of Science

Independent Study Advisor: Dr.Tejtasin Phiasai; Academic year: 2023

### **Abstract**

The objectives of this independent study were (1) to study the scratched detection on images of aluminum metal surfaces using image subdivision and convolutional neural network methods and (2) to develop and evaluate the performance of a model for the scratched detection on images of aluminum metal surfaces using image subdivision and convolutional neural network methods.

This study developed the scratched detection method on aluminum metal surfaces using the subdivision method of the input images that had high pixel dimensions and resolution. Each sub-image was processed using the convolutional neural network (CNN) to detect the scratched regions. The image subdivision method could separate the image into 4, 9, 16, 36, and 64 parts. The separated images could preserve the content in the image, although the CNN processing reduced the pixel dimensions of the images. The CNN method of this study used the YOLOv8n, YOLOv8s, and YOLOv8x models for processing the sub-images.

The study results found that (1) the image subdivision and CNN methods could detect more and better scratches than no image subdivision, and (2) the YOLOv8s model of the CNN method achieved appropriate results for the image subdivision methods that included the 4, 9, 16, 36, and 64 parts. The performance evaluation of this method used the training image dataset, 2,600 images, that had several scratches, and the testing image dataset, 7 images, that had 48 scratches. The scratch detection results of the non-subdivision images found 37 scratches or 77.08% and the testing images that separated into 4 parts could detect the remaining scratches of 10 scratches or about 90.90%. The remaining results of the testing images that were separated into 9, 16, 36, and 64 parts could detect all remaining scratches of 11 scratches or about 100%.

**Keywords** : Scratch detection, Image Subdivision, Convolution Neural Network

## กิตติกรรมประกาศ

การค้นคว้าอิสระเล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา อาจารย์ ดร.เดชต์ จู สິณป์ เพี้ยซ้าย ที่ให้ความช่วยเหลือ ให้คำปรึกษา ให้คำแนะนำ ให้ความรู้ ให้วิธีคิดมุมมองใหม่ๆ ตลอดจนช่วยแก้ไขปรับปรุงข้อบกพร่องต่างๆ ด้วยความเอาใจใส่เป็นอย่างดี จนการค้นคว้าอิสระฉบับนี้เสร็จสมบูรณ์ สามารถนำความรู้ไปพัฒนาต่อยอดเพื่อให้เกิดประโยชน์ต่อสังคม ขอกราบขอบพระคุณกรรมการสอบการค้นคว้าอิสระ รศ.นศัप्พชาณัณ ชินปัญช์ธนะ ที่สละเวลาในการสอบในครั้งนี้ และให้คำแนะนำที่เป็นประโยชน์ในการปรับปรุงให้การค้นคว้าอิสระนี้มีคุณภาพและมีความสมบูรณ์ยิ่งขึ้น ขอกราบขอบพระคุณอาจารย์ทุกท่านที่ได้อบรมมอบความรู้ อันมีค่ายิ่ง

นายณัฒธี มละสาร



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ง
บทคัดย่อภาษาอังกฤษ .....	จ
กิตติกรรมประกาศ .....	ฉ
สารบัญ .....	ช
สารบัญตาราง .....	ฎ
สารบัญรูปภาพ .....	ฏ
สารบัญรูปภาพ (ต่อ).....	ฐ
สารบัญรูปภาพ (ต่อ).....	ฑ
บทที่ 1 บทนำ.....	1
1. ความเป็นมาและความสำคัญของปัญหา.....	1
2. วัตถุประสงค์ในการศึกษา.....	3
3. กรอบแนวคิดของการศึกษา.....	3
4. ขอบเขตของการศึกษา.....	4
5. นิยามศัพท์เฉพาะ.....	5
6. ประโยชน์ที่คาดว่าจะได้รับ .....	6
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง.....	7
1. การเรียนรู้เชิงลึก (Deep Learning).....	7
2. โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network (CNN)) .....	9
2.1 คอนโวลูชัน (Convolution).....	10

2.2	โครงข่ายการเรียนรู้เชิงลึก (Deep Neural Network)	15
3.	การตรวจจับวัตถุ (Object detection)	15
3.1	ขั้นตอนหลักของการตรวจจับวัตถุ (Object Detection)	16
3.2	เทคนิคการตรวจจับวัตถุ	16
3.3	YOLO (You Only Look Once) version 8	17
4.	การวัดผลประสิทธิภาพโมเดล	21
5.	งานวิจัยที่เกี่ยวข้อง	24
บทที่ 3	วิธีดำเนินการศึกษา	30
1.	วิเคราะห์ปัญหาและศึกษาเทคโนโลยีที่นำมาใช้	30
1.1	One-stage object detection	31
1.2	Two-stage object detection	31
2.	รูปแบบและวิธีการศึกษา	33
3.	ขั้นตอนการเก็บรวบรวมข้อมูล และการสร้างชุดข้อมูลการฝึกสอน	34
3.1	ขั้นตอนการเก็บรวบรวมข้อมูล	34
3.2	กำหนดประเภทของวัตถุ (Image Annotation)	37
3.3	การเพิ่มภาพ Augmentations	38
3.4	การแบ่งข้อมูล Data Split	39
4.	ขั้นตอนการสร้างโมเดล การตรวจจับวัตถุ	40
4.1	ฝึกสอนโมเดล (Model Training)	40
5.	การนำโมเดลไปใช้งาน	43
5.1	ติดตั้ง YOLOv8 และเตรียมความพร้อม	43
5.2	ออกแบบการทำงานของระบบ	43



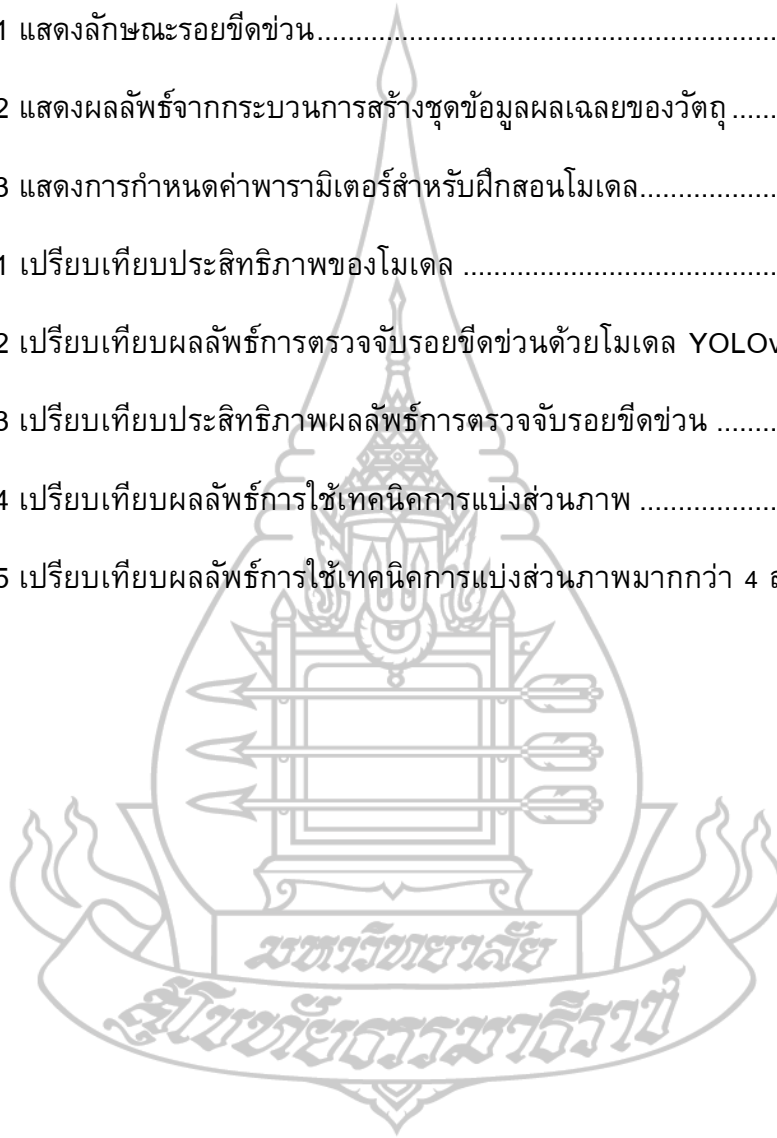
5.3	พัฒนาส่วนประสานผู้ใช้.....	45
5.4	พัฒนาแอปพลิเคชัน.....	46
6.	วิธีการประเมินประสิทธิภาพ.....	47
6.1	การประเมินประสิทธิภาพการตรวจจับวัตถุ.....	47
6.2	การวัดประสิทธิภาพของโมเดล.....	48
บทที่ 4	ผลการวิเคราะห์ข้อมูล.....	50
1.	สรุปผลการพัฒนาโมเดล.....	50
1.1	เปรียบเทียบประสิทธิภาพของโมเดล.....	50
1.2	เปรียบเทียบการตรวจจับรอยขีดข่วนของโมเดลทั้ง 3 แบบ.....	51
1.3	เปรียบเทียบประสิทธิภาพการตรวจจับรอยขีดข่วนและเวลาที่ใช้ในการตรวจจับ.....	52
1.4	เปรียบเทียบผลลัพธ์การตรวจจับรอยขีดข่วนด้วยเทคนิคการแบ่งส่วนภาพ.....	54
1.5	การวิเคราะห์ผลลัพธ์จากการฝึกฝนโมเดล YOLOv8s.....	58
1.5.1	วิเคราะห์ผลลัพธ์ค่า mAP50.....	58
1.5.2	วิเคราะห์ผลลัพธ์ตาราง Confusion Matrix.....	58
1.5.3	กราฟแสดงความสัมพันธ์.....	59
1.5.4	วิเคราะห์กราฟผลการฝึกฝนโมเดล.....	60
2.	สรุปผลการพัฒนาส่วนประสานผู้ใช้ GUI.....	63
3.	ทดสอบและสรุปผลการทำงานของโปรแกรม.....	64
บทที่ 5	สรุปผลการศึกษา และข้อเสนอแนะ.....	67
1.	สรุปผลการศึกษา.....	67

2. ข้อเสนอแนะ .....	68
บรรณานุกรม.....	69
ภาคผนวก.....	73
ประวัติผู้ศึกษา.....	80



## สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงลักษณะรอยขีดข่วน.....	35
ตารางที่ 3.2 แสดงผลลัพธ์จากกระบวนการสร้างชุดข้อมูลผลเฉลยของวัตถุ .....	37
ตารางที่ 3.3 แสดงการกำหนดค่าพารามิเตอร์สำหรับฝึกสอนโมเดล.....	42
ตารางที่ 4.1 เปรียบเทียบประสิทธิภาพของโมเดล .....	50
ตารางที่ 4.2 เปรียบเทียบผลลัพธ์การตรวจจับรอยขีดข่วนด้วยโมเดล YOLOv8 n,s,x .....	51
ตารางที่ 4.3 เปรียบเทียบประสิทธิภาพผลลัพธ์การตรวจจับรอยขีดข่วน .....	52
ตารางที่ 4.4 เปรียบเทียบผลลัพธ์การใช้เทคนิคการแบ่งส่วนภาพ .....	54
ตารางที่ 4.5 เปรียบเทียบผลลัพธ์การใช้เทคนิคการแบ่งส่วนภาพมากกว่า 4 ส่วน .....	57



## สารบัญรูปภาพ

	หน้า
ภาพที่ 2.1 ความสัมพันธ์ของ AI Machine Learning และ Deep Learning .....	7
ภาพที่ 2.2 โครงสร้างของ Deep Neural Network แบบหลายชั้น.....	9
ภาพที่ 2.3 ขั้นตอนของกระบวนการสกัดคุณลักษณะของวัตถุ.....	10
ภาพที่ 2.4 การทำงานของตัวกรองเคอร์เนล .....	11
ภาพที่ 2.5 การเพิ่มพื้นที่ขอบภาพ .....	12
ภาพที่ 2.6 การทำ Reflection padding.....	12
ภาพที่ 2.7 แสดง Convolution Layer ที่ได้จากการทำ Feature maps .....	13
ภาพที่ 2.8 แสดงสมการ Rectified Linear Unit และสมการ Sigmoid .....	13
ภาพที่ 2.9 การลดขนาดข้อมูลด้วย Max Pooling .....	14
ภาพที่ 2.10 กระบวนการแปลงข้อมูลให้อยู่ในรูปแบบหนึ่งมิติ.....	14
ภาพที่ 2.11 แสดงการนำข้อมูลเข้า Fully Connected Layer.....	15
ภาพที่ 2.12 เปรียบเทียบสถาปัตยกรรมเทคนิคในการตรวจจับวัตถุ .....	17
ภาพที่ 2.13 การตีกรอบ Bounding Box .....	18
ภาพที่ 2.14 หลักการทำงานของอัลกอริทึม YOLO .....	18
ภาพที่ 2.15 ผลลัพธ์หลังจากกระบวนการ Non-Maximum Suppression .....	19
ภาพที่ 2.16 สถาปัตยกรรมของอัลกอริทึม YOLOv8.....	21
ภาพที่ 2.17 การหาค่า IoU .....	22
ภาพที่ 2.18 ภาพตัวอย่างการซ้อนทับของ Ground Truth กับ Predicted Box .....	22
ภาพที่ 2.19 กำหนดเป็น TP เมื่อค่า IoU > 0.5 มีการทำนายคลาสคำตอบถูกต้อง.....	23

## สารบัญรูปภาพ (ต่อ)

	หน้า
ภาพที่ 2.20 กำหนดเป็น FP เมื่อทำนายคลาสคำตอบไม่ตรงกับผลเฉลย.....	24
ภาพที่ 2.21 กำหนดเป็น FN เมื่อมีการทำนายคลาสผิดพลาดหรือไม่มีผลการทำนาย .....	24
ภาพที่ 2.22 ผลการทดลองการตรวจจับฉลากตามกลุ่มที่ออกแบบ.....	25
ภาพที่ 2.23 เปรียบเทียบประสิทธิภาพของโมเดล CNN, Faster R-CNN .....	28
ภาพที่ 3.1 การแบ่งภาพอินพุตภาพที่มีความละเอียดสูงขนาด 3000x4000 pixel แบบ 4 ส่วน	32
ภาพที่ 3.2 แผนผังการทำงานของระบบตรวจจับรอยขีดข่วนโดยใช้เทคนิคการแบ่งส่วนภาพ .	32
ภาพที่ 3.3 แผนผังการสร้างชุดข้อมูลการฝึกสอน.....	34
ภาพที่ 3.4 แสดงการตีกรอบวัตถุและกำหนดประเภทของวัตถุ .....	38
ภาพที่ 3.5 กระบวนการทำ Augmentation.....	39
ภาพที่ 3.6 แสดงการปรับค่าพารามิเตอร์กระบวนการ Augmentation .....	39
ภาพที่ 3.7 แสดงการเปรียบเทียบอัลกอริทึม YOLO ในแต่ละเวอร์ชัน .....	40
ภาพที่ 3.8 แผนผังขั้นตอนการฝึกสอนโมเดล.....	41
ภาพที่ 3.9 กำหนดพารามิเตอร์ data.yaml สำหรับฝึกสอนโมเดล.....	42
ภาพที่ 3.10 แสดงผลลัพธ์การฝึกฝนโมเดล YOLOv8s .....	43
ภาพที่ 3.11 แผนผังการทำงานของระบบการตรวจจับรอยขีดข่วนบนผิวอลูมิเนียม .....	45
ภาพที่ 3.12 การออกแบบ User Interface ด้วยเครื่อง QT Designer .....	46
ภาพที่ 3.13 ภาพตัวอย่างการพัฒนาโปรแกรมด้วยภาษา Python .....	47
ภาพที่ 3.14 วิธีคำนวณค่า Intersection Over Union.....	48
ภาพที่ 3.15 แสดงความสัมพันธ์ของค่า mAP .....	49

## สารบัญรูปภาพ (ต่อ)

	หน้า
ภาพที่ 4.1 แสดงผลลัพธ์ของกระบวนการตรวจสอบโมเดล .....	58
ภาพที่ 4.2 แสดงตาราง Confusion Matrix .....	59
ภาพที่ 4.3 แสดงกราฟความสัมพันธ์ของแต่ละค่าในการตรวจจับวัตถุ .....	60
ภาพที่ 4.4 กราฟสรุปผลการฝึกฝนและตรวจสอบของโมเดล .....	62
ภาพที่ 4.5 แสดงตัวอย่าง Train_batch และ Validate_batch .....	62
ภาพที่ 4.6 แสดงหน้าจอโปรแกรมตรวจสอบรอยขีดข่วนบนผิวโลหะอลูมิเนียม .....	63
ภาพที่ 4.7 แสดงการเริ่มต้นการทำงานของโปรแกรม .....	64
ภาพที่ 4.8 แสดงหน้าจอโปรแกรมเริ่มทำการประมวลผล .....	65
ภาพที่ 4.9 แสดงหน้าจอโปรแกรมตรวจจับรอยขีดข่วนบนชิ้นงานเสีย .....	65
ภาพที่ 4.10 แสดงหน้าจอโปรแกรมตรวจจับรอยขีดข่วนบนชิ้นงานดี .....	66



## บทที่ 1

### บทนำ

#### 1. ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันธุรกิจที่ผลิตกล่องผลิตภัณฑ์ของสินค้าราคาสูง ( Luxury Product ) มีการแข่งขันสูงเพื่อให้ได้คำสั่งซื้อจากลูกค้าจำเป็นต้องผลิตสินค้าที่มีคุณภาพสูง มีความรวดเร็วในการส่งมอบให้กับลูกค้า เพื่อให้เกิดความพึงพอใจกับลูกค้า จึงมีการจัดตั้งมาตรฐานในการผลิตสินค้าขึ้นมาควบคุมการผลิต อีกทั้งยังมีมาตรฐานของลูกค้าที่ถูกกำหนดขึ้นมาควบคุมคุณภาพของสินค้า โดยมีการควบคุมมาตรฐานตั้งแต่วัตถุดิบนำเข้าการผลิต กระบวนการผลิต และสินค้าสำเร็จรูป ทั้งนี้การตรวจสอบวัตถุดิบที่เป็นอโลหะนิยม ซึ่งเป็นชิ้นส่วนหนึ่งของผลิตภัณฑ์ ซึ่งพบปัญหาว่าเกิดรอยขีดข่วนได้ง่าย ซึ่งมีกระบวนการตรวจสอบรอยขีดข่วนด้วยสายตาคนอยู่แล้ว แต่ยังมีชิ้นงานที่มีรอยขีดข่วนเข้าสู่กระบวนการผลิต เนื่องจากการตรวจสอบด้วยสายตาเป็นเวลานานเกิดความเมื่อยล้าและผู้ตรวจสอบบางคนยังขาดประสบการณ์ในการตรวจสอบหากนำวัตถุดิบที่มีรอยขีดข่วนบนผิวหน้าเข้าสู่กระบวนการผลิต ชิ้นงานดังกล่าวจะไม่ผ่านมาตรฐานคุณภาพที่กำหนดและจะเป็นงานเสียทันที ทำให้เกิดต้นทุนในการผลิตที่สูงขึ้น ไม่สามารถส่งสินค้าให้ลูกค้าได้ตามกำหนด ทั้งยังทำให้ขาดความเชื่อมั่นจากลูกค้า ส่งผลให้ขาดความสามารถในการแข่งขันทางธุรกิจ

เพื่อให้ได้สินค้าที่มีคุณภาพสูงและสร้างรายได้เปรียบในการแข่งขัน จึงได้นำเทคโนโลยีเข้ามาช่วยในการผลิต โดยการนำ AI (Artificial Intelligence) มาช่วยในกระบวนการตรวจสอบคุณภาพของสินค้า เพื่อให้ได้สินค้าที่มีคุณภาพ มั่นใจว่าผลิตภัณฑ์ที่ผลิตออกมามีคุณภาพที่สูงและตรงตามมาตรฐานที่กำหนดไว้ ลดความผิดพลาดและข้อบกพร่องที่อาจเกิดขึ้นในการผลิต ลดความสูญเสียที่อาจเกิดขึ้นจากการผลิตสินค้าที่ไม่ตรงตามมาตรฐาน ทำให้มีศักยภาพในการแข่งขันสูงขึ้น สอดคล้องกับการผลักดันของภาครัฐที่สนับสนุนให้มีการปฏิวัติอุตสาหกรรม (Industrial Revolution) ซึ่งปัจจุบันเข้าสู่การปฏิวัติอุตสาหกรรมครั้งที่ 4 หรืออุตสาหกรรม 4.0 ซึ่งมีการพัฒนาจากยุคอุตสาหกรรม 3.0 ที่เริ่มนำมาประยุกต์ใช้ในอุตสาหกรรมใช้เครื่องจักรอัตโนมัติ หรือหุ่นยนต์ในการผลิต เพราะมีความละเอียดแม่นยำ ใช้กำลังคนควบคุมน้อยลงทำให้กระบวนการผลิตมีประสิทธิภาพและมีมาตรฐานสูงขึ้น โดยการพัฒนาเข้าสู่ยุคอุตสาหกรรม 4.0 มีการนำเทคโนโลยีดิจิทัลมาประยุกต์ใช้ในกระบวนการผลิต ทำให้เกิดการผลิตที่หลากหลาย เชื่อมโยงกันเป็นระบบเครือข่าย มีการนำข้อมูลขนาดใหญ่มาวิเคราะห์ อุปกรณ์มีการเชื่อมโยงแลกเปลี่ยนข้อมูลกันในรูปแบบอินเทอร์เน็ตของสรรพสิ่ง รวมไปถึงการใช้งานระบบ

จักรกลเรียนรู้ และปัญญาประดิษฐ์ รวมถึงนำเทคโนโลยีเข้ามาช่วยเหลือในการบริหารระบบภายในองค์กร

ในปัจจุบันเทคโนโลยีคอมพิวเตอร์วิชัน (Computer vision) เป็นแขนงหนึ่งของวิทยาการปัญญาประดิษฐ์หรือ AI ที่มุ่งเน้นในการเข้าใจและวิเคราะห์ภาพหรือวิดีโอโดยใช้คอมพิวเตอร์และเทคโนโลยีการประมวลผลภาพ เทคโนโลยีคอมพิวเตอร์วิชันมีการพัฒนาไปมาก มีความรวดเร็วในการทำงานและมีความแม่นยำสูง มีการนำมาใช้งานทางด้านกรจำแนกวัตถุ (Classification) การตรวจจับวัตถุ (Detection) การแยกประเภทวัตถุ (Segmentation) อย่างแพร่หลาย และใช้เวลาในการทำงานที่รวดเร็ว ทั้งนี้ความเร็วในการทำงานขึ้นอยู่กับความซับซ้อนในการออกแบบโครงสร้างสถาปัตยกรรม ผู้ศึกษาจึงได้นำเทคโนโลยีนี้เข้ามาช่วยในการตรวจสอบรอยขีดข่วนบนผิวอลูมิเนียมที่เป็นวัตถุดิบในการผลิตทดแทนการตรวจสอบด้วยสายตาจากคนเพียงอย่างเดียว

โดยบทความนี้ต้องการที่จะนำเสนอวิธีการใช้ Convolution Neural Network (CNN) ในการแก้ไขปัญหาดังกล่าว โดยออกแบบพัฒนาด้วยโมเดล YOLOv8 (You Only Look Once) สำหรับเป็นแม่แบบในการฝึกสอนโมเดลการตรวจจับรอยขีดข่วน ซึ่ง YOLOv8 เป็นปัญญาประดิษฐ์การเรียนรู้เชิงลึก (Deep Learning) ในการตรวจจับวัตถุแบบเรียลไทม์ โดยนำโมเดลที่ผ่านการฝึกสอนมาแล้วทำงานร่วมกับเทคนิคการแบ่งภาพอินพุตที่มีความละเอียดสูงก่อนเข้าโมเดลการตรวจจับรอยขีดข่วนบนผิวโลหะอลูมิเนียม โดย YOLOv8 มีการทำงานแบบโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Networks) ที่เป็นการสกัดคุณลักษณะที่สำคัญของข้อมูลที่สนใจ (Feature Extraction) เพื่อนำข้อมูลที่สกัดได้ส่งเข้าฝึกสอนโมเดลเพื่อให้ได้โมเดลที่มีประสิทธิภาพสูงในการตรวจจับวัตถุภายในภาพหรือวิดีโอ YOLOv8 เป็นอัลกอริทึมเวอร์ชันล่าสุดที่มีการพัฒนาปรับปรุงประสิทธิภาพจากเวอร์ชันก่อนทั้งด้านความเร็วและความแม่นยำในการตรวจจับวัตถุมาตรวจจับรอยขีดข่วนและระบุตำแหน่งบนผิวอลูมิเนียม

การออกแบบและพัฒนาระบบตรวจจับรอยขีดข่วนและระบุตำแหน่งบนผิวโลหะอลูมิเนียม โดยใช้โครงข่ายประสาทแบบคอนโวลูชัน ด้วยอัลกอริทึม YOLOv8 เพื่อสร้างโมเดลในการตรวจจับรอยขีดข่วนบนผิวอลูมิเนียม ทำงานร่วมกับกล่องที่มีความละเอียดสูงภายในสภาพแวดล้อมที่มีการควบคุมแสงพร้อมทั้งยังออกแบบและพัฒนาส่วนประสานผู้ใช้ (User Interface) เพื่อให้สะดวกในการใช้งาน การพัฒนาระบบนี้เพื่อเป็นต้นแบบใช้งานในกระบวนการตรวจสอบคุณภาพของวัตถุดิบ



## 2. วัตถุประสงค์ในการศึกษา

2.1 เพื่อศึกษาวิธีการการตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผลการแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน

2.2 เพื่อพัฒนาและประเมินประสิทธิภาพโมเดล การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผลการแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน

## 3. กรอบแนวคิดของการศึกษา

3.1 การวิเคราะห์และออกแบบระบบตรวจจับรอยขีดข่วนและระบุตำแหน่งบนผิวโลหะอลูมิเนียม โดยมีฟังก์ชันการทำงาน ดังนี้

- 1) ตรวจจับรอยขีดข่วน
- 2) ระบุตำแหน่งของการเกิดรอยขีดข่วน
- 3) มีโปรแกรมบริหารจัดการ การตรวจสอบรอยขีดข่วน

3.2 การพัฒนาระบบตรวจจับรอยขีดข่วนและระบุตำแหน่งบนผิวโลหะอลูมิเนียม

1) พัฒนาชุดข้อมูลฝึกสอนโมเดลจากภาพรอยที่เกิดขึ้น และทำการระบุตำแหน่งผลเฉลยรอยขีดข่วนด้วย Website Roboflow.com

2) พัฒนาด้วยอัลกอริทึมการเรียนรู้เชิงลึกโครงข่ายประสาทเทียมคอนโวลูชัน (Convolutional Neural Network) ในการศึกษาที่ใช้โมเดล YOLOv8 เป็นแม่แบบ

3) ฝึกสอนโมเดลด้วย Google Colab

4) พัฒนาโปรแกรมเพื่อบริหารจัดการ การตรวจสอบรอยขีดข่วนและระบุตำแหน่ง

3.3 ทดสอบและเปรียบเทียบประสิทธิภาพของโมเดล ระบบตรวจจับรอยขีดข่วนและระบุตำแหน่งบนผิวโลหะอลูมิเนียม

#### 4. ขอบเขตของการศึกษา

ศึกษาและพัฒนาแบบจำลองการตรวจจับรอยขีดข่วนและระบุตำแหน่งบนพื้นผิววัสดุอลูมิเนียม โดยมีขอบเขตการศึกษาและพัฒนา ดังนี้

**4.1 ฟังก์ชันการทำงานของระบบ** โมเดลที่ศึกษาและพัฒนาขึ้นใช้ในการตรวจจับรอยขีดข่วนที่อยู่บนผิวโลหะอลูมิเนียมเท่านั้น หากนำไปตรวจจับกับผิววัสดุอื่นที่มีลักษณะพื้นผิวที่แตกต่างจากพื้นผิวอลูมิเนียม อาจมีความคลาดเคลื่อนของผลลัพธ์เกิดขึ้น หรือตรวจจับไม่พบได้ โดยมีฟังก์ชันการทำงาน ดังนี้

- 1) สามารถตรวจจับรอยขีดข่วนบนพื้นผิววัสดุที่เป็นอลูมิเนียมได้ ภายใต้พื้นที่ควบคุมแสงสว่าง
- 2) สามารถระบุตำแหน่งของรอยที่เกิดขึ้นได้
- 3) มีซอฟต์แวร์ในการตรวจสอบ สามารถตรวจสอบภาพจากการตรวจสอบก่อนหน้าได้

**4.2 ชุดข้อมูลที่ใช้ในการฝึกสอนโมเดล** โดยมีจำนวนชนิดของวัตถุ 1 ชนิด คือ รอยขีดข่วน ซึ่งมีหลายลักษณะ เช่น เป็นเส้นสั้นและยาว รอยเส้นแบบจุดไขปลา รอยเส้นลักษณะม้วนกลม รอยเส้นแบบกว้าง รอยบุ่ม รอยนูน ระบุตำแหน่งผลเฉลยรอยขีดข่วนด้วยรอยขีดข่วนด้วย Website Roboflow.com ผ่านกระบวนการเพิ่มภาพ (Augmentation) ได้ภาพจำนวน 2,968 ภาพ

**4.3 ฝึกสอนและพัฒนาด้วยอัลกอริทึมการเรียนรู้เชิงลึกโครงข่ายประสาทเทียม** คอนโวลูชัน (Convolutional Neural Network) ในงานค้นคว้าอิสระนี้ใช้โมเดล YOLOv8 เป็นแม่แบบ 3 เวอร์ชัน คือ YOLOv8n , YOLOv8s , YOLOv8x ฝึกสอนโมเดลด้วย Google Colab ด้วย GPU T4 Memory 16GB

**4.4 พัฒนาส่วนประสานกับผู้ใช้งาน** จะมีการแสดงผลภาพหน้าเข้า และภาพการตรวจจับพร้อมระบุตำแหน่งที่ตรวจจับพบด้วย Python และ PyQt5

#### 4.5 ทดสอบและประเมินประสิทธิภาพของโมเดล

1) ประเมินประสิทธิภาพของโมเดลด้วยค่า *mean Average Precision (mAP50)* เป็นค่าเฉลี่ยของค่า *Average Precision (AP)* ซึ่งค่า AP คือค่าเฉลี่ยของผลการทำนายที่ถูกต้องของโมเดล โดยมีการกำหนดค่า *Threshold IoU (Intersection over Union) = 0.5*

2) เปรียบเทียบประสิทธิภาพจากตารางชุดทดสอบ เพื่อหาเปอร์เซ็นต์ค่าความผิดพลาดและค่าความแม่นยำ

## 5. นิยามศัพท์เฉพาะ

**5.1 Computer Vision** หมายถึง การนำคอมพิวเตอร์มาฝึกสอนเพื่อให้สามารถวิเคราะห์แยกแยะวัตถุภายในภาพหรือวิดีโอให้มองเห็นภาพหรือวิดีโอเช่นเดียวกับมนุษย์

**5.2 Artificial Intelligence** หมายถึง ความสามารถของเครื่องคอมพิวเตอร์ในการจำลองและนำเสนอความคิดและการกระทำที่มนุษย์ทำได้ ซึ่งเป็นส่วนหนึ่งของด้านปัญญาประดิษฐ์

**5.3 Deep Learning** หมายถึง ซอฟต์แวร์คอมพิวเตอร์ที่มีการเรียนรู้จากข้อมูลและปรับความสามารถการเรียนรู้จากข้อมูลที่ได้รับคล้ายคลึงกับโครงข่ายประสาทของมนุษย์

**5.4 Convolution Neural Networks** หมายถึง โครงข่ายประสาทแบบคอนโวลูชัน เป็นเครื่องมือในการประมวลผลภาพและวิดีโอที่ทำการเลียนแบบการมองเห็นของมนุษย์

**5.5 Feature Extraction** หมายถึง การสกัดคุณลักษณะวัตถุภายในภาพหรือวิดีโอ เพื่อให้ได้คุณลักษณะของวัตถุนั้นเพื่อนำข้อมูลไปประมวลผล

**5.6 YOLOv8** หมายถึง เวอร์ชันที่ 8 ของโมเดลการตรวจจับวัตถุในภาพที่เรียกว่า YOLO (You Only Look Once)

**5.7 Accuracy** หมายถึง ความแม่นยำในการให้คำตอบของโมเดล จะให้ข้อมูลเกี่ยวกับประสิทธิภาพของระบบในการทำนายทั้งหมด

**5.8 Precision** หมายถึง ค่าความแม่นยำทางเชิงบวก เป็นหนึ่งในค่าวัดประสิทธิภาพของระบบการจำแนกหรือตรวจจับ

**5.9 Recall** หมายถึง ค่าความครบถ้วนที่โมเดลให้คำตอบในแต่ละคลาสครบถ้วนหรือไม่

**5.10 Mean Average Precision (mAP50)** หมายถึง ค่าเฉลี่ยของค่า Average Precision (AP) ซึ่งค่า AP คือค่าเฉลี่ยของผลการทำนายที่ถูกต้องของโมเดลในแต่ละรอบการฝึกสอน โดยมีการกำหนดค่า Threshold IoU (Intersection over Union) = 0.5

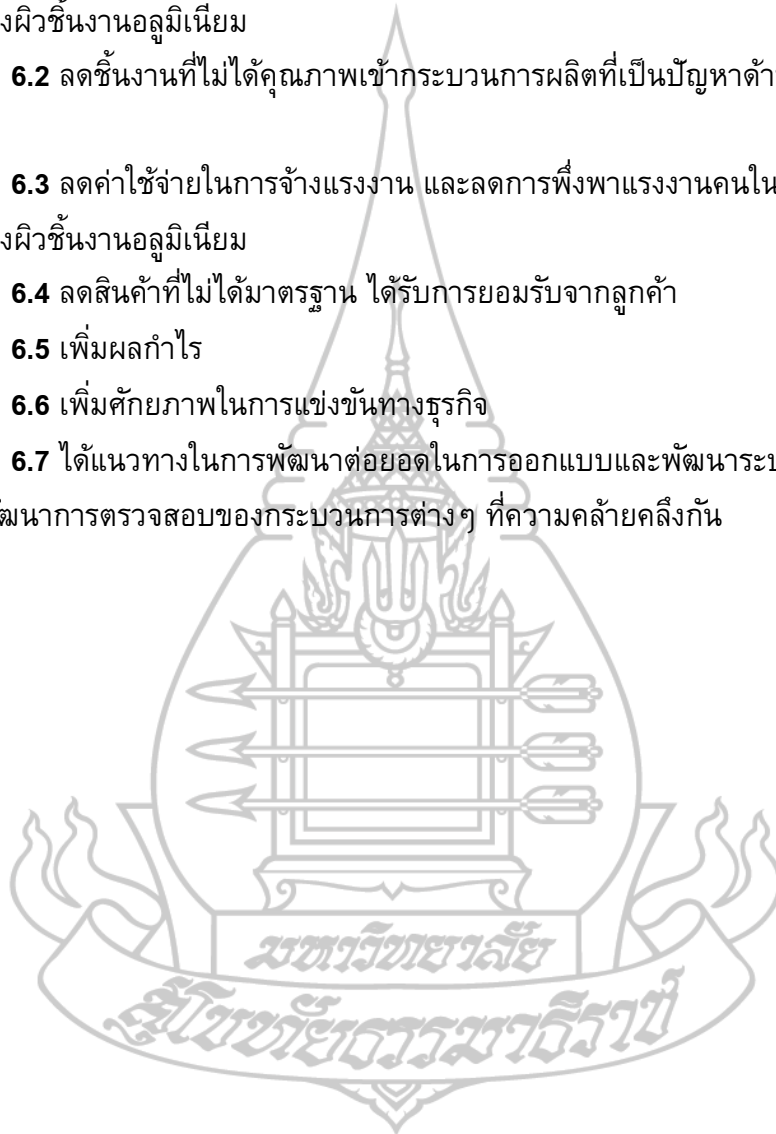
**5.11 Mean Average Precision (mAP50-95)** หมายถึง ค่าเฉลี่ยของค่า Average Precision (AP) ซึ่งค่า AP คือค่าเฉลี่ยของผลการทำนายที่ถูกต้องของโมเดลในแต่ละรอบการฝึกสอน โดยมีการกำหนดค่า Threshold IoU (Intersection over Union) = 0.5 ถึง 0.95

**5.12 Iteration** หมายถึง จำนวนรอบในการฝึกสอนโมเดลซ้ำจากชุดข้อมูลทั้งหมด

**5.13 Class** หมายถึง ประเภทหรือชนิดของวัตถุที่นำไปฝึกสอนโมเดล

## 6. ประโยชน์ที่คาดว่าจะได้รับ

- 6.1 เพื่อช่วยลดภาระการทำงานทางสายตาของคนในตำแหน่งการตรวจสอบคุณภาพของผิวชิ้นงานอลูมิเนียม
- 6.2 ลดชิ้นงานที่ไม่ได้คุณภาพเข้ากระบวนการผลิตที่เป็นปัญหาด้านคุณภาพของสินค้า
- 6.3 ลดค่าใช้จ่ายในการจ้างแรงงาน และลดการพึ่งพาแรงงานคนในการตรวจสอบคุณภาพของผิวชิ้นงานอลูมิเนียม
- 6.4 ลดสินค้าที่ไม่ได้มาตรฐาน ได้รับการยอมรับจากลูกค้า
- 6.5 เพิ่มผลกำไร
- 6.6 เพิ่มศักยภาพในการแข่งขันทางธุรกิจ
- 6.7 ได้แนวทางในการพัฒนาต่อยอดในการออกแบบและพัฒนาระบบการตรวจสอบคุณภาพ พัฒนาการตรวจสอบของกระบวนการต่างๆ ที่ความคล้ายคลึงกัน



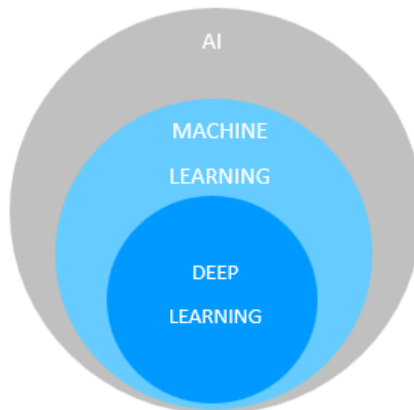
## บทที่ 2

### วรรณกรรมที่เกี่ยวข้อง

การศึกษาค้นคว้า การออกแบบและพัฒนาระบบตรวจจับรอยขีดข่วน และระบุตำแหน่งบนผิวโลหะอลูมิเนียม โดยใช้โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network) ด้วยอัลกอริทึม YOLOv8 (You Only Look Once) กรณีศึกษาในกระบวนการตรวจสอบคุณภาพวัตถุดิบในโรงงานอุตสาหกรรมผลิตกล่องผลิตภัณฑ์ของสินค้าแบรนด์เนมราคาแพง ผู้ศึกษาได้ศึกษา แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อมาสนับสนุน การสร้างกรอบแนวคิดในการศึกษา โดยมีรายละเอียดในการนำเสนอต่อไปนี้

1. การเรียนรู้เชิงลึก (Deep Learning)
2. โครงข่ายประสาทแบบคอนโวลูชัน (Convolution Neural Network)
3. การตรวจจับวัตถุ (Object Detection)
4. การวัดผลประสิทธิภาพโมเดล
5. งานวิจัยที่เกี่ยวข้อง

#### 1. การเรียนรู้เชิงลึก (Deep Learning)

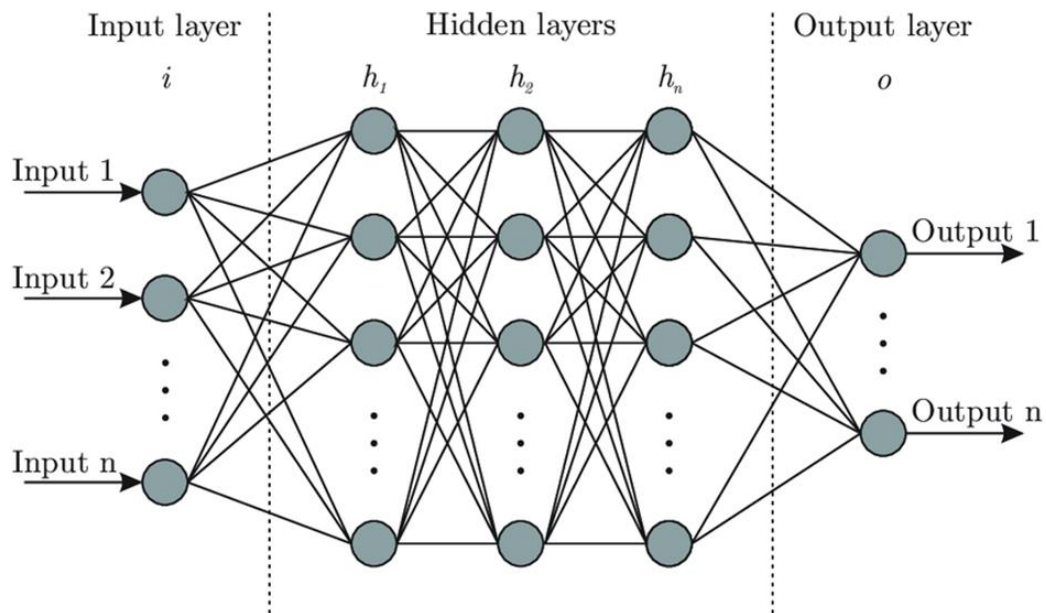


ภาพที่ 2.1 ความสัมพันธ์ของ AI Machine Learning และ Deep Learning

ปัญญาประดิษฐ์ (Artificial Intelligence (AI)) หมายถึง ความสามารถของระบบหรือโปรแกรมที่ถูกสร้างขึ้นให้มีความสามารถทำงานเสมือนหรือคล้ายคลึงกับการคิดของมนุษย์ ให้ประมวลผล คิด ตัดสินใจ เรียนรู้ ได้ด้วยตัวเองโดยใช้เทคนิคและอัลกอริทึมต่างๆ ทางคณิตศาสตร์ ปัจจุบันมีการพัฒนา AI มีให้มีความสามารถหลากหลายด้าน เช่น การเข้าใจภาษาธรรมชาติ (Natural Language Processing) การเข้าใจเสียงพูดและประมวลผลภาษาของมนุษย์ (Speech and Language Processing) การเข้าใจคุณลักษณะของภาพคล้ายคลึงกับความความสามารถในการมองเห็นของมนุษย์ (Computer Vision) ระบบอัตโนมัติให้หุ่นยนต์ทำงานตามหน้าที่ต่างๆ โดยเลียนแบบการกระทำของมนุษย์ (Robotic Processes Automation) ระบบผู้เชี่ยวชาญมีความสามารถตัดสินใจอย่างมนุษย์ (Expert Systems)

การเรียนรู้ของเครื่องจักร (Machine Learning (ML)) เป็นส่วนหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligence (AI)) ที่เน้นการเรียนรู้โดยที่คอมพิวเตอร์จะเรียนรู้และปรับปรุงความสามารถได้เอง โดยเรียนการเรียนรู้จะคล้ายคลึงกับการเรียนรู้ของสมองมนุษย์ ที่มีการเชื่อมต่อเส้นประสาท (Neural) เข้าเป็นโครงข่ายใยประสาท (Neural Network) โดยการสอนจะมี 2 ประเภท คือ การเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยรูปแบบนี้จะมีการสร้างคำตอบไว้ในชุดของข้อมูลที่ป้อนเข้าไปฝึกสอน (Label) และอีกประเภท คือการเรียนรู้แบบไม่มีผู้สอน (Unsupervised Learning) เพื่อให้โมเดลจำแนกและประมวลผลข้อมูลได้เอง

การเรียนรู้เชิงลึก (Deep Learning (DL)) เป็นส่วนหนึ่งของการเรียนรู้ของเครื่องจักร (Machine Learning) โดยที่คอมพิวเตอร์จะเรียนรู้และปรับปรุงความสามารถได้เอง โดยเรียนการเรียนรู้จะคล้ายคลึงกับการเรียนรู้ของสมองมนุษย์ ที่มีการเชื่อมต่อประสาท (Neural) เป็นโครงข่ายเชื่อมเข้าหากันหลายชั้น (Deep Neural Network) มีชั้น Hidden Layer หลายชั้น ซึ่งมีความแม่นยำมากกว่าการเรียนรู้ของเครื่องจักร (Machine Learning) การเรียนรู้เชิงลึก ประกอบไปด้วย 3 ส่วนได้แก่ 1) ชั้นรับข้อมูล (Input Layer) 2) ชั้นการเรียนรู้และประมวลผลของโมเดล (Hidden Layer) 3) ชั้นผลลัพธ์ (Output Layer) ซึ่งการประมวลผลในชั้น Hidden Layer ดังแสดงในภาพที่ 2.2 โครงสร้างของ Deep Neural Network แบบหลายชั้น



ภาพที่ 2.2 โครงสร้างของ Deep Neural Network แบบหลายชั้น

ที่มา: iaviral.medium (2023) An introduction to machine learning algorithms.

การทำงานของการเรียนรู้เชิงลึก (Deep Learning) จะมีการรับค่า Input เข้ามาคูณกับค่าพารามิเตอร์ที่ใช้ในการปรับน้ำหนักของข้อมูลนำเข้า (Weight) ในแต่ละเส้นเชื่อมของแต่ละโหนด และมีการปรับค่าด้วยค่าเอนเอียง (Bias) เพื่อให้โมเดลมีความสามารถในการปรับปรุงการเรียนรู้ให้ดีขึ้น โดยฟังก์ชันที่ใช้คำนวณนี้เรียกว่า Activation Function โดย Activation Function มีอยู่หลายสมการ เช่น Linear, Sigmoid, LeRu เป็นต้น โดยในแต่ละชั้นของ Hidden Layer มีการคำนวณที่เป็นอิสระจากกัน การนำข้อมูลจำนวนมากเข้าไปฝึกสอนโมเดล โมเดลจะมีการปรับแต่งค่า Weight และค่า Bias ในแต่ละรอบการฝึกสอนเองทำให้มีความแม่นยำมากขึ้นกว่ามีข้อมูลน้อยในการฝึกสอน

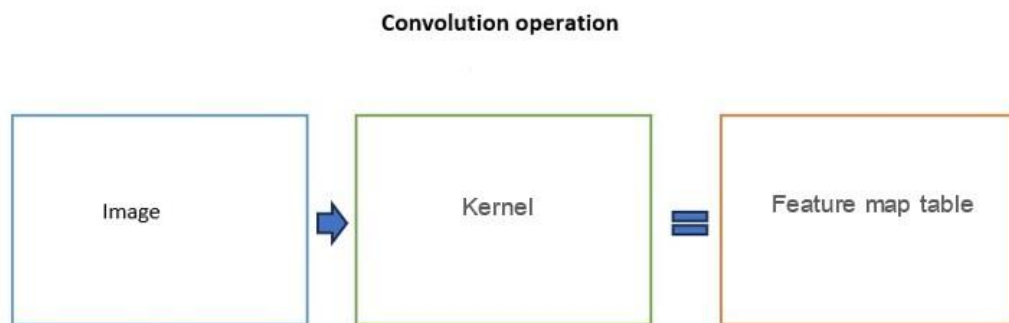
## 2. โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network (CNN))

เป็นอัลกอริทึม Deep Learning ออกแบบมาทำงานในด้านคอมพิวเตอร์วิชัน สำหรับการประมวลผลภาพ โดยการจำลองการมองเห็นให้คอมพิวเตอร์มองเห็นคล้ายคลึงกับการมองเห็นในมนุษย์ ซึ่งเป็นการมองวัตถุที่คุณลักษณะของวัตถุ โดยมีการทำงานอยู่สองส่วนหลักๆ คือ ส่วนของคอนโวลูชัน (Convolution) ใช้ในการสกัดคุณลักษณะของคลาสที่ต้องการและลด

ขนาดของข้อมูล และส่วนของโครงข่ายการเรียนรู้เชิงลึก (Deep Neural Network) ใช้ในการประมวลผลผลลัพธ์

## 2.1 คอนโวลูชัน (Convolution)

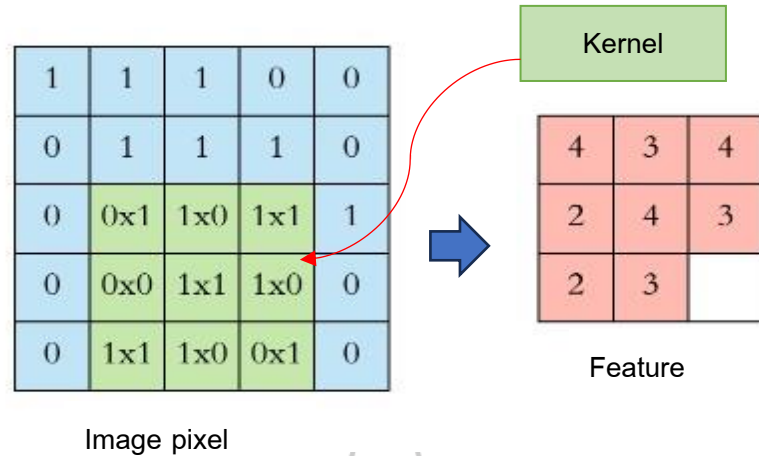
เป็นกระบวนการในการสกัดเอาคุณลักษณะเฉพาะของวัตถุที่สนใจในภาพ ซึ่งเป็นกระบวนการหนึ่งที่กำลังการมองเห็นให้เหมือนมนุษย์ ซึ่งเมื่อมนุษย์มองวัตถุใดๆ จะมองที่คุณลักษณะที่สำคัญ เช่น มองไปที่รถจักรยานยนต์จะมองที่คุณลักษณะสำคัญ คือมีสองล้อ มีเครื่องยนต์ มีท่อไอเสีย มีแฮนด์บังคับ ในกระบวนการคอนโวลูชันก็เช่นเดียวกัน คือมีข้อมูลคุณลักษณะที่สำคัญของวัตถุ แล้วนำมาซ้อนทับไปบนภาพและเลื่อนไปทุกส่วนบนภาพโดยกระบวนการหาคุณลักษณะของวัตถุนี้ว่า การสกัดคุณลักษณะของวัตถุ (Feature Extraction) ซึ่งมีขั้นตอนกระบวนการสกัด ดังภาพที่ 2.3



ภาพที่ 2.3 ขั้นตอนของกระบวนการสกัดคุณลักษณะของวัตถุ

กระบวนการนี้จะมีการสร้างตัวกรองตารางคุณลักษณะ เคอร์เนล (Kernel) ที่ได้จากการฝึกสอนเป็นตัวกรองที่มีคุณลักษณะที่ต้องการ โดยเคอร์เนลจะเป็นตารางเมทริกซ์ ขนาด  $3 \times 3$  หรือ ขนาด  $k \times k$  วางทับลงบนเมทริกซ์ของข้อมูลรูปภาพ และมีการเลื่อนเมทริกซ์จนครบทุกพิกเซลของรูปภาพ ระยะการเลื่อนเคอร์เนลนี้เรียกว่า Stride โดยค่า Stride นี้เป็นตัวกำหนดว่าจะเลื่อนไปครั้งละกี่ช่องของเมทริกซ์รูปภาพ และนำข้อมูลจากเคอร์เนลคูณแบบ Element-wise กับข้อมูลเมทริกซ์ของรูปภาพจากนั้นหาผลรวมของข้อมูลในแต่ละช่องมาบวกกัน และนำผลลัพธ์ที่ได้บันทึกในตาราง Feature map ส่งผลให้ได้ตารางเมทริกซ์ใหม่ที่มีขนาดลดลง ดังภาพที่ 2.4





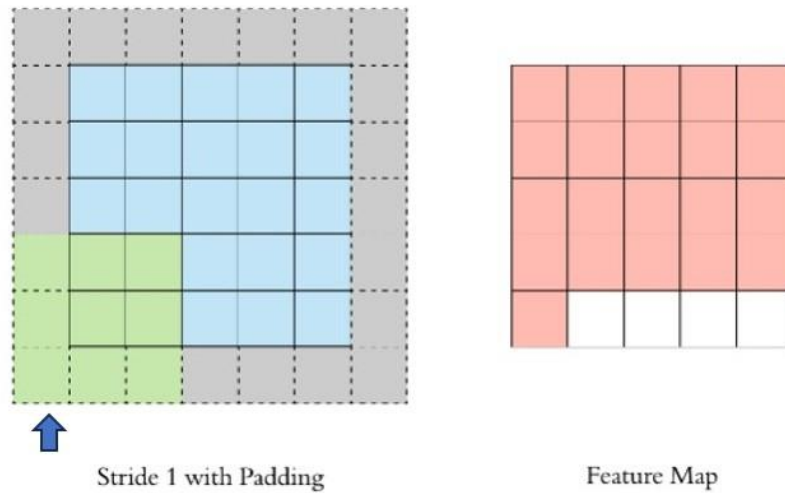
ภาพที่ 2.4 การทำงานของตัวกรองเคอร์เนล

กระบวนการ การทำ Padding ในคอนโวลูชัน (Convolution) เป็นกระบวนการเพิ่มขอบหรือเสริมพื้นที่ที่รอบกรอบของรูปภาพ เพื่อปรับขนาดของภาพก่อนที่นำไปสกัดคุณลักษณะด้วยเคอร์เนลมีวัตถุประสงค์ ดังนี้

- (1) ควบคุมขนาดของข้อมูลที่อาจมีการลดขนาดมากเกินไปทำให้ขาดคุณลักษณะที่สำคัญของวัตถุ
- (2) ช่วยแก้ไขพื้นที่ขอบของภาพให้มีการสกัดคุณลักษณะสำคัญของวัตถุที่ปรากฏอยู่ในตำแหน่งของของภาพ

หากไม่มีการทำ Padding ในกระบวนการคอนโวลูชัน (Convolution) คุณลักษณะที่สำคัญที่ต้องการอาจจะอยู่ตามขอบของภาพ ทำให้เมื่อเข้าสู่การเรียนรู้เชิงลึก (Deep Learning) เพื่อทำการจำแนก หรือตรวจจับวัตถุที่ต้องบนภาพที่มีตำแหน่งอยู่ใกล้ขอบภาพอาจให้ผลลัพธ์ที่ได้ไม่แม่นยำเท่าที่ควร จากภาพที่ 2.5 พื้นที่สีเทา คือพื้นที่ที่เพิ่มเข้าไป การทำ Padding สามารถทำได้ด้วยวิธีการดังนี้

- (1) *Zero Padding* คือการเพิ่มค่าศูนย์เข้าไปในช่องว่างของขอบภาพ โดยให้ค่าศูนย์เป็นค่าคงที่ การใช้ Zero Padding ทำให้ขนาดของ Feature map มีขนาดเท่าเดิม
- (2) *Constant Padding* คือการเพิ่มค่าคงที่ที่ช่องว่างของภาพนำเข้าไป ซึ่งค่าคงที่นี้สามารถกำหนดได้เองตามต้องการ
- (3) *Reflection Padding* คือการสะท้อนภาพที่นำเข้าไปยังพื้นที่ Padding ซึ่งจะทำให้พื้นที่ ที่ได้ขยายมีลักษณะเหมือนกันกับภาพนำเข้าไป ดังภาพที่ 2.6



ภาพที่ 2.5 การเพิ่มพื้นที่ขอบภาพ

### Reflection padding

สะท้อนภาพที่นำเข้าไปยังพื้นที่ Padding

3	5	1
3	6	1
4	7	9

1	6	3	6	1	6	3
1	5	3	5	1	5	3
1	6	3	6	1	6	3
9	7	4	7	9	7	4
1	6	3	6	1	6	3

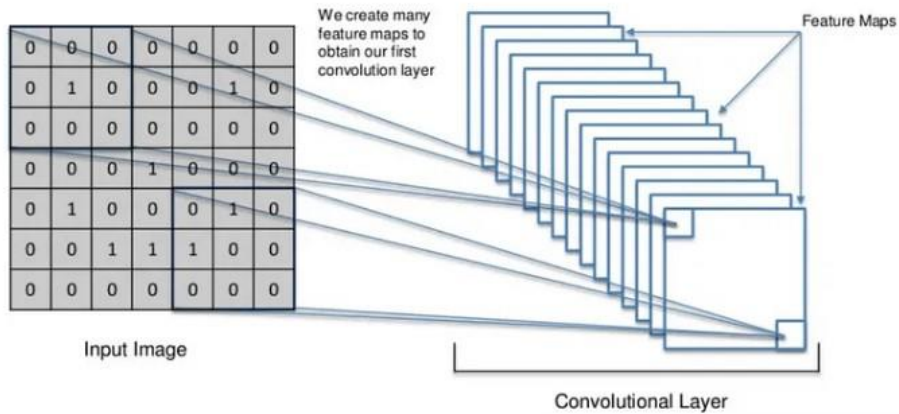
No padding

(1, 2) reflection padding

ภาพที่ 2.6 การทำ Reflection padding

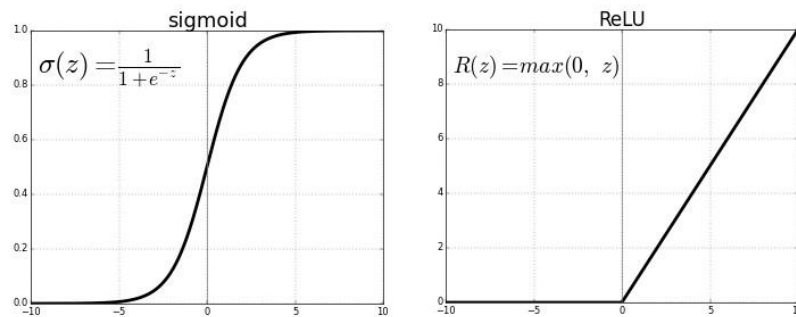
ที่มา: Dev.to (2023) Padding in Neural Network.

เมื่อทำการสกัดคุณลักษณะในหลายคุณลักษณะของคลาสที่ต้องการ จะได้ Convolution Layer ที่มีหลาย Feature map ดังภาพที่ 2.7



ภาพที่ 2.7 แสดง Convolution Layer ที่ได้จากการทำ Feature maps  
ที่มา: medium.com (2023) Convolution Neural Network.

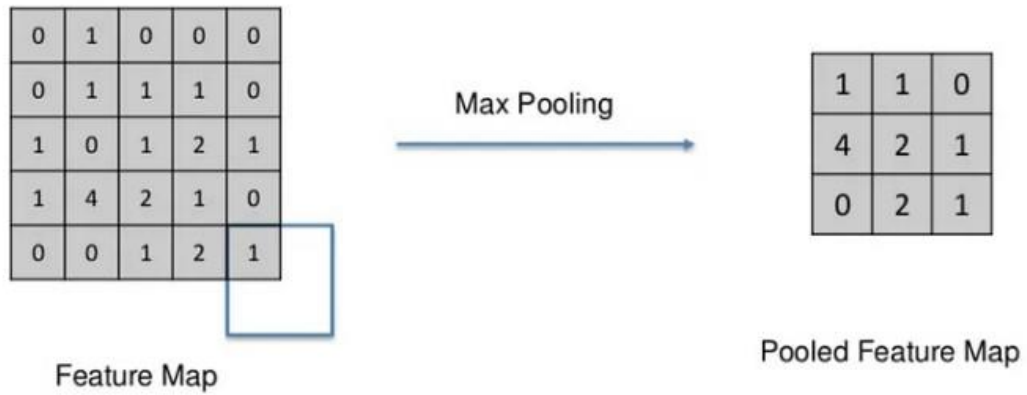
หลังจากกระบวนการสกัดคุณลักษณะที่สำคัญจะได้ผลลัพธ์เป็นข้อมูลตาราง Feature map หลายตาราง ซึ่งจะเรียกว่า Convolution Layer จากนั้นจะนำ Convolution Layer เข้ากระบวนการ Activation Function โดยสมการ Rectified Linear Unit (ReLU) หรือ Sigmoid ซึ่งเป็นฟังก์ชันทางคณิตศาสตร์ที่ใช้ในโครงข่ายประสาทเทียม (Neural Network) เพื่อลดการเกิด Vanishing Gradient ทำให้เพิ่มความเร็วในการฝึกสอน ซึ่งค่าผลลัพธ์ของสมการ ReLU จะมีค่าตั้งแต่ศูนย์ถึงอินฟินิตี้ (Infinity) ข้อมูลใดๆ ที่เข้าสมการ ReLU ค่าที่ติดลบจะเปลี่ยนเป็นค่าศูนย์ ค่าที่เป็นค่าบวกจะคงเดิม ส่วนสมการ Sigmoid จะให้ผลลัพธ์ที่มีค่าตั้งแต่ศูนย์ถึงหนึ่ง ดังในภาพที่ 2.8



ภาพที่ 2.8 แสดงสมการ Rectified Linear Unit และสมการ Sigmoid

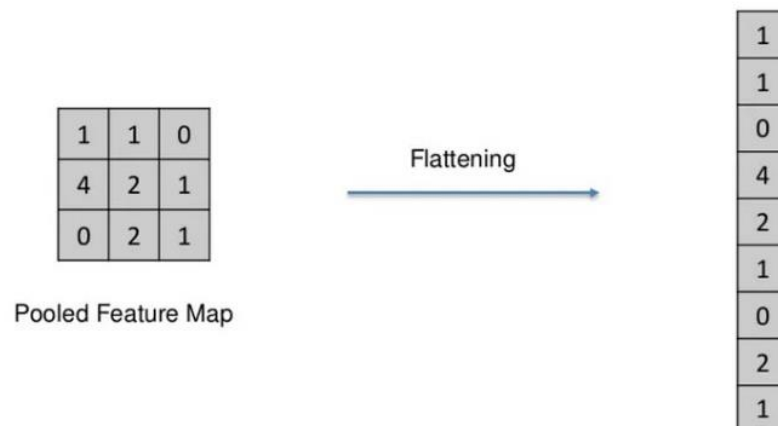
จากนั้นเข้าสู่กระบวนการลดขนาดของข้อมูล (Pooling) ซึ่งการลดขนาดของข้อมูล โดยการสร้างเมทริกซ์ขนาด  $K \times K$  เช่น  $K=2$  ตามภาพที่ 2.9 ซึ่งวิธีการ Pooling มีอยู่ 3 แบบ คือ 1) Max Pooling คือการนำค่าสูงสุดมาเขียนในตาราง Pooled Feature Map 2) Average

Pooling คือการนำค่าเฉลี่ยมาเขียนในตาราง Pooled Feature Map 3) Min Pooling คือการนำค่าต่ำสุดเขียนในตาราง Pooled Feature Map แต่ที่เป็นที่นิยมคือ Max Pooling

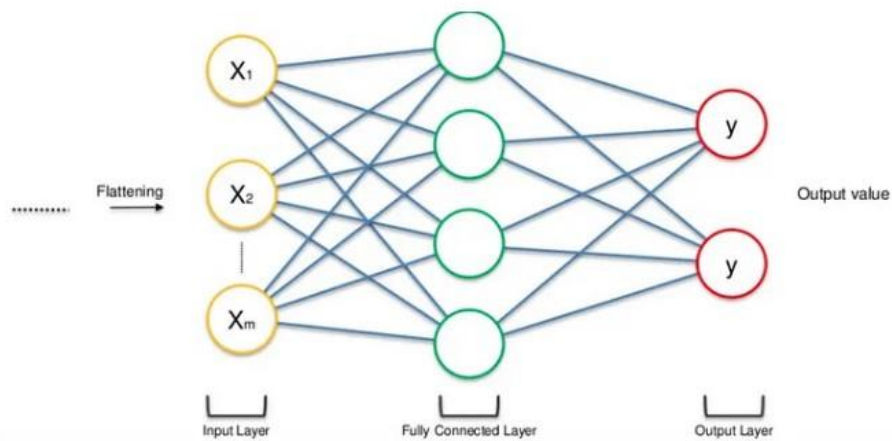


ภาพที่ 2.9 การลดขนาดข้อมูลด้วย Max Pooling

จากนั้นเข้าสู่กระบวนการ Flattening เป็นกระบวนการนำข้อมูล Feature Map ที่ผ่านกระบวนการ Pooled Feature Map แล้วแปลงข้อมูลที่อยู่ในรูปแบบสองมิติให้เป็นแบบหนึ่งมิติก่อนนำเข้าสู่ Fully Connected Layer ดังรูปที่ 2.10



ภาพที่ 2.10 กระบวนการแปลงข้อมูลให้อยู่ในรูปแบบหนึ่งมิติ



ภาพที่ 2.11 แสดงการนำข้อมูลเข้า Fully Connected Layer

## 2.2 โครงข่ายการเรียนรู้เชิงลึก (Deep Neural Network)

เป็นขั้นตอนของการประมวลผลผลลัพธ์ด้วยโครงข่ายการเรียนรู้เชิงลึก (Deep Neural Network) โดยกระบวนการ Fully Connected คือขั้นใช้ในการรวบรวมข้อมูลผลลัพธ์ของชั้นก่อนหน้าที่ได้คุณลักษณะสำคัญของวัตถุที่มีการสกัดมาแล้ว มาทำการประมวลผลให้โมเดลทำนายผลลัพธ์ที่ต้องการ โดยผลลัพธ์ที่ได้จากโมเดล Deep Learning จะเข้าสู่ Activation Function ด้วยฟังก์ชัน Softmax ซึ่งเป็นฟังก์ชันทางคณิตศาสตร์ที่ใช้ในการแปลงค่าของกลุ่มตัวเลขให้อยู่ในรูปของค่าความน่าจะเป็น ซึ่งมักใช้ในการจำแนกหรือการระบุประเภทของวัตถุ หรือคลาสในโมเดลปัญญาประดิษฐ์ (Deep Learning Models) หรือโมเดลที่เกี่ยวข้องกับการจำแนกหมวดหมู่ (Classification) โดยมีค่าผลลัพธ์อยู่ที่ 0-1 โดยมีผลรวมของทุกคลาสเท่ากับ 1 สูตรของฟังก์ชัน Softmax คือ

$$\text{Softmax} = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K$$

โดยมีค่าคงที่ e มีค่าประมาณเท่ากับ 2.71828

## 3. การตรวจจับวัตถุ (Object detection)

ในกระบวนการค้นหาวัตถุที่ต้องการที่อยู่ในภาพหรือวิดีโอ โดยโมเดลหรือคอมพิวเตอร์ที่พัฒนาขึ้นสามารถตรวจจับวัตถุ (Object detection) แยกประเภทวัตถุ (Classification) แยกกลุ่มวัตถุ (Segmentation) พร้อมทั้งระบุตำแหน่งของวัตถุภายในภาพ

โดยโมเดลได้ถูกฝึกสอนให้รู้จักคุณลักษณะที่สำคัญของวัตถุเพื่อค้นหาวัตถุที่ต้องการภายในภาพหรือวิดีโอ กระบวนการตรวจจับวัตถุส่วนใหญ่จะพัฒนาโมเดลด้วย Deep Learning ที่มีความสามารถในการเรียนรู้และสกัดลักษณะของวัตถุเพื่อใช้ในการตรวจจับ

### 3.1 ขั้นตอนหลักของการตรวจจับวัตถุ (Object Detection)

ประกอบไปด้วย

#### 3.1.1 การสกัดคุณลักษณะของวัตถุ (Feature Extraction)

โดยในขั้นตอนนี้โมเดล Deep Learning จะใช้อัลกอริทึม Convolution Neural Network (CNN) ที่กล่าวมาในหัวข้อ 1.2.1 คอนโวลูชัน เพื่อสกัดเอาคุณลักษณะที่สำคัญของวัตถุภายในภาพที่ป้อนเข้ามา

#### 3.1.2 การตัดแยกวัตถุ (Object Localization)

โมเดลจะค้นหาพื้นที่ที่วัตถุอยู่ในภาพ และกำหนดพื้นที่ของวัตถุนั้น โดยใช้กรอบสี่เหลี่ยม (Bounding Box) ติกรอบวัตถุนั้นเพื่อระบุตำแหน่งและขนาดของวัตถุ

#### 3.1.3 การตรวจจับและจำแนกวัตถุ (Object Detection and Classification)

โมเดลจะตรวจจับและจำแนกวัตถุที่ตรวจพบคืออะไร โดยเปรียบเทียบกับคลาสที่ได้ฝึกสอน เช่น คลาสรถยนต์ คลาสรถจักรยานยนต์ คลาสรถบรรทุก เป็นต้น

#### 3.1.4 Non-maximum Suppression การลดกรอบ (bounding boxes)

ในการจับวัตถุที่ซ้ำกันภายในภาพเพื่อให้คงเหลือ bounding boxes รอบวัตถุนั้นเพียงอันเดียว

### 3.2 เทคนิคการตรวจจับวัตถุ

แบ่งออกได้เป็น 2 ประเภทหลักๆ ดังภาพที่ 2.12 1) One-stage object detection  
2) Two-stage object detection

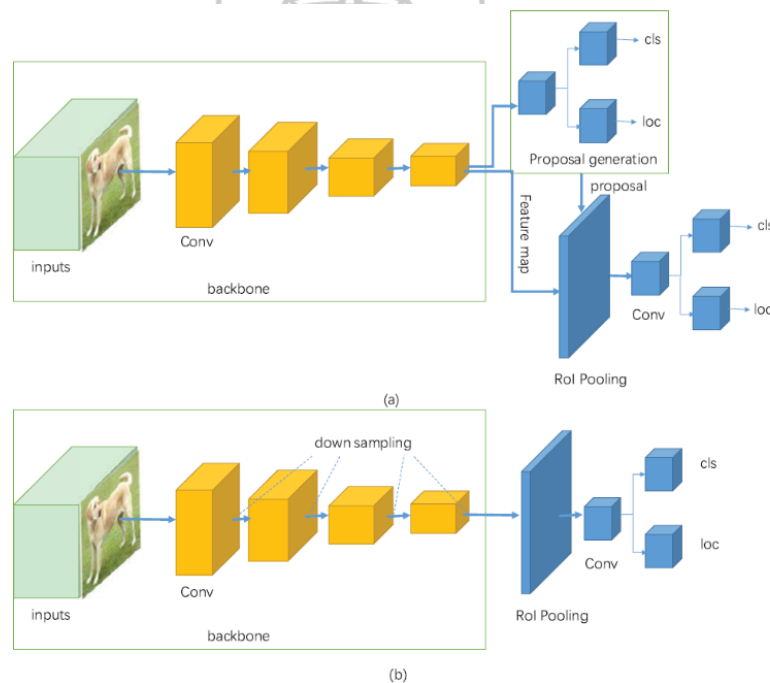
#### 3.2.1 One-stage object detection

คือเทคนิคในการตรวจจับวัตถุ (Object Detection) ที่ใช้วิธีการส่งภาพเข้าโมเดลเพียงครั้งเดียว เพื่อหาคำตอบของคลาสและระบุตำแหน่งของวัตถุที่อยู่ภายในภาพ โดยไม่จำเป็นต้องสร้าง Region Proposals (Region Proposal Network – RPN) หรือขั้นตอนในการเลือกพื้นที่ที่เป็นความเป็นไปได้ของวัตถุ (Region Proposal) ก่อนการทำนาย (Predict) ข้อดี

ของ One-stage object detection คือมีความแม่นยำเร็วในการประมวลผลสูง ซึ่งปัจจุบันมีการพัฒนาโมเดลใหม่ๆ ขึ้นมาหลายเวอร์ชัน เช่น SDD (Single Shot Multi-Box Detector), YOLO (You Only Look Once) ซึ่งเป็นรุ่นล่าสุดในปัจจุบันพัฒนาไปถึงเวอร์ชันที่ 8 (YOLOv8) ที่เป็นโมเดลที่ได้รับความนิยมในการทำงานด้านการตรวจจับวัตถุภายในรูปภาพ (Object detection)

### 3.2.2 Two-stage object detection

คือเทคนิคในการตรวจจับวัตถุ (Object Detection) มีขั้นตอนในการทำงานสองขั้นตอนหลักในการประมวลผลภาพ โดยการส่งภาพเข้าโมเดลเพื่อตรวจจับวัตถุ แต่ก่อนที่จะมีการทำนาย (Predict) โมเดลจะทำการสร้าง Region Proposals (Region Proposal Network – RPN) ขึ้นมาก่อนเพื่อเลือกพื้นที่ที่เป็นความเป็นไปได้ของวัตถุ (Region Proposal) ซึ่งพื้นที่ที่ถูกเลือกนี้จะส่งไปให้โมเดลทำนายต่อในขั้นตอนถัดไป จุดเด่นของประเภทนี้คือมีความแม่นยำที่สูงมาก แต่มีข้อจำกัดเรื่องของความเร็วในการประมวลผล โมเดลที่ใช้เทคนิคนี้ เช่น R-CNN, Fast R-CNN และ Faster R-CNN



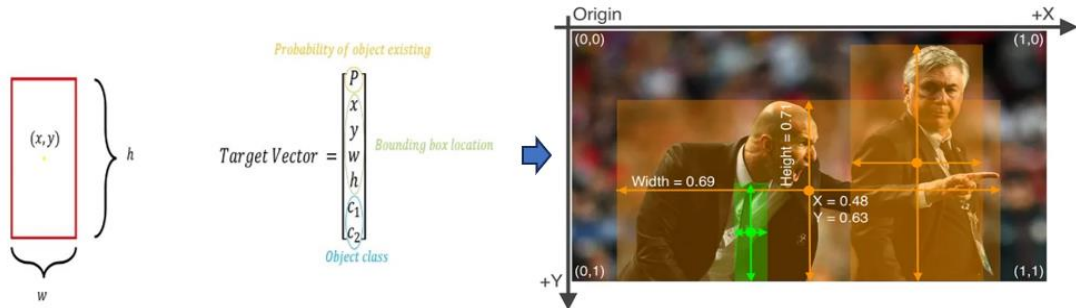
ภาพที่ 2.12 เปรียบเทียบสถาปัตยกรรมเทคนิคในการตรวจจับวัตถุ

ที่มา: [ieeexplore.ieee.org](http://ieeexplore.ieee.org) (2023) A Survey of Deep Learning-Based Object Detection.

### 3.3 YOLO (You Only Look Once) version 8

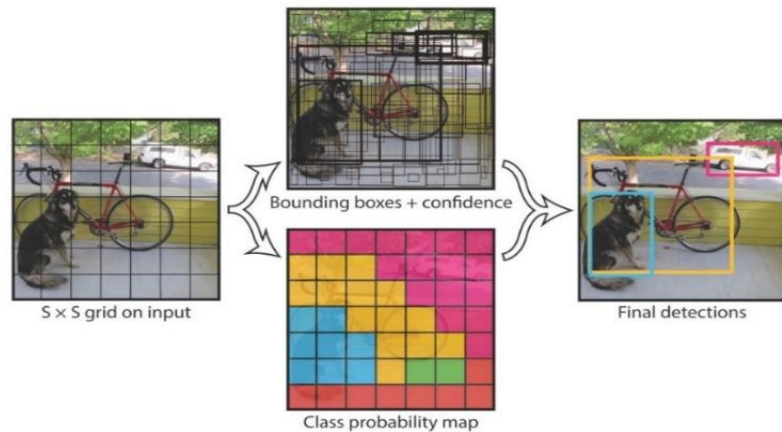
YOLO พัฒนาจากข้อจำกัดของการตรวจจับวัตถุแบบแบบ Two-stage object detection ที่ปัญหาด้านความเร็ว โดยอัลกอริทึม Yolo มีการพัฒนาแบบ One-stage object detection เพื่อ

เพิ่มประสิทธิภาพของความเร็ว YOLO version 1 ถูกพัฒนาขึ้นโดยนักวิจัยชื่อ Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi และได้รับการเผยแพร่ครั้งแรกในบทความที่ชื่อ "You Only Look Once: Unified, Real-Time Object Detection" ในปี 2016 ที่งานประชุมด้านคอมพิวเตอร์วิชั่นความสามารถในการแสดงผล (Conference on Computer Vision and Pattern Recognition - CVPR)



ภาพที่ 2.13 การตีกรอบ Bounding Box

หลักการงานในการตรวจจับวัตถุของ YOLO เมื่อป้อนภาพอินพุตเข้าสู่โมเดล โมเดลจะทำการตีกรอบล้อมวัตถุเรียกว่าการทำ Bounding Box โดยจะมีค่าตัวแปรพารามิเตอร์ คือ  $(P, x, y, w, h, c)$  จากภาพที่ 2.13 การทำ Bounding Box จะวัดจากจุดกึ่งกลางของวัตถุ โดยมีการกำหนดค่าพิกัดจุดกึ่งกลางของวัตถุในแนวแกน  $x$  และแนวแกน  $y$  พร้อมด้วยค่าความกว้างกล่อง  $w$  ค่าความสูงของกล่อง  $h$   $P$  คือค่าความน่าจะเป็น (Probability) ที่มีวัตถุอยู่ใน Grid cell นั้น  $c_1, c_2, \dots, c_n$  คือค่าผลลัพธ์ของวัตถุอยู่ในคลาสอะไร การทำนายในหนึ่งภาพจะให้คำตอบได้เพียงหนึ่งคำตอบเท่านั้น หากต้องการตรวจจับวัตถุหลายวัตถุภายในหนึ่งภาพนั้นจะต้องทำการแบ่งภาพ ดังภาพที่ 2.14

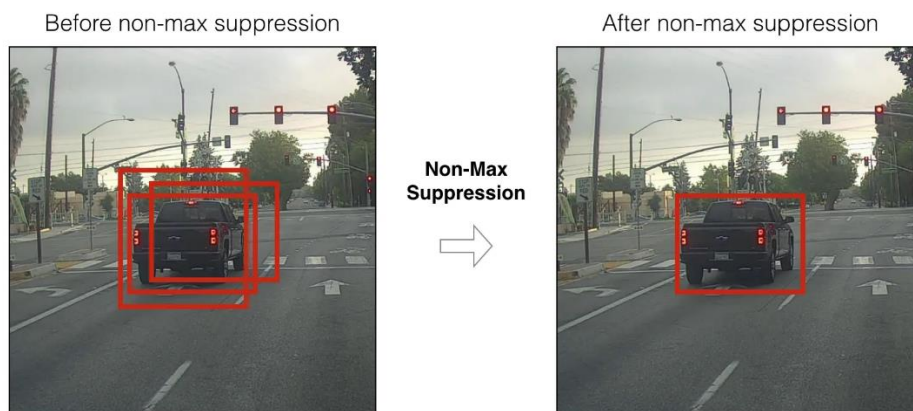


ภาพที่ 2.14 หลักการทำงานของอัลกอริทึม YOLO

ที่มา : labellerr.com (2023) Importance of the YOLO algorithm.



การทำงานของ YOLO (Joseph Redmon, et al., 2016) จะมีการแบ่งภาพอินพุตที่ป้อนเข้าสู่โมเดลออกเป็นส่วนย่อยๆ โดยแบ่งเป็นขนาด  $S \times S$  เช่น แบ่งขนาด  $7 \times 7$  ดังตัวอย่างในภาพที่ 2.14 เนื่องจากภาพหนึ่งภาพสามารถจำแนกวัตถุให้ผลลัพธ์ได้แค่หนึ่งวัตถุต่อหนึ่งภาพ หากต้องการจำแนกวัตถุหลายวัตถุภายในหนึ่งภาพนั้นจะต้องใช้เทคนิคการแบ่งภาพเป็นส่วนย่อยๆ เรียกชื่องเหล่านี้ว่า Grid cell โดยแต่ละ Grid cell จะทำนายวัตถุได้หนึ่งคลาสเท่านั้น จากภาพ 2.14 ด้านซ้ายจะพบว่าสุนัข จักรยาน และรถยนต์จะอยู่ใน Grid cell ที่แยกจากกัน อาจมีบาง Grid cell ที่มีวัตถุหลายวัตถุอยู่ใน Grid cell เดียวกัน แต่โมเดลก็ทำนายร่วมกับองค์ประกอบของ Grid cell อื่น ในขั้นตอนการทำงานลึกลงไปจะมีการปรับค่า Grid cell ให้มีขนาดเล็กลงไปเรื่อยๆ เพื่อให้ผลลัพธ์ในการทำนายที่แม่นยำที่สุด เมื่อทำการตีกรอบ Bounding Box ดังภาพที่ 2.14 ตรงกลางด้านบนจะพบว่าวัตถุหนึ่งวัตถุมีการทำ Bounding Box ซ้ำกันหลายอัน โมเดลจะตัด Bounding Box ของวัตถุที่สนใจให้เหลือวัตถุละหนึ่ง Bounding Box เท่านั้น ด้วยเทคนิค Non-Maximum Suppression (NMS) โดยการคำนวณค่า Intersection over Union (IoU) เพื่อลดความซ้ำซ้อนของกรอบ Bounding Box ในกระบวนการ Non-Maximum Suppression ค่าความน่าจะเป็น (Confidence scores) ของกรอบจะถูกนำมาเปรียบเทียบกับกลุ่มของกรอบที่มีการซ้อนทับกันอยู่ และจากนั้นกำจัดกรอบที่มีความน่าจะเป็นต่ำที่สุดในกลุ่มนั้นๆ ซึ่งเหลือเพียงกรอบที่มีความน่าจะเป็นสูงสุดเท่านั้น ส่วนกรอบอื่น ๆ ที่ไม่ถูกเลือกจะถูกลบทิ้งไป ภาพผลลัพธ์เมื่อเข้ากระบวนการ Non-Maximum Suppression แล้วแสดงในภาพที่ 2.14 ด้านขวา และภาพที่ 2.15



ภาพที่ 2.15 ผลลัพธ์หลังจากกระบวนการ Non-Maximum Suppression

ที่มา: lengyi.medium.com (2023) Non-maximum suppression using python.

การทำงานในแต่ละชั้นของสถาปัตยกรรม YOLOv8 ดังภาพที่ 2.16 มีการทำงานของแต่ละชั้นดังนี้ P1-P5 หมายถึง Panet ซึ่งเป็นส่วนหนึ่งของโครงสร้างโครงข่ายที่ใช้ในการปรับปรุงความละเอียดและการรวมข้อมูลที่เกี่ยวข้องในการตรวจจับวัตถุ ใช้ในกระบวนการปรับปรุงข้อมูลขนาดใหญ่หรือข้อมูลที่มีความละเอียดต่าง ๆ ซึ่งจะช่วยให้ประสิทธิภาพในการตรวจจับวัตถุ การใช้ชั้น P1-P5 ช่วยให้ YOLOv8 สามารถตรวจจับวัตถุในหลายระดับของความละเอียดภาพ และทำให้มีประสิทธิภาพในการตรวจจับวัตถุที่หลากหลายขนาดและที่ต่างกันในภาพหรือวิดีโอได้อย่างเหมาะสม จากนั้นส่งข้อมูลเข้าไปยังชั้น Conv (Convolutional Layer) เพื่อใช้สกัดคุณลักษณะที่สำคัญของภาพจากนั้นนำข้อมูลคุณลักษณะที่ได้เข้าสู่ชั้น C2f หมายถึง SPDarknet53-C2 เป็นชั้นที่ถูกออกแบบมาเพื่อสกัดลักษณะ (feature extraction) จากภาพในขั้นตอนแรกของโครงสร้าง YOLOv8 โดยใช้สถาปัตยกรรม Darknet53-CSP (CSP หมายถึง "Cross-Stage Partial Networks") ซึ่งเป็นส่วนหนึ่งของโครงสร้าง Darknet ที่ถูกปรับปรุงและปรับแต่งให้มีประสิทธิภาพสูงในการสกัดลักษณะของภาพ จากนั้นส่งผลลัพธ์เข้าสู่ส่วนของ Neck ซึ่งเป็นส่วนหนึ่งของโครงสร้างของโมเดล YOLOv8 ที่มีหน้าที่ในการปรับปรุงและรวมข้อมูลที่ได้จากส่วนหน้า (backbone) ของโมเดลเพื่อให้สามารถทำนายและตรวจจับวัตถุในภาพหรือวิดีโอได้อย่างถูกต้องและมีประสิทธิภาพสูง ส่วนของ Neck ใน YOLOv8 มีหน้าที่สำคัญดังนี้

1) *Feature Fusion: Neck* จะรวมข้อมูลจากหลายระดับของความละเอียดที่ได้จากส่วนหน้าของโมเดล เช่น ข้อมูลจากชั้น C2f เพื่อสร้างแผนผังลักษณะ (feature map) ที่มีข้อมูลจากทุกระดับ เพื่อให้สามารถตรวจจับวัตถุในหลายความละเอียด

2) *Feature Refinement: Neck* อาจมีการปรับปรุงและเลือกข้อมูลที่สำคัญในแผนผังลักษณะ และอาจมีการใช้เทคนิคต่าง ๆ เช่น การใช้คอนโวลูชันหลายความละเอียด (multi-scale convolution) เพื่อปรับปรุงคุณภาพข้อมูล

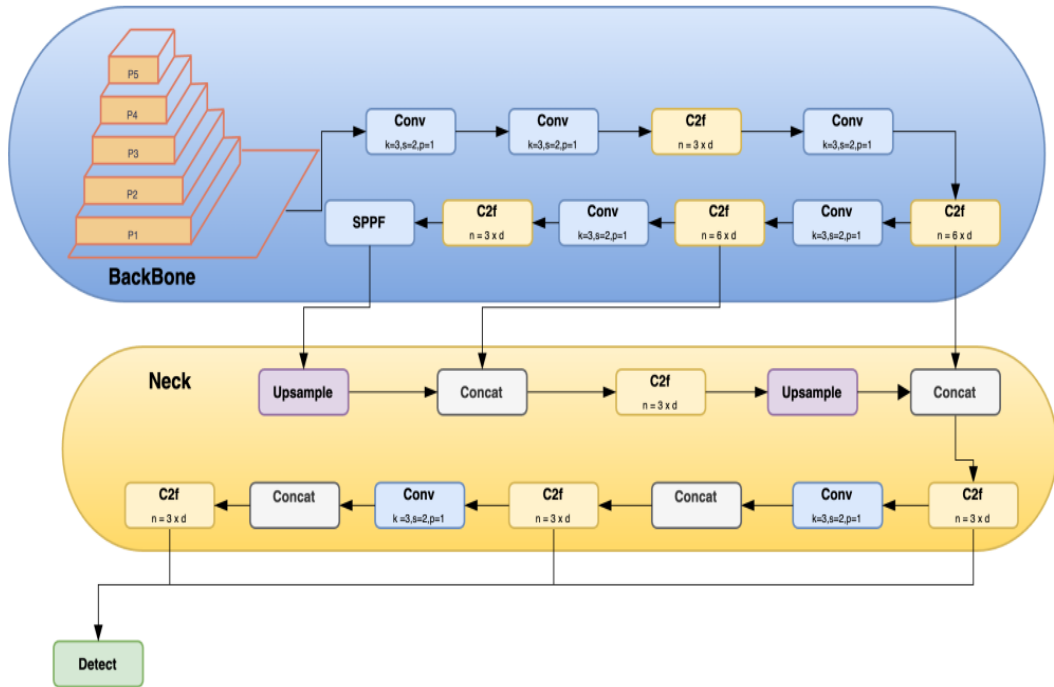
3) *Prediction:* หลังจากการปรับปรุงและรวมข้อมูลใน Neck เสร็จสิ้น โมเดล YOLOv8 จะใช้ข้อมูลที่ผ่านการปรับปรุงใน Neck เพื่อทำการทำนาย (prediction) ของการตรวจจับวัตถุ ซึ่งสามารถให้ผลลัพธ์การตรวจจับวัตถุออกมาในรูปแบบของ bounding boxes และคลาสของวัตถุ

เหตุผลในการนำ YOLOv8 ไปใช้ในงานตรวจจับรอยขีดข่วนบนผิวอลูมิเนียมคือ

1) ประสิทธิภาพสูง YOLOv8 มีประสิทธิภาพการตรวจจับวัตถุที่สูงและมีความเร็วในการทำงานเกือบเท่าเวลาจริง (real-time) ซึ่งเหมาะงานการตรวจจับชิ้นงานผ่านสายพานการผลิต

2) ความยืดหยุ่นในการใช้งาน YOLOv8 มีความยืดหยุ่นในการปรับแต่ง เลือกสถาปัตยกรรมความซับซ้อนให้เหมาะสมกับงานที่ต้องการ

- 3) ความสามารถในการทำงานในสภาพแวดล้อมจริง YOLOv8 มีความสามารถในการทำงานในสภาพแวดล้อมจริงที่มีการเปลี่ยนแปลงของแสงที่ตกกระทบชิ้นงานได้ดี
- 4) มีข้อมูลสนับสนุนจากนักพัฒนาหลากหลาย ทำให้ง่ายในการใช้งานหรือแก้ไขปัญหาต่างๆ

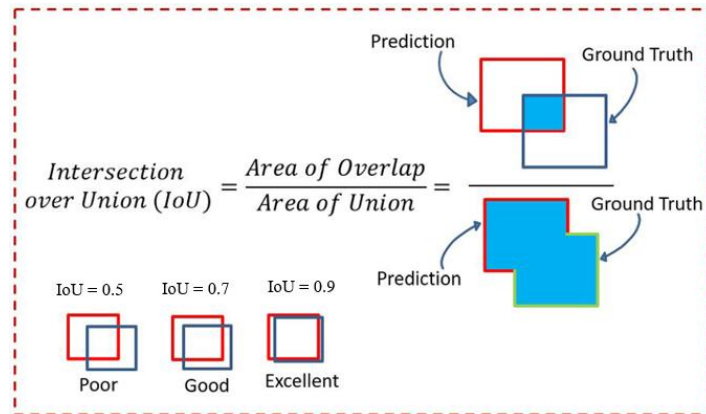


ภาพที่ 2.16 สถาปัตยกรรมของอัลกอริทึม YOLOv8

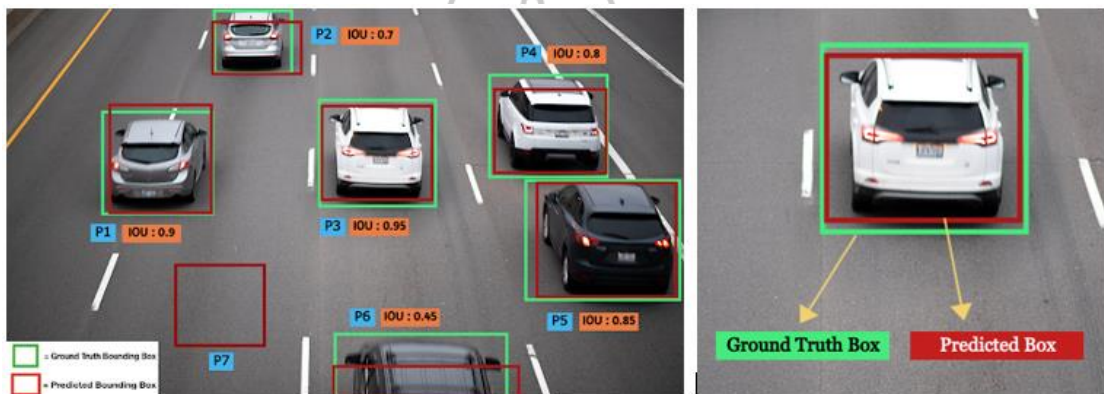
ที่มา: Nabin Sharma et al. (2023) Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms.

#### 4. การวัดผลประสิทธิภาพโมเดล

การวัดประสิทธิภาพโมเดลในการการตรวจจับวัตถุ (Object Detection) Intersection over Union (IoU) เป็นวิธีที่นิยมใช้ เพื่อประเมินความเกี่ยวข้องกันระหว่างผลการทำนาย (Predict) Bounding box กับผลเฉลย (Ground truth) ซึ่งหาได้จากพื้นที่ที่มีการทับซ้อนกันของ Bounding box กับ Ground truth หรือเรียกพื้นที่ Intersection หารด้วยพื้นที่รวมของทั้งสอง หากผลลัพธ์ IoU มีค่าเข้าใกล้ 1 แสดงว่าโมเดลทำนาย (Precision) ได้แม่นยำ แสดงถึงพื้นที่ทำนายตรงหรือใกล้เคียงกับพื้นที่ผลเฉลย หากซึ่งผลลัพธ์ไม่ถึง 0.5 แสดงว่าโมเดลทำนายพื้นที่ต่างจากผลเฉลยมากถือว่ายอมรับไม่ได้จะเป็นค่า False Positive หากค่าผลลัพธ์ที่ได้ IoU > 0.5 ถึงจะเป็นที่ยอมรับได้ ดังภาพที่ 2.17



ภาพที่ 2.17 การหาค่า IoU



ภาพที่ 2.18 ภาพตัวอย่างการซ้อนทับของ Ground Truth กับ Predicted Box

ที่มา: v7labs.com (2023) Mean Average Precision (mAP).

การคำนวณค่า IoU จะคำนวณกับชุดข้อมูลที่ใช้ทดสอบทั้งหมดแล้วนำมาคำนวณหาค่าเฉลี่ยของค่า IoU เพื่อประเมินประสิทธิภาพของโมเดลว่ามีประสิทธิภาพเพียงพอหรือไม่ ค่า IoU ควรมากกว่า 0.5 โดยคำนวณได้จากสมการ

$$IoU = \frac{\text{intersection}}{\text{union}}$$

Mean Average Precision (mAP) คือค่าเฉลี่ยของค่า Average Precision (AP) ของแต่ละคลาสแล้วนำค่า AP ทุกคลาสมาเฉลี่ยเข้าด้วยกันเพื่อให้ได้ค่า mAP ที่เป็นตัวบ่งชี้ความสามารถของโมเดลในการทำนายคลาสหรือวัตถุที่แตกต่างกันในภาพ โดยมีกระบวนการหาค่า mAP ดังนี้

- 1) คำนวณค่า Precision และ Recall ของโมเดลที่ตรวจจับวัตถุในแต่ละคลาส
- 2) คำนวณค่า AP ของแต่ละคลาสจาก Precision และ Recall ที่ได้

3) คำนวณค่า  $mAP$  โดยเอาค่า AP ของแต่ละคลาสมาเฉลี่ย  
คำนวณได้จากสมการดังนี้

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Accuracy คืออัตราส่วนของข้อมูลที่ทำนายถูกต้องต่อจำนวนทั้งหมดในชุดทดสอบ  
ค่า Accuracy สูงแสดงถึงโมเดลทำนายถูกต้องได้มากกับข้อมูลชุดทดสอบทั้งหมด คำนวณได้  
จากสมการ

Accuracy = (จำนวนข้อมูลที่ทำนายถูกต้อง) / (จำนวนข้อมูลทั้งหมดในชุดทดสอบ)

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision คืออัตราส่วนของวัตถุที่โมเดลทำนายว่าเป็นวัตถุและทำนายถูกต้องต่อ  
จำนวนวัตถุที่โมเดลทำนายว่าเป็นวัตถุทั้งหมด ซึ่งค่า Precision สูงแสดงถึงโมเดลมีการทำนาย  
วัตถุเป็นบวก (Positive) ได้แม่นยำมาก คำนวณได้จากสมการ

Precision = (จำนวนวัตถุที่ตรวจจับถูกต้อง) / (จำนวนวัตถุที่โมเดลทำนายว่าเป็นวัตถุ)

$$Precision = TP / (TP + FP)$$

Recall คืออัตราส่วนของวัตถุที่โมเดลทำนายว่าเป็นวัตถุและทำนายถูกต้องต่อวัตถุ  
ที่เป็นบวกจริง (True Positive) ค่า Recall สูงแสดงถึงโมเดลมีความสามารถตรวจจับวัตถุได้ครบ  
ทุกวัตถุที่เป็นบวกจริงในภาพ คำนวณได้จากสมการ

Recall = (จำนวนวัตถุที่ตรวจจับถูกต้อง) / (จำนวนวัตถุที่เป็นบวกจริงทั้งหมด)

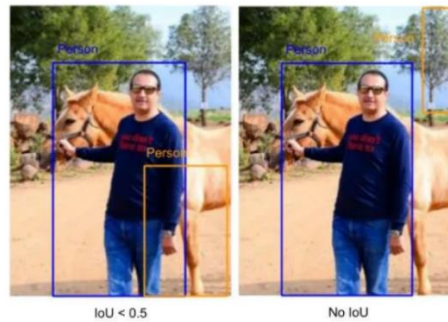
$$Recall = TP / (TP + FN)$$

True Positive (TP) หมายถึง วัตถุที่โมเดลทำนายว่าจริง ค่าคำตอบเท่ากับจริง



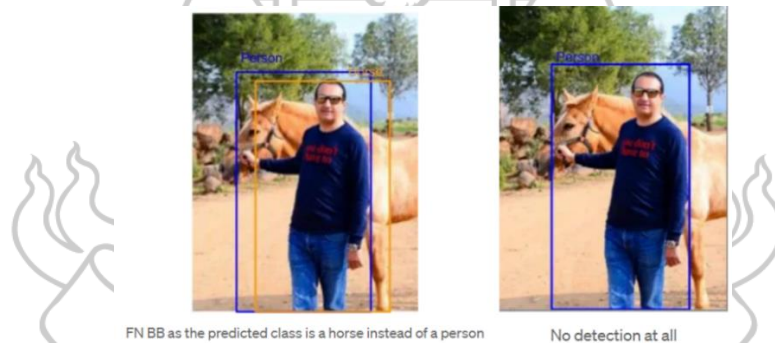
ภาพที่ 2.19 กำหนดเป็น TP เมื่อค่า IoU > 0.5 มีการทำนายคลาสคำตอบถูกต้อง

False Positive (FP) หมายถึง วัตถุที่โมเดลทำนายว่าจริง คำตอบเท่ากับไม่จริง กำหนดเป็น FP เมื่อค่า IoU < 0.5 มีการทำนายคลาสคำตอบไม่ตรงกับผลเฉลย



ภาพที่ 2.20 กำหนดเป็น FP เมื่อทำนายคลาสคำตอบไม่ตรงกับผลเฉลย

False Negative (FN) หมายถึง วัตถุที่โมเดลทำนายว่าไม่จริง คำคำตอบเท่ากับจริง จากภาพที่ 2.21 ภาพด้านซ้ายกำหนดเป็น FN เมื่อค่า IoU > 0.5 แต่มีการทำนายคลาสมผิดพลาด ภาพด้านขวากำหนดเป็น FN เมื่อ IoU = 0 ไม่มีผลการทำนายหรือไม่มีผลการตรวจจับวัตถุ



ภาพที่ 2.21 กำหนดเป็น FN เมื่อมีการทำนายคลาสมผิดพลาดหรือไม่มีผลการทำนาย

True Negative (TN) หมายถึง วัตถุที่โมเดลทำนายว่าไม่จริง คำตอบเท่ากับไม่จริง

## 5. งานวิจัยที่เกี่ยวข้อง

5.1 หุ่นยนต์คู่กายคัดแยกผลจากโดยใช้อัลกอริทึม Yolo และ บรรจุภัณฑ์ในกระบวนการผลิตน้ำตาล (กรณีการ มูลโพธิ์, 2565)

งานวิจัยนี้ได้กล่าวถึงการนำการเรียนรู้เชิงลึก CNN ด้วยอัลกอริทึม YOLOv5 ประยุกต์ใช้งานในการคัดแยกผลจากน้ำดื่มร่วมกันหุ่นยนต์ Kuka มีการสร้าง Dataset สำหรับฝึกสอนโมเดลจำนวน 1,333 ภาพ จากนั้นทำการสร้างคลาสให้กับวัตถุ (Annotation) โดยมีจำนวนคลาสทั้งหมด 3 คลาส ใช้จำนวนรอบในการฝึกสอน 500 รอบ

ระบบหุ่นยนต์ถูกำกัดแยกผลจากโดยใช้อัลกอริทึม Yolo และบรรจุภัณฑ์ในกระบวนการผลิตน้ำดื่ม มีจุดเด่นที่ใช้อัลกอริทึม YOLOv5 ซึ่งมีประสิทธิภาพและความเร็วในการตรวจจับสูงเนื่องจากอัลกอริทึมทำงานแบบ One-stage object detection จากการฝึกสอนโมเดลได้ผลลัพธ์ค่าความแม่นยำสูงถึง 97.23% ข้อสังเกตเมื่อนำโมเดลไปใช้งานจริงพบว่ามีค่าความผิดพลาดสูง ดังภาพที่ 2.22 สาเหตุอาจเกิดจาก

- 1) *Overfitting* การที่โมเดลอาจได้รับการฝึกสอนกับข้อมูลฝึกสอนมากเกินไป ทำให้โมเดลเรียนรู้รายละเอียดของข้อมูลชุดฝึกสอนมากเกินไป ทำให้ทำนายข้อมูลที่ใช้สำหรับฝึกสอนได้ดี แต่เมื่อนำไปใช้กับข้อมูลจริงได้ค่าความถูกต้องต่ำ
- 2) *Class Imbalance* การไม่สมดุลของจำนวนข้อมูลในแต่ละคลาส
- 3) *Distribution Shift* ข้อมูลที่ใช้ในการทดสอบอาจแตกต่างจากชุดข้อมูลที่ใช้ในการฝึกสอน
- 4) *ปัจจัยด้านแสงสว่าง* ทำให้ภาพแตกต่างจากข้อมูลฝึกสอน

ตารางที่ 1 ผลการทดลองการตรวจจับผลจากตามกลุ่มที่ออกแบบ

ครั้งที่	ฉลากสมบูรณ์ (ขวด)	ฉลากไม่สมบูรณ์ (ขวด)	ไม่มีฉลาก (ขวด)	ตรวจคัดแยกผิดพลาด
1	27	18	15	5.00%
2	31	14	15	1.67%
3	29	16	15	1.67%
4	30	15	15	0.00%
5	28	17	15	3.33%

ภาพที่ 2.22 ผลการทดลองการตรวจจับผลจากตามกลุ่มที่ออกแบบ

**5.2 วิจัยฉยมะเร็งผิวหนังจากภาพถ่ายโทรศัพท์ด้วยโครงข่ายประสาทเทียมสังวัตนาการ (ขวัญกมล ดิฐกัญจน์, ปิยะวัฒน์ หนูเล็ก และกรวิทย์ พฤษชัยนimit, 2565)**

งานวิจัยนี้ได้กล่าวถึงการพัฒนาโมเดลในการตรวจหาเมะเร็งผิวหนังด้วย Convolutional Neural Network ด้วยภาพถ่ายจากโทรศัพท์มือถือ โดยการใช้ CNN Efficient NetB7, VGG19, Xception, ResNet 152V2, DensNet121 โดยใช้ชุดข้อมูลการฝึกสอน PAD-UFES-20 จากสถาบัน Federal University of Espirito Santo มีการแบ่งข้อมูลสำหรับฝึกสอน จำนวน 1,600 ภาพ สำหรับทดสอบ 400 ภาพ มีการปรับปรุงภาพ dataset ลดสิ่งรบกวน เช่น เส้นขน ด้วย Digital Hair Removal มีการปรับขนาดของภาพให้เหมาะสมในแต่ละสถาปัตยกรรม มีการใช้เทคนิคการถ่ายโอนความรู้ของโมเดล พบว่าโมเดล DenseNet121 มีค่าความแม่นยำมากที่สุดที่ 81.50% ทั้งนี้ผู้วิจัยให้เหตุผลว่ารูปภาพในการฝึกสอน model มีจำนวนน้อยทำให้ค่าความแม่นยำน้อยตามไปด้วย

### 5.3 การตรวจสอบข้อบกพร่องวงจรพิมพ์ด้วยเครื่องมือทางการประมวลผลภาพ (จักรกฤษ อ่อนชื่นจิตร, 2564)

งานวิจัยนี้ได้กล่าวถึงการหาจุดบกพร่อง พร้อมทั้งระบุตำแหน่ง ขนาด รูปร่าง ลักษณะของจุดบกพร่อง และสามารถจำแนกประเภทจุดบกพร่องที่เกิดขึ้นบนแผงวงจรพิมพ์ PCB โดยใช้การประมวลผลภาพ โดยมีขั้นตอนหลัก 3 ขั้นตอนคือ 1) การประมวลผลภาพเบื้องต้น 2) การตรวจจับจุดบกพร่อง 3) จำแนกจุดบกพร่อง ในขั้นตอนการประมวลผลภาพจะทำการแปลงภาพ RGB เป็นภาพระดับเทา จากนั้นใช้การกรองเพื่อขจัดสัญญาณรบกวน ใช้การแปลง Affine เพื่อให้ภาพ PCB ที่มีจุดบกพร่องอยู่ในรูปแบบเดียวกับต้นแบบ การแบ่งส่วนภาพใช้อัลกอริทึม Fuzzy c-mean เพื่อแยกภาพลายวงจรออกจากภาพพื้นหลัง การตรวจจับจุดบกพร่องใช้วิธีการลบภาพโดยเปรียบเทียบกับ PCB ต้นแบบ การจำแนกประเภทใช้อัลกอริทึม CHT MR และ CCL ร่วมกับการคำนวณทางคณิตศาสตร์ ผลการทดสอบสามารถตรวจจับจุดบกพร่องได้ 100% มีความแม่นยำในการจำแนกชนิดจุดบกพร่องเฉลี่ยประมาณ 99.05%

### 5.4 เทคนิคการคัดเลือกคุณภาพของตัวอ่อนปูแสมด้วยวิธีการตรวจจับการเคลื่อนที่ (เชิด คงห้อย และ นิคม สุวรรณวร, 2563)

งานวิจัยนี้ได้กล่าวถึงการตรวจจับการเคลื่อนไหวของลูกปูแสมในกระบะเพื่อหาค่าเฉลี่ยอัตราการเคลื่อนที่ในการประเมินความแข็งแรงของลูกปู ด้วยเทคนิคการประมวลผลภาพจาก OpenCV ข้อมูลอินพุตจากภาพวิดีโอ มีขั้นตอนดังนี้ 1) บันทึกภาพวิดีโอเพื่อนำเข้าสู่ระบบ 2) ตรวจจับวัตถุที่เคลื่อนไหวด้วยการลบภาพพื้นหลังด้วยเทคนิค (Motion detection by background subtraction) โดยการหาความแตกต่างจากเฟรมที่ติดกัน แปลงวิดีโอในระดับเทาและแปลงเป็น Binary ด้วยการ fix Threshold = 128 3) คัดกรองวัตถุที่เคลื่อนที่ด้วยกระบวนการทางสัญญาณวิทยา



Morphological เพื่อลดสัญญาณรบกวนและเพิ่มประสิทธิภาพโดยการใช้การขยาย Dilation mask = 5x5 และการกัดเซาะ Erosion mask = 3x3 เพื่อกรองสิ่งรบกวนขนาดเล็กออกไป แล้ว Dilation อีกครั้ง 4) วิเคราะห์การเคลื่อนไหวของวัตถุ โดยการตรวจสอบภาพจากเฟรมก่อนหน้า มาเทียบกับเฟรมปัจจุบันเพื่อหาตำแหน่งการเคลื่อนที่โดยใช้ตัวกรอง Kalman Filter ผลลัพธ์ สามารถตรวจจับการเคลื่อนที่ของลูกปู้ได้ จากการทดสอบการเคลื่อนที่ของลูกปู้พบว่าระยะที่ 3 มีอัตราเฉลี่ยการเคลื่อนที่ที่ดีที่สุด คิดเป็นร้อยละ 87.92%

### 5.5 ระบบตรวจจับและคัดแยกป้ายจราจร (ตะวัน ขุนอาสา, 2558)

งานวิจัยนี้ได้กล่าวถึงระบบช่วยเหลือผู้ขับขี่ยานพาหนะเพื่อลดอุบัติเหตุบนท้องถนน ด้วยการตรวจจับป้ายจราจรประเภทป้ายบังคับ นำมาแสดงผลแบบเรียลไทม์ ด้วยการค้นหาภาพที่มีลักษณะคล้ายกับป้ายจราจรที่กำหนดไว้ ด้วยหลักการค้นหาวัตถุภายในภาพ Haar Like Feature detection เมื่อทำการตรวจพบแล้วจะทำการ capture เฉพาะภาพป้ายนั้นออกมาเพื่อนำไปคัดแยกประเภทของป้ายด้วยหลักการจับคู่กับแม่แบบแล้วทำการแสดงผล โดยมีการนำข้อมูลป้ายจำนวน 800 ป้ายเข้าฝึกสอนด้วยฟังก์ชัน haartraining ในไลบรารี ของ OpenCV เพื่อให้ได้ฐานข้อมูลภาพ ผลการทดสอบพบว่าอัตราการตรวจจับป้ายได้ถูกต้องอยู่ในอัตราที่สูงถึงร้อยละ 96.95 ซึ่งเป็นผลการทดสอบจากวิดีโอที่คุณภาพดีไม่สั่นไหวมากนัก ประกอบกับช่วงเวลากลางวันให้แสงสว่างมากภาพมีอัตราความคมชัดมาก และมีอัตราความถูกต้องร้อยละ 80.45 หากภาพมีความมืดหรือสว่างมากเกินไป

### 5.6 การปรับปรุงประสิทธิภาพของกระบวนการบริการลูกค้าโดยใช้การวิเคราะห์จากกล้องวงจรปิด (พุดิพงษ์ จันทร์แจ่ม, 2563)

งานวิจัยนี้ได้กล่าวถึงการตรวจสอบการให้บริการลูกค้าในร้านอาหารของพนักงาน นับจากเวลาที่ลูกค้าเข้าร้านจนถึงวันที่โต๊ะ จากข้อมูลภาพกล้องของกล้องวงจรปิดด้วยการเรียนรู้เชิงลึก (Deep Learning) ด้วยอัลกอริทึม YOLOv4 การสร้าง Dataset จำนวน 2,010 ภาพ โดยมีการทำความสะอาดข้อมูลด้วยการลบข้อมูลซ้ำ รูปที่ไม่มีวัตถุที่ต้องการ แบ่งรูปภาพขนาดใหญ่ (มุมกว้าง) ตามต้นฉบับของวิดีโอและรูปภาพขนาดเล็ก (มุมแคบ) ประเภทละ 1,005 รูป ทำการแบ่งข้อมูลเพื่อฝึกสอนโมเดลในอัตราส่วน ฝึกสอน 80% ทดสอบ 20%

จุดเด่นผลการวัดประสิทธิภาพของโมเดลค่า mean Average Precision (mAP) สูงถึง 96% คาดว่าใช้เทคนิคการ Transfer Learning จากโมเดลโดยใช้ Pre-Trained ของ AlexeyAB เป็นต้นแบบ โดยโมเดลดังกล่าวมีการฝึกสอนไว้เป็นอย่างดีแล้ว ข้อมูลที่เข้าไปฝึกสอนมีความคมชัดและมีข้อมูลที่หลากหลายรูปแบบ

### 5.7 การตรวจจับมัจจุด้วยการเรียนรู้เชิงลึก ฐานข้อมูลงานวิจัย (ราตรี คำโหลง, 2565)

งานวิจัยนี้ได้กล่าวถึงการตรวจสอบมัลติคัลที่สุกพร้อมเก็บเกี่ยวโมเดลในการด้วยการเรียนรู้เชิงลึก (Deep Learning) โดยพิจารณาผลสุกและดิบด้วยสีของผลที่มีสีแดงอมม่วง หรือม่วงเข้ม โดยใช้อัลกอริทึมจำนวน 2 อัลกอริทึมในการเปรียบเทียบคือ CNN, Faster R-CNN ซึ่งเป็นอัลกอริทึมทำงานแบบ Two-stage object detection มีการสร้าง Dataset ในการฝึกสอนแบบจำลองด้วยภาพจำนวน 320 ภาพ นำมาไขว้กัน 10 ชุด (10 fold cross validation) มีการฝึกฝนโมเดล CNN จำนวน 10 รอบ ฝึกสอนโมเดล Faster R-CNN จำนวนรอบ 20,000 รอบ

ผลการทดสอบโมเดล พบว่าในการวัดประสิทธิภาพของโมเดลด้วยคอนฟิวชันเมทริกซ์ ทดสอบแบบจำลองด้วยภาพจำนวน 80 ภาพ มีลูกดิบอย่างเดียว 62 ภาพ สุกอย่างเดียว 17 ภาพ และทั้งสุกทั้งดิบ 17 ภาพ ภาพมะพร้าวจำนวน 20 ภาพ ผลการทดสอบ พบว่าอัลกอริทึม Faster R-CNN จำแนกได้ถูกต้อง 90.22% CNN จำแนกได้ 50.62% ทั้งนี้อัตราความถูกต้องยังไม่สูงมากนักอาจมาจากข้อมูลฝึกสอนน้อย แต่ใช้จำนวนรอบในการฝึกฝนแบบจำลองมาก โมเดลมีโอกาสเกิดการ Overfitting คือการทำนายข้อมูลในชุดฝึกได้แม่นยำ แต่นำไปใช้งานจริงข้อมูลใหม่ ๆ มีผลการทำนายไม่ดีเท่าที่ควร มีจำนวนรอบในการฝึกฝนโมเดลทั้งสองอัลกอริทึมแตกต่างกันมาก ทำให้ข้อมูลในการเปรียบเทียบประสิทธิภาพไม่สะท้อนความเป็นจริงของผลลัพธ์การวัดประสิทธิภาพโมเดล ดังภาพที่ 2.23

	CNN		Faster R-CNN	
	ค่า	ร้อยละ	ค่า	ร้อยละ
ความถูกต้อง	0.5062	50.62	0.9020	90.2
ความเที่ยง	0.2642	26.42	0.6316	63.16
ความครบถ้วน	0.9333	93.33	0.4898	48.98
เอฟวันสกอร์	0.4118	41.18	0.5517	55.17

ภาพที่ 2.23 เปรียบเทียบประสิทธิภาพของโมเดล CNN, Faster R-CNN

**5.8 Fast Classifying Non-Helmeted Motorcyclists by Using Convolutional Neural Networks ( Kietikul Jearanaitanakij, Karnnumart lamthammarak, Nattakitt Wangcharoen, 2021)**

งานวิจัยนี้ได้กล่าวถึงการจำแนกผู้ไม่สวมหมวกนิรภัยด้วย Convolutional Neural Network โดยใช้อัลกอริทึม Resnet-152-v2, Inception-Resnet-v2, Inception-v3 ใช้ข้อมูลสำหรับฝึกสอนโมเดลจำนวน 3,450 ภาพ สำหรับทดสอบ 1,150 ภาพ มีการแบ่งชุดข้อมูลออกเป็น 5 folds

cross validation มีค่าความแม่นยำอยู่ที่ 93.79%, 95.21%, 95.67% ตามลำดับ พบว่าโมเดลที่ได้จากอัลกอริทึม Inception-Resnet-v2 และ Inception-v3 มีความแม่นยำดีที่สุดจึงได้ใช้โมเดลจาก Inception-Resnet-v2 สำหรับจำแนกรถจักรยานยนต์กับวัตถุอื่น และใช้โมเดลจาก Inception-v3 สำหรับจำแนกผู้สวมใส่หมวกนิรภัยกับไม่สวมใส่ จากผลการทดสอบโมเดลจาก Inception-v3 ใช้ Proposed (Filter) ตรวจสอบการสวมหมวกนิรภัยพบว่าค่า Test Accuracy อยู่ที่ 95.65%

#### **5.9 Deep Learning and Image Processing for Disc Brake Pad Identification: A Case Study of Brake Pads Company (Nicha Khathinhorm, Rungrod Samankitesakul, Wasakorn Laesanklang, Banpot Horbanluekit, Somkid Amornsamankul, 2022)**

งานวิจัยนี้ได้กล่าวถึงการตรวจสอบผ้าเบรกรถยนต์ในแต่ละชนิดเพื่อส่งมอบให้ลูกค้าได้อย่างถูกต้อง โดยใช้กระบวนการ CNN YOLOv5 และ Faster R-CNN เพื่อเปรียบเทียบประสิทธิภาพการทำงานของทั้งสองอัลกอริทึม มีการสร้าง Dataset สำหรับฝึกสอนโมเดลจำนวน 1,710 ภาพ จากนั้นทำการสร้างคลาสให้กับวัตถุ (Annotation) แบ่งข้อมูลสำหรับ Train 80% = 1,368 รูป สำหรับ Validate 20% = 342 รูป มีจำนวนคลาสทั้งหมด 4 คลาส โดยโมเดลสำหรับการตรวจจับและแยกชนิดของผ้าเบรกทั้ง 2 อัลกอริทึมสามารถตรวจจับและแยกชนิดของผ้าเบรกได้ แต่อัลกอริทึม YOLOv5 มีประสิทธิภาพที่ดีกว่า ทั้งด้าน Precision, detection speed และ loss function ถึงแม้ว่า R-CNN จะใช้เวลาในการฝึกสอนแบบจำลองที่น้อยกว่า

ระบบ Deep Learning and Image Processing for Disc Brake Pad Identification มีจุดเด่นที่อัลกอริทึม YOLOv5 ให้ผลลัพธ์ค่าเฉลี่ยรวมทุกคลาสของค่า Average Precision (AP) สูงถึง 99.39% บ่งชี้ว่าโมเดลมีประสิทธิภาพในการตรวจจับวัตถุได้สูง และสามารถทำนายคลาสหรือวัตถุที่แตกต่างกันในภาพได้ดี คาดว่ารูปแบบคุณลักษณะของผ้าเบรกในแต่ละชนิดมีลักษณะแตกต่างกันชัดเจนโมเดลจึงสามารถจำแนกประเภทได้ง่าย

## บทที่ 3

### วิธีดำเนินการศึกษา

การดำเนินการศึกษาค้นคว้าครั้งนี้ เป็นการศึกษาเพื่อหาวิธีการตรวจจับรอยขีดข่วนบนผิวอลูมิเนียม ที่เป็นวัตถุดิบในการผลิตของอุตสาหกรรมผลิตกล่องบรรจุภัณฑ์ ด้วยวิธีการเรียนรู้เชิงลึก โดยมีการใช้โครงข่ายประสาทเทียม แบบคอนโวลูชัน มีเนื้อหาประกอบไปด้วย

1. วิเคราะห์ปัญหาและศึกษาเทคโนโลยีที่นำมาใช้
2. รูปแบบและวิธีการศึกษา
3. ขั้นตอนการเก็บรวบรวมข้อมูล และการสร้างชุดข้อมูลการฝึกสอน
4. ขั้นตอนการสร้างโมเดล การตรวจจับวัตถุ
5. การนำโมเดลไปใช้งาน
6. วิธีการประเมินประสิทธิภาพ

#### 1. วิเคราะห์ปัญหาและศึกษาเทคโนโลยีที่นำมาใช้

จากการศึกษาสภาพปัญหาของชิ้นงานอลูมิเนียมที่เป็นวัตถุดิบในการผลิต พบว่าผิวหน้าของแผ่นอลูมิเนียมเมื่อถูกสัมผัสด้วยของแข็งเกิดรอยขีดข่วนได้ง่าย ซึ่งลักษณะรอยขีดข่วน มีทั้งรอยขนาดใหญ่ รอยเส้นขนาดเล็ก รอยนูน รอยหนูน จึงได้ศึกษาเทคโนโลยีการตรวจจับรอยขีดข่วน และศึกษางานวิจัยที่เกี่ยวข้อง พบว่ามีวิธีการตรวจสอบรอยขีดข่วนอยู่ 2 วิธีหลักๆ คือ วิธีการประมวลผลภาพ (Image Processing) และการใช้โมเดลการเรียนรู้เชิงลึก Convolution Neural Network (CNN) มาทำการแก้ไขปัญหา ผู้ศึกษาจึงได้สร้างระบบการตรวจสอบรอยขีดข่วนด้วยวิธีการประมวลผลภาพ เพื่อศึกษาความเป็นไปได้ในการนำวิธีการดังกล่าวมาแก้ไขปัญหา จากทดสอบใช้ Library OpenCV กับภาษา Python พบปัญหาวิเคราะห์ภาพรอยขีดข่วนให้ผลลัพธ์ที่ได้ไม่ดี ภาพที่ทำการวิเคราะห์มีแสงเงาแตกต่างกันทำให้ผลลัพธ์ในการวิเคราะห์รอยขีดข่วนผิดพลาดสูง และติกรอบ (Bounding Box) ผิดพลาดไม่ตรงกับตำแหน่งที่เกิดรอยขีดข่วน ผู้ศึกษาจึงได้ศึกษาและทดสอบความเป็นไปได้ในการนำโมเดลการเรียนรู้เชิงลึก Convolution Neural Network (CNN) มาใช้งาน เบื้องต้นพบว่าทำงานได้ดีในระดับที่น่าพอใจ ผู้ศึกษาจึงได้ศึกษาอัลกอริทึมในปัจจุบันที่ใช้ในงาน Object detection ซึ่งในปัจจุบันพบอัลกอริทึมที่นิยมใช้อยู่โดยแบ่งออกเป็น 2 ประเภท คือ

### 1.1 One-stage object detection

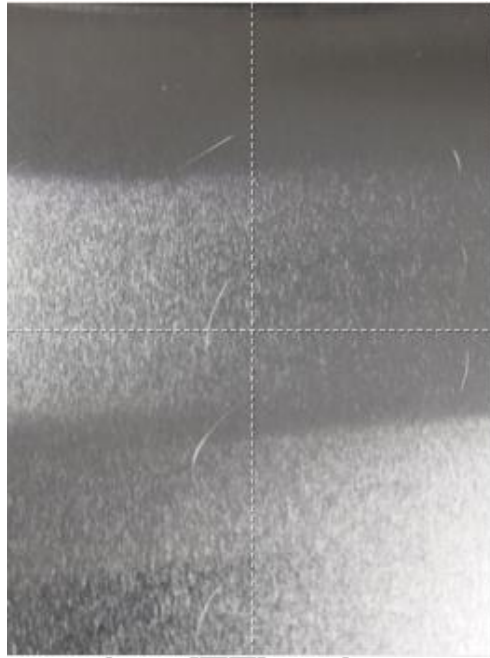
เป็นเทคนิคในการตรวจจับวัตถุ (Object Detection) ในการส่งอินพุตภาพเข้าไปครั้งเดียวโมเดลจะทำนายผลลัพธ์มาออกมาเลย อัลกอริทึมที่ใช้เทคนิคนี้ คือ YOLO (You Only Look Once) และ SDD (Single Shot Multi-Box Detector) เทคนิคมีความเร็วในการทำงานสูงและในเวอร์ชันปัจจุบันมีความแม่นยำสูงขึ้นมา

### 1.2 Two-stage object detection

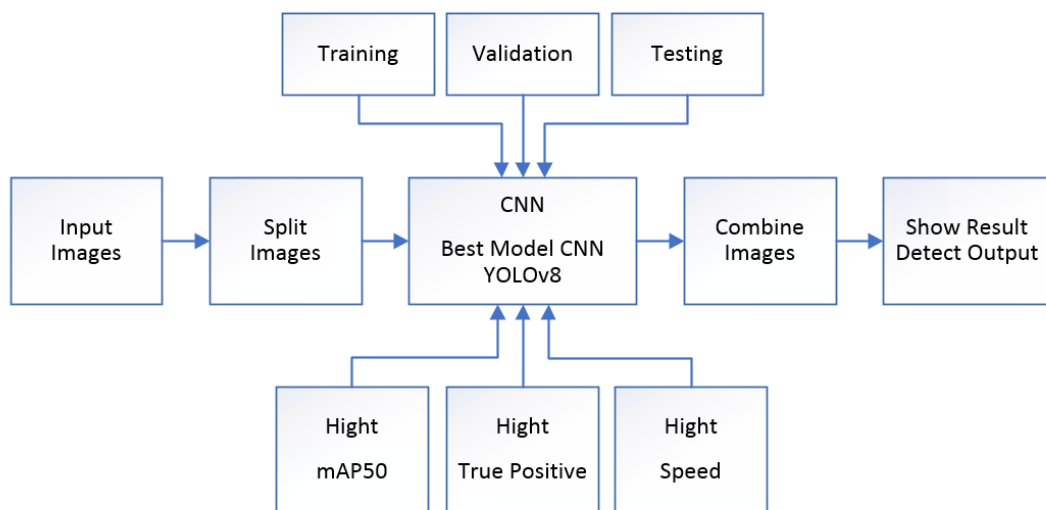
เป็นเทคนิคในการตรวจจับวัตถุ (Object Detection) มีขั้นตอนในการทำงานสองขั้นตอนหลักในการประมวลผลภาพ คือก่อนการทำนายจะสร้าง Region Proposals (Region Proposal Network - RPN) ขึ้นมาก่อนเพื่อเลือกพื้นที่ที่เป็นความเป็นไปได้ของวัตถุ อัลกอริทึมที่ใช้เทคนิคนี้ เช่น R-CNN, Fast R-CNN และ Faster R-CNN เทคนิคนี้ให้ความแม่นยำสูงแต่มีปัญหาด้านความเร็ว

ผู้ศึกษาจึงได้ทำการเลือกใช้อัลกอริทึม YOLO (You Only Look Once) ในเวอร์ชัน 8 ซึ่งเป็นเวอร์ชันล่าสุดเปิดตัวเมื่อปี 2023 ซึ่งเป็นเวอร์ชันที่มีความเร็วและมีความแม่นยำสูงมากเมื่อเทียบเวอร์ชันก่อน

ผู้ศึกษาได้ทำการศึกษาการใช้เทคนิคการแบ่งส่วนภาพอินพุตที่มีความละเอียดสูงจากการศึกษาการทำงานของการทำงานของตรวจจับวัตถุแบบ CNN พบว่า ภาพอินพุตที่เข้าสู่โมเดล CNN จะมีการลดขนาดลงให้เหลือความละเอียดตามที่โมเดลต้องการ YOLOv8 ต้องการความละเอียดของอินพุตภาพขนาด 640x640 pixel เมื่อนำภาพอินพุตที่มีความละเอียดสูงกว่า 640 pixel โมเดลจะทำการลดขนาดลงอัตโนมัติ ผู้ศึกษาจึงใช้เทคนิคการแบ่งอินพุตภาพความละเอียดสูงเป็น 4 ส่วน ก่อนเข้าโมเดล ทำให้ยังไม่ถูกลดคุณสมบัติที่สำคัญของวัตถุลงจากการตรวจจับรอยขีดข่วนบางรอยตรวจไม่พบ โดยทำการแบ่งอินพุตภาพเป็นส่วนด้วยการแบ่งครึ่งด้านแกน X และ แกน Y การแบ่งอินพุตควรแบ่งในอัตราส่วนที่ขนาดอินพุตของโมเดลหารลงตัวมากที่สุด ดังแสดงการแบ่งในภาพที่ 3.1



ภาพที่ 3.1 การแบ่งภาพอินพุตภาพที่มีความละเอียดสูงขนาด 3000x4000 pixel แบบ 4 ส่วน โดยมีแผนผังการทำงานของระบบตรวจจับรอยขีดข่วนโดยใช้เทคนิคการแบ่งส่วนภาพดังในภาพที่ 3.2



ภาพที่ 3.2 แผนผังการทำงานของระบบตรวจจับรอยขีดข่วนโดยใช้เทคนิคการแบ่งส่วนภาพ

## 2. รูปแบบและวิธีการศึกษา

ผู้ศึกษาได้มีขั้นตอนการพัฒนาดังนี้

**2.1 พัฒนาแบบจำลองด้วยอัลกอริทึม Convolutional Neural Network (CNN) หรือ โครงข่ายประสาทแบบคอนโวลูชัน โดยใช้โมเดล YOLOv8 เป็นแม่แบบ**

**2.2 เตรียมข้อมูลที่นำเข้าสำหรับการฝึกสอนโมเดล (Model Training) สร้างพื้นที่คุมแสงเป็นกล่องขนาด 1 เมตรคูณ 1 เมตร ติดตั้งหลอดไฟ LED เพื่อให้แสงที่เหมาะสม ถ่ายภาพด้วยกล้องโทรศัพท์มือถือ โดยมีขนาดของภาพกว้างคูณสูง 1477 x 1108 pixels ชนิดของภาพ JPG โดยถ่ายรอยขีดขีดบนพื้นผิวอลูมิเนียมที่มีลักษณะที่แตกต่างจากวัสดุที่ทำให้เกิดรอยและมีความสว่างของภาพที่แตกต่างกัน ทำการครอบภาพในส่วนของภาพที่เกิดรอยขนาด 640x640 pixel สำหรับฝึกสอนโมเดล**

**2.3 การฝึกสอนโมเดล** โดยฝึกสอนโมเดลทั้งหมด 3 แบบ คือ YOLOv8n , YOLOv8s, YOLOv8x ซึ่ง Model ทั้ง 3 ชนิดจะแตกต่างกันที่ความซับซ้อนของสถาปัตยกรรมที่แตกต่างกันส่งผลต่อความแม่นยำและความเร็วในการทำงาน

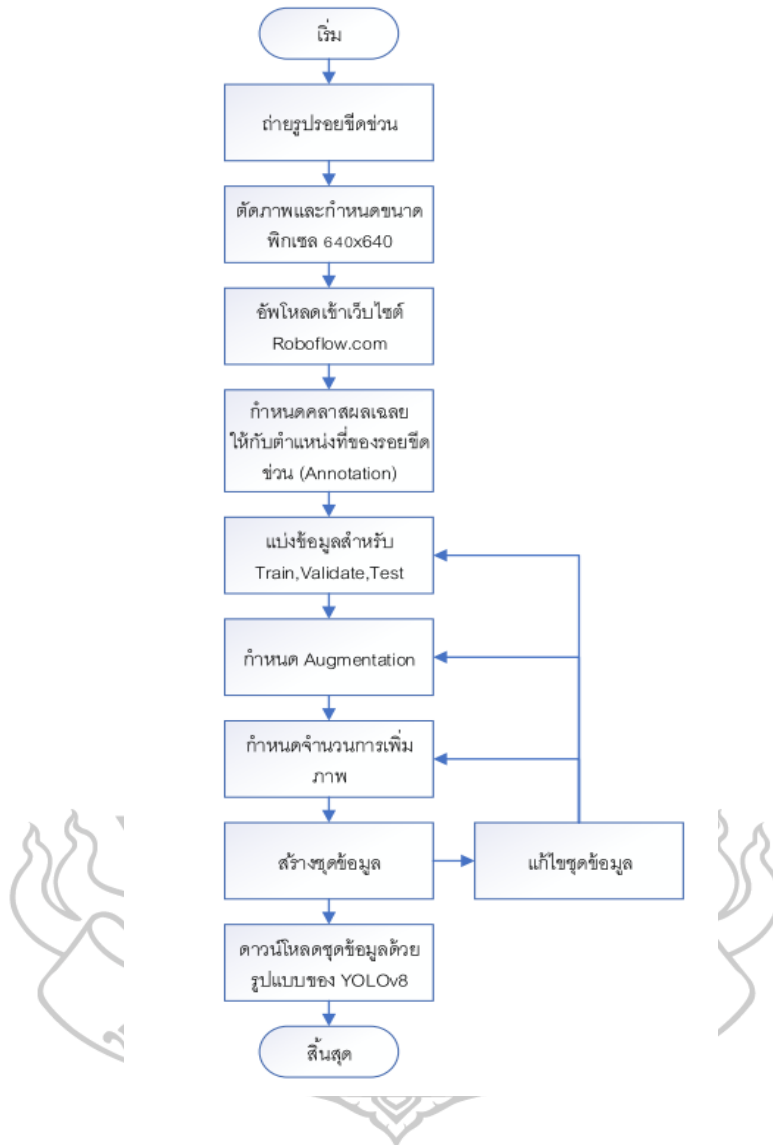
**2.4 การวัดประสิทธิภาพโมเดล** ในการวัดประสิทธิภาพของโมเดลจะวัดจากค่า Mean Average Precision (mAP) และตารางเปรียบเทียบประสิทธิภาพผลลัพธ์ในการตรวจจ็บรอย ขีดข่วนและเวลาที่ใช้ประมวลผลลัพธ์

**2.5 สร้างส่วนประสานผู้ใช้** พัฒนาส่วนประสานผู้ใช้ด้วยภาษา Python และ QT Designer



### 3. ขั้นตอนการเก็บรวบรวมข้อมูล และการสร้างชุดข้อมูลการฝึกสอน

มีขั้นตอนการดำเนินการตามแผนผังในภาพที่ 3.3



ภาพที่ 3.3 แผนผังการสร้างชุดข้อมูลการฝึกสอน





#### 3.1 ขั้นตอนการเก็บรวบรวมข้อมูล

ทำการเก็บข้อมูลภาพถ่ายของรอยขีดข่วน และทำการถ่ายภาพเพิ่มเติมเนื่องจากข้อมูลไม่เพียงพอในการฝึกสอนโมเดล ด้วยโทรศัพท์สมาร์ทโฟน ขนาดของภาพ 1,477 x 1,108 pixels โดยการสร้างพื้นที่คุมแสงสำหรับถ่ายภาพ และถ่ายภาพในพื้นที่คุมแสง และไม่คุมแสง


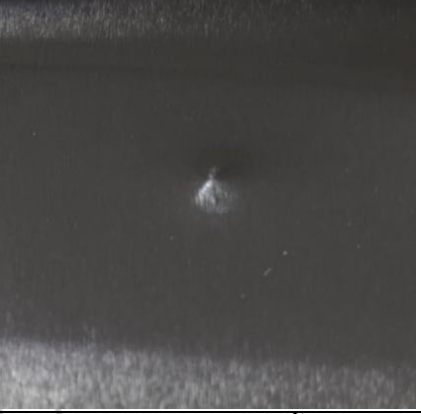






เพื่อให้ได้ภาพที่มีความหลากหลายของแสงตกกระทบบนผิวอลูมิเนียม โดยลักษณะของรอยขีดข่วนที่นำเข้าไปฝึกสอนโมเดล มีลักษณะแสดงในตารางที่ 3.1

ตารางที่ 3.1 แสดงลักษณะรอยขีดข่วน

	
<p>รอยขีดข่วนมีลักษณะเป็นเส้น ความยาวตั้งแต่ 2 มิลลิเมตรขึ้นไป</p>	<p>รอยขีดข่วนมีลักษณะเป็นเส้นหยัก ความยาวตั้งแต่ 2 มิลลิเมตรขึ้นไป</p>
	
<p>รอยขีดข่วนมีลักษณะเป็นจุดแบบกลุ่ม</p>	<p>รอยขีดข่วนมีลักษณะเป็นเส้นหนาหรือเป็นกลุ่ม</p>

## ตารางที่ 3.1 (ต่อ)

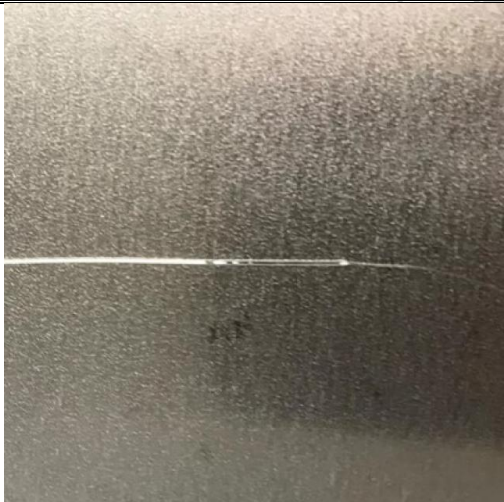
	
รอยบวม ขนาด 1 มิลลิเมตรขึ้นไป	รอยหนูน ขนาด 1 มิลลิเมตรขึ้นไป
	
รอยขีดข่วนเป็นทางยาว	รอยขีดข่วนเส้นเป็นกลุ่ม
	
รอยขีดข่วนมีลักษณะเป็นเส้นบางเป็นทางยาว	รอยขีดข่วนมีลักษณะเป็นเส้นบาง

เมื่อรวบรวมรูปภาพจากโทรศัพท์มือถือครบตามจำนวนที่ต้องการ จากนั้นทำการครอบภาพให้มีขนาด 640x640 Pixels เพื่อนำเข้าสู่ขั้นตอนต่อไป

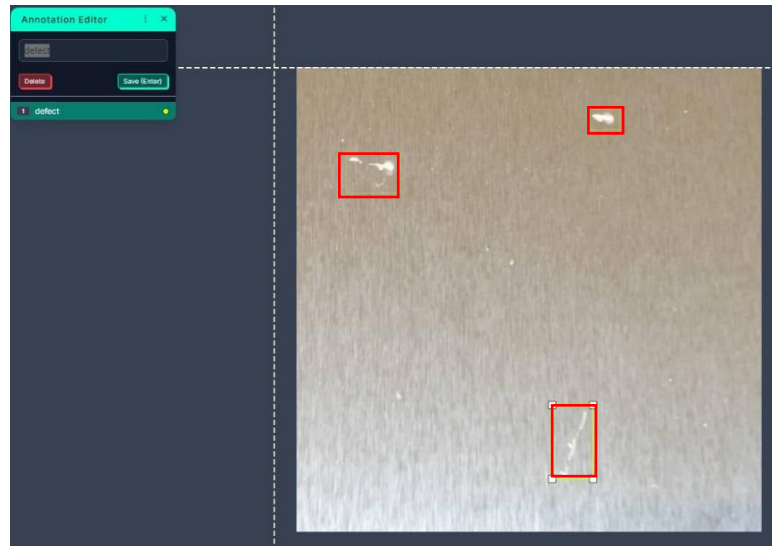
### 3.2 กำหนดประเภทของวัตถุ (Image Annotation)

ในขั้นตอนนี้จะทำการตีกรอบ (Bounding Box) หรือการสร้างคำตอบ (Label) ซึ่งจะระบุขอบเขตของวัตถุที่สนใจภายในภาพ เพื่อกำหนดคุณลักษณะที่สำคัญของวัตถุ และกำหนดชื่อของคลาสของวัตถุนั้น เพื่อใช้เป็นข้อมูลคำตอบในการฝึกฝนโมเดล การทำ Annotation มีความสำคัญอย่างมากในการฝึกสอนและพัฒนาโมเดล เนื่องจากข้อมูลที่ถูกรับ Annotation แล้วจะเป็นแหล่งข้อมูลที่ใช้สอนให้โมเดลเรียนรู้และเข้าใจคุณลักษณะที่สำคัญของวัตถุในภาพ การระบุขอบเขตที่สนใจภายในภาพจะประกอบไปด้วย ชื่อของวัตถุ พิกัดแกน x และพิกัดแกน y ของตำแหน่งที่ได้ทำการตีกรอบ ข้อมูลที่ได้จะมี 2 ส่วนคือ ภาพ และ บ้ายกำกับ TXT ดังตารางที่ 3.2

ตารางที่ 3.2 แสดงผลลัพธ์จากกระบวนการสร้างชุดข้อมูลผลเฉลยของวัตถุ

	พิกัดแกน x	พิกัดแกน y
	มุมซ้ายบน	มุมซ้ายบน
	0	0.5
	0.54375	1
	0.09140625	
คลาส	พิกัดแกน x	พิกัดแกน y
	มุมขวาล่าง	มุมขวาล่าง
s13_jpg.rf.4c4e2fb3dfb3e5b3f1206503318	s13_jpg.rf.4c4e2fb3dfb3e5b3f120650331	
810e2.jpg	8810e2.txt	

ขั้นตอนการตีกรอบผลเฉลยของวัตถุ ทำที่ Website Roboflow.com โดยทำการตีกรอบผลเฉลยกับทุกวัตถุภายในภาพ และทำจนครบทุกภาพ ซึ่งเป็นกระบวนการที่ใช้เวลานานในการทำ แสดงดังภาพที่ 3.4



ภาพที่ 3.4 แสดงการตีกรอบวัตถุและกำหนดประเภทของวัตถุ

### 3.3 การเพิ่มภาพ Augmentations

เพิ่มภาพด้วยกระบวนการ Augmentation เพื่อเพิ่มรูปภาพหรือปรับเปลี่ยนรูปแบบของภาพและให้ภาพมีความหลากหลายรูปแบบมากขึ้นส่งผลให้การเรียนรู้ของโมเดลมีประสิทธิภาพการทำนายได้ดีขึ้น ลดปัญหาการเกิด Overfitting และประหยัดเวลาในการเก็บข้อมูลโดยใช้ภาพต้นฉบับที่ผ่านการสร้างคำตอบของวัตถุเรียบร้อยแล้วจำนวน 1,235 ภาพ ใช้ฟังก์ชัน Augmentation บน Website roboflow.com ทำการเพิ่มภาพแล้วจะได้ภาพในการฝึกสอน 2,600 ภาพ เลือกฟังก์ชันการทำ Augmentation ดังนี้

3.3.1 *Flip* การพลิกภาพ ช่วยลดการเกิด overfitting

3.3.2 *Rotate 90 degree* การหมุนในมุม 90 องศาช่วยให้โมเดลเรียนรู้และจดจำวัตถุในมุม 90 องศา และ 225 องศา ได้ดีขึ้น

3.3.3 *Crop* ตัดขอบของภาพออก เพื่อลดขนาดของภาพหรือโฟกัสในพื้นที่ที่สนใจช่วยให้โมเดลเรียนรู้จากพื้นที่สำคัญของภาพได้ดีขึ้น

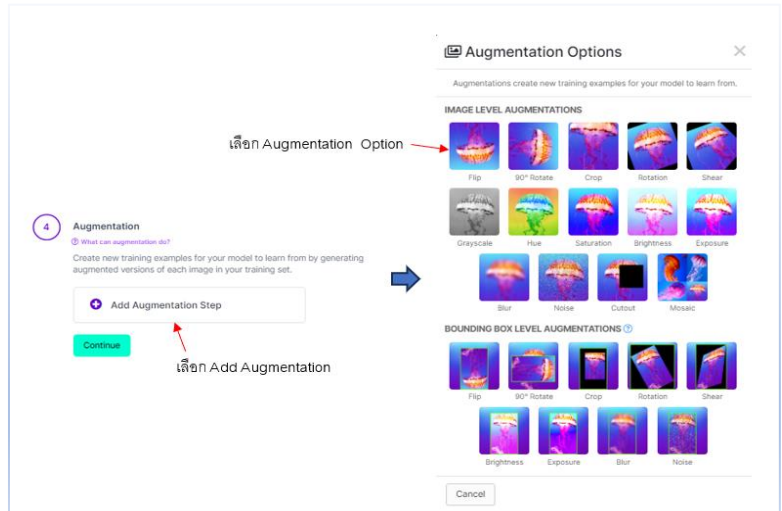
3.3.4 *Rotation* การหมุนภาพออกมาใหม่ในมุมที่แตกต่างจากภาพเดิม เช่น หมุนภาพไปทางซ้าย หรือทางขวา ช่วยให้โมเดลเรียนรู้และจดจำวัตถุในมุมต่าง ๆ ได้ดีขึ้น

3.3.5 *Blur* ลดความคมชัดของภาพ ช่วยลดการเกิด overfitting

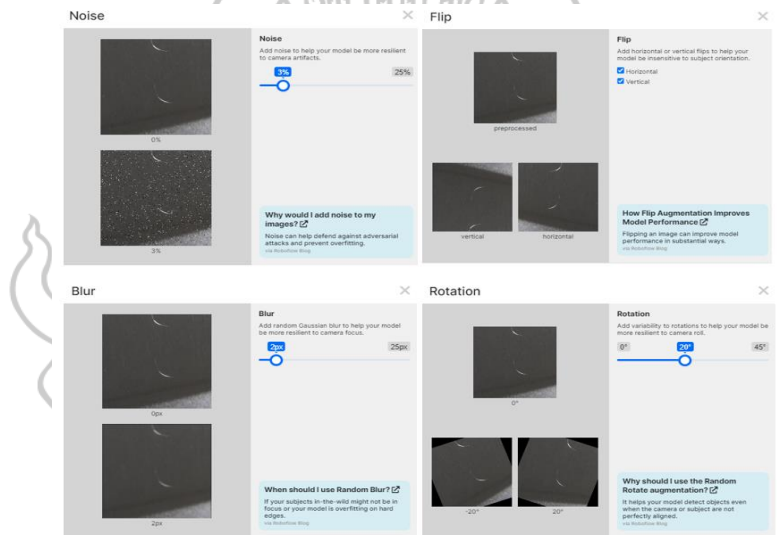
3.3.6 *Noise* เพิ่มจุดสัญญาณรบกวนภาพ ช่วยลดการเกิด overfitting

3.3.7 *Cutout* ตัดหรือลบพื้นที่บางส่วนออกจากภาพให้คงเหลือไว้พื้นหลังสีดำ ช่วยลดการเกิด overfitting เพื่อให้โมเดลเรียนรู้ภาพที่ถูกลบหรือตัดบางส่วนออกได้ดีขึ้น

แสดงกระบวนการทำ Augmentation ในภาพที่ 3.5 และแสดงการปรับค่าพารามิเตอร์กระบวนการ Augmentation แสดงในภาพที่ 3.6



ภาพที่ 3.5 กระบวนการทำ Augmentation



ภาพที่ 3.6 แสดงการปรับค่าพารามิเตอร์กระบวนการ Augmentation

### 3.4 การแบ่งข้อมูล Data Split

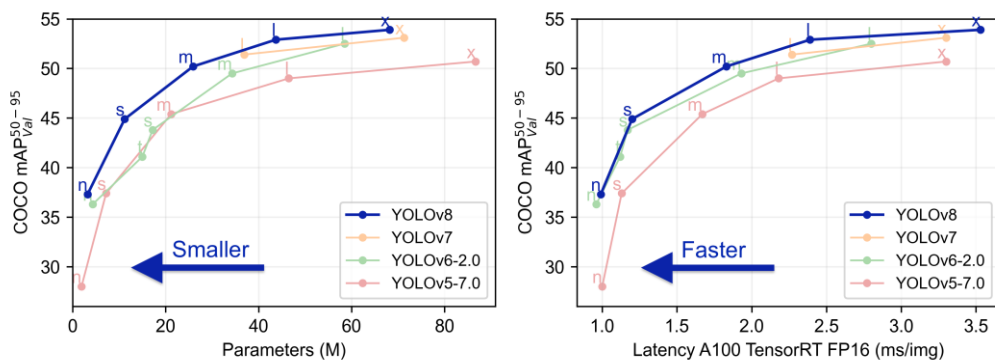
ในการดำเนินศึกษานี้ได้ทำการแบ่งข้อมูลสำหรับฝึกสอน (Data Split) โดยแบ่งตามอัตราส่วน 70:20:10 จากข้อมูลจริงจำนวน 1,235 ภาพ การกำหนดอัตราส่วนข้อมูลการฝึกสอน ที่ 70% เนื่องจากข้อมูลมีจำนวนที่ไม่มาก จึงเน้นข้อมูลส่วนใหญ่สำหรับฝึกสอนเพื่อให้โมเดลเรียนรู้กับข้อมูลที่หลากหลายและมีปริมาณที่มากพอ เพื่อให้ได้โมเดลที่มีประสิทธิภาพดี

เมื่อผ่านกระบวนการเพิ่มภาพจะได้ข้อมูล สำหรับฝึกสอน (Training) จำนวน 2,600 ภาพ สำหรับตรวจสอบ (Validation) จำนวน 246 ภาพ และทดสอบ (Testing)จำนวน 122 ภาพ กระบวนการเพิ่มภาพจะเพิ่มชุดข้อมูลการฝึกสอนเท่านั้น

## 4. ขั้นตอนการสร้างโมเดล การตรวจจับวัตถุ

### 4.1 ฝึกสอนโมเดล (Model Training)

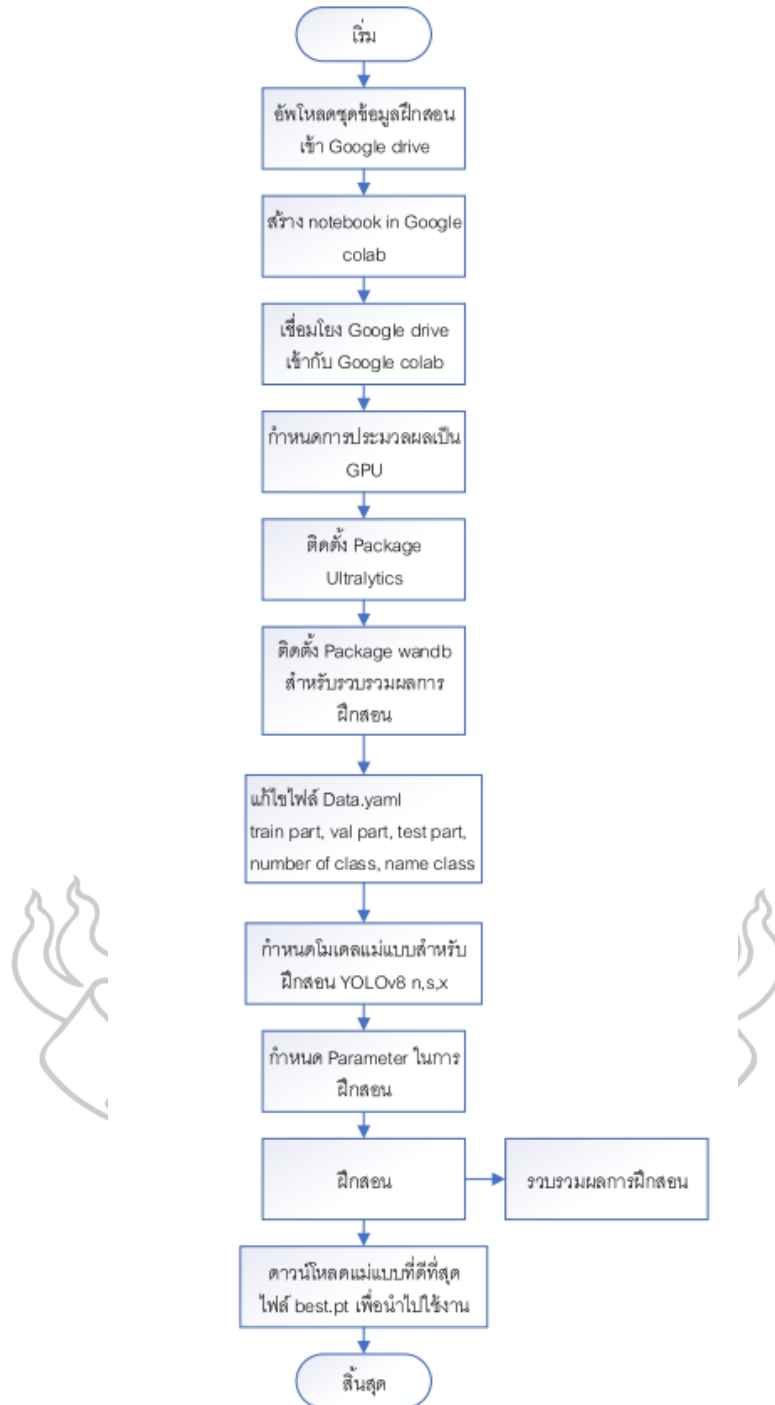
ในการศึกษาครั้งนี้ผู้ศึกษาได้ใช้ฝึกสอนโมเดลการตรวจจับวัตถุ YOLOv8s (You Only Look Once version8) และฝึกสอนโมเดล YOLOv8n, YOLOv8x เป็นตัวเปรียบเทียบในการประเมินประสิทธิภาพเพื่อให้ได้โมเดลที่เหมาะสมในการนำไปใช้งาน YOLOv8 ได้พัฒนาโดย Ultralytics มีพื้นฐานการพัฒนามาจาก YOLOv5 โดยมีการพัฒนาปรับปรุงประสิทธิภาพในด้านความเร็วและความแม่นยำ แสดงการเปรียบเทียบประสิทธิภาพของ YOLO แต่ละเวอร์ชันแสดงดังภาพที่ 3.7



ภาพที่ 3.7 แสดงการเปรียบเทียบอัลกอริทึม YOLO ในแต่ละเวอร์ชัน

ที่มา: learnopencv.com (2023) YOLOv8 : Comprehensive Guide to State Of The Art Object Detection.

มีกระบวนการขั้นตอนการฝึกสอนโมเดลแสดงในแผนผังภาพที่ 3.8



ภาพที่ 3.8 แผนผังขั้นตอนการฝึกสอนโมเดล

การฝึกฝนโมเดลได้ฝึกฝนบน Google Colab เนื่องจากมีทรัพยากรในการฝึกฝนที่มีประสิทธิภาพสูง โดยผู้ศึกษาได้ใช้ Google Colab version free ที่มี Graphic card T4 Memory 16 GB ซึ่งไม่มีค่าใช้จ่าย สำหรับการฝึกฝนโมเดลจะกำหนดพารามิเตอร์กับชุดข้อมูล Dataset ดังต่อตารางที่ 3.3

ตารางที่ 3.3 แสดงการกำหนดค่าพารามิเตอร์สำหรับฝึกสอนโมเดล

Parameter	Configure	Description
number of class	1	จำนวนคุณลักษณะสำคัญวัตถุหรือจำนวนชื่อของวัตถุที่สนใจ
epochs	300	จำนวนรอบในการฝึกสอนของข้อมูลทั้งหมด
imgsz	640	ขนาดของภาพหน่วยเป็น pixels
batch	16	จำนวนของภาพในการนำเข้าฝึกฝนโมเดลในหนึ่งชุด
patience	50	การหยุดสอนหากไม่มีการปรับปรุงที่ดีขึ้น early stopping of training
optimizer	auto	อัลกอริทึมในการปรับปรุงการเรียนรู้ของโมเดล [SGD, Adam, Adamax, AdamW]
lr0	0.001	อัตราการเรียนรู้เริ่มต้น
lrf	0.001	อัตราการเรียนรู้สิ้นสุดท้าย

ปรับตั้งค่าไฟล์ data.yaml โดยไฟล์นี้ใช้เพื่อกำหนดการตั้งค่าและข้อมูลที่เกี่ยวข้องกับข้อมูลของชุดข้อมูลที่ใช้ในการฝึกสอน รวมถึงรายละเอียดของคลาส (classes) ที่ต้องการตรวจจับในการศึกษาคั้งนี้กำหนดหนึ่งคลาสคือ defect แสดงดังภาพที่ 3.9

Parameter	Description
<b>train</b>	ที่อยู่ข้อมูลสำหรับฝึกสอน โมเดล
<b>val</b>	ที่อยู่ข้อมูลสำหรับตรวจสอบความถูกต้องของ โมเดล
<b>test</b>	ที่อยู่ข้อมูลสำหรับทดสอบ โมเดล
<b>nc</b>	จำนวนของคลาส
<b>names</b>	ชื่อของคลาส

```

data.yaml X
1 train: /content/AluminiumDect.V18 testset/Aluminium-Defect.v18i.yolov8/train
2 val: /content/AluminiumDect.V18 testset/Aluminium-Defect.v18i.yolov8/valid
3 test: /content/AluminiumDect.V18 testset/Aluminium-Defect.v18i.yolov8/test
4
5 nc: 1
6 names: ['defect']

```

ภาพที่ 3.9 กำหนดพารามิเตอร์ data.yaml สำหรับฝึกสอนโมเดล



```

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
207/500 4.2G 0.9135 0.659 1.067 53 640: 100% ██████████ 185/185 [00:45<00:00, 4.08it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% ██████████ 8/8 [00:02<00:00, 2.97it/s]
all 246 636 0.893 0.881 0.923 0.605

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
208/500 4.18G 0.9005 0.6361 1.061 59 640: 100% ██████████ 185/185 [00:45<00:00, 4.03it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% ██████████ 8/8 [00:02<00:00, 2.95it/s]
all 246 636 0.895 0.869 0.92 0.607

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
209/500 4.2G 0.8966 0.6376 1.06 67 640: 100% ██████████ 185/185 [00:45<00:00, 4.07it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% ██████████ 8/8 [00:02<00:00, 3.02it/s]
all 246 636 0.896 0.866 0.919 0.607

Epoch GPU_mem box_loss cls_loss dfl_loss Instances Size
210/500 4.17G 0.9031 0.639 1.061 52 640: 100% ██████████ 185/185 [00:45<00:00, 4.06it/s]
Class Images Instances Box(P R mAP50 mAP50-95): 100% ██████████ 8/8 [00:07<00:00, 1.11it/s]
all 246 636 0.903 0.871 0.922 0.609

Stopping training early as no improvement observed in last 50 epochs. Best results observed at epoch 160, best model saved as best.pt.
To update EarlyStopping(patience=50) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

210 epochs completed in 2.816 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 22.5MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.139 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11125971 parameters, 0 gradients
Class Images Instances Box(P R mAP50 mAP50-95): 100% ██████████ 8/8 [00:06<00:00, 1.23it/s]
all 246 636 0.909 0.865 0.924 0.61

Speed: 0.8ms preprocess, 4.7ms inference, 0.0ms loss, 1.6ms postprocess per image
Results saved to runs/detect/train2

```

ภาพที่ 3.10 แสดงผลลัพธ์การฝึกฝนโมเดล YOLOv8s

## 5. การนำโมเดลไปใช้งาน

มีขั้นตอนการทำงานดังนี้

### 5.1 ติดตั้ง YOLOv8 และเตรียมความพร้อม

เมื่อได้ฝึกฝนโมเดลเสร็จแล้วทำการโหลดโมเดลที่ได้ฝึกฝนบน Google Colab ที่มีการฝึกฝนเสร็จแล้วชื่อไฟล์ best.pt และทำการติดตั้ง YOLOv8 ที่คอมพิวเตอร์ส่วนบุคคล นำไฟล์โมเดล best.pt ไปวางใน directory ที่ทำการติดตั้ง YOLOv8

### 5.2 ออกแบบการทำงานของระบบ

ทำการออกแบบขั้นตอนการทำงานของระบบตรวจสอบรอยขีดข่วนบนผิวอลูมิเนียมมีขั้นตอนการทำงานของระบบดังนี้

5.2.1 เริ่มต้นทำงานของระบบเมื่อโปรแกรมถูกเปิด

5.2.2 ระบบแสดงคำแนะนำและแจ้งสถานะพร้อมทำงาน

5.2.3 ปรับปรุงพื้นที่เก็บข้อมูลเดิมหากมีข้อมูลเดิมอยู่จะทำการย้ายไปเก็บยัง

พื้นที่ที่กำหนด

5.2.4 ผู้ใช้งานเลือกฟังก์ชันการทำงาน

1) ตรวจสอบชิ้นงาน

(1) สั่งเปิดกล้อง นำภาพวิดีโอแสดงในกรอบด้านซ้าย

- (2) ผู้ใช้กดปุ่มจับภาพแบบปกติ / แบบแบ่งส่วนภาพ
- (3) ระบบจับภาพจากวิดีโอด้วยคำสั่งของ Library OpenCV
- (4) ระบบทำการแบ่งส่วนภาพ หากผู้ใช้เลือกฟังก์ชันตรวจจับแบบแบ่ง

ส่วน

พร้อมทั้งทำการบันทึกภาพ

- (5) ระบบนำภาพเข้าโมเดลการตรวจสอบรอยขีดข่วนพร้อมระบุตำแหน่ง
- (6) ระบบแสดงผลลัพธ์ของการตรวจจับ
- (7) ระบบรวมภาพเข้าด้วยกัน หากผู้ใช้เลือกฟังก์ชันตรวจจับแบบแบ่ง

ส่วน

- (8) ระบบนำภาพผลลัพธ์มาแสดงผลบนหน้าจอ กรอบด้านขวา
- (9) รอผู้ใช้งานเลือกฟังก์ชันการทำงาน หรือสิ้นสุดการทำงานออกจาก

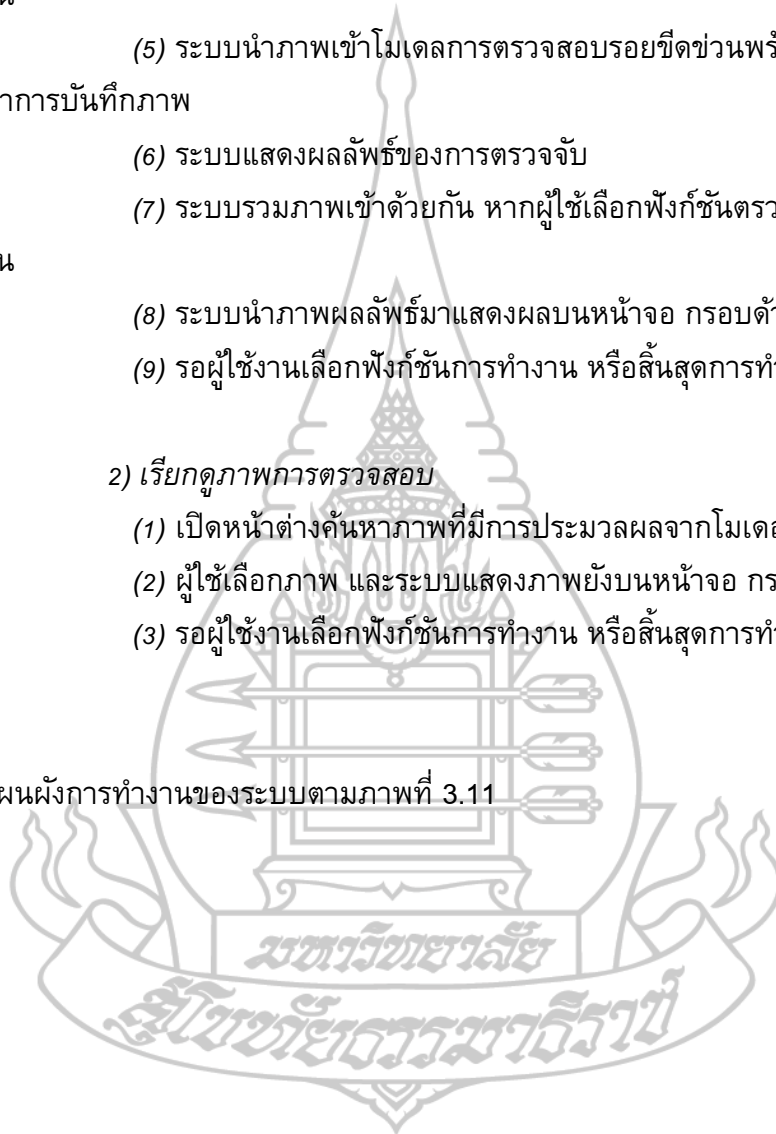
โปรแกรม

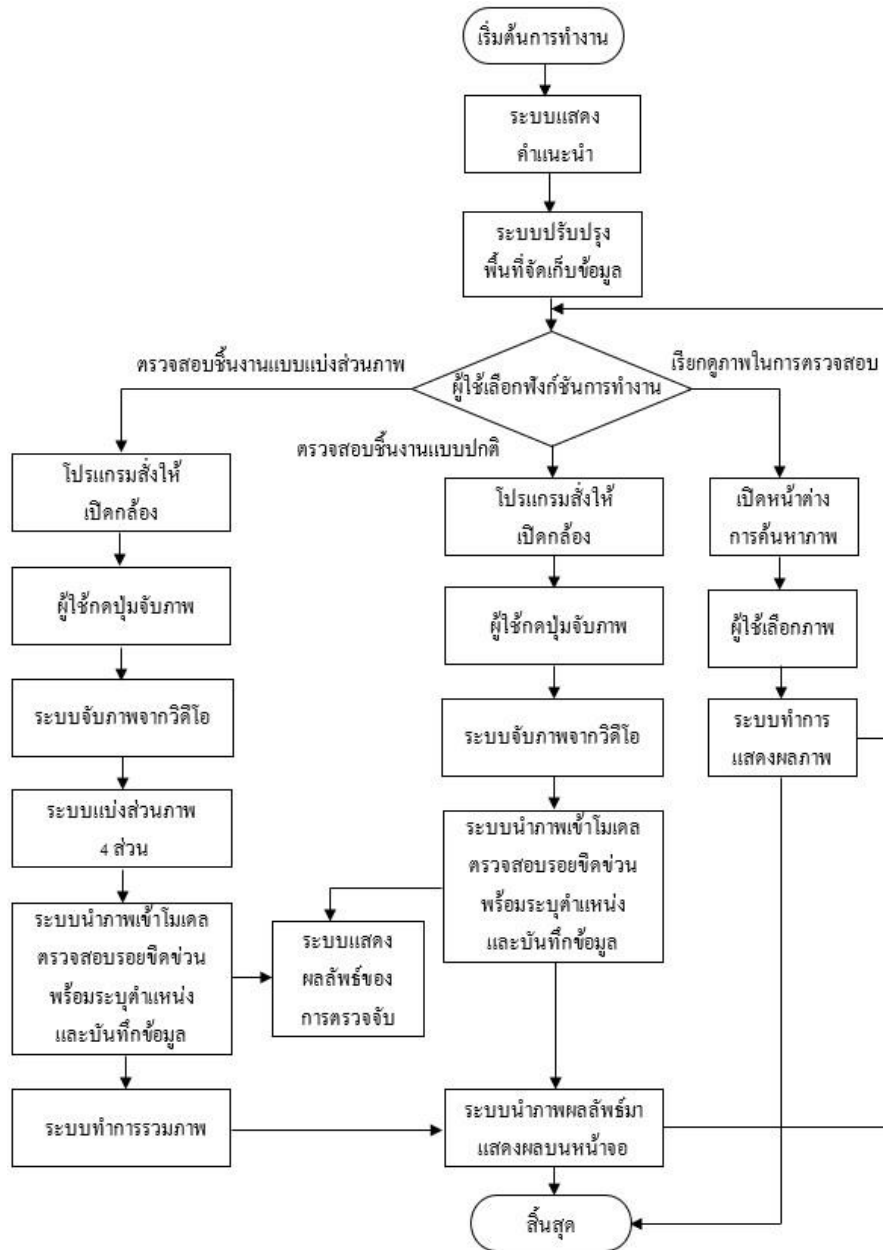
## 2) เรียกดูภาพการตรวจสอบ

- (1) เปิดหน้าต่างค้นหาภาพที่มีการประมวลผลจากโมเดลแล้ว
- (2) ผู้ใช้เลือกภาพ และระบบแสดงภาพยังบนหน้าจอ กรอบด้านขวา
- (3) รอผู้ใช้งานเลือกฟังก์ชันการทำงาน หรือสิ้นสุดการทำงานออกจาก

โปรแกรม

โดยแสดงแผนผังการทำงานของระบบตามภาพที่ 3.11



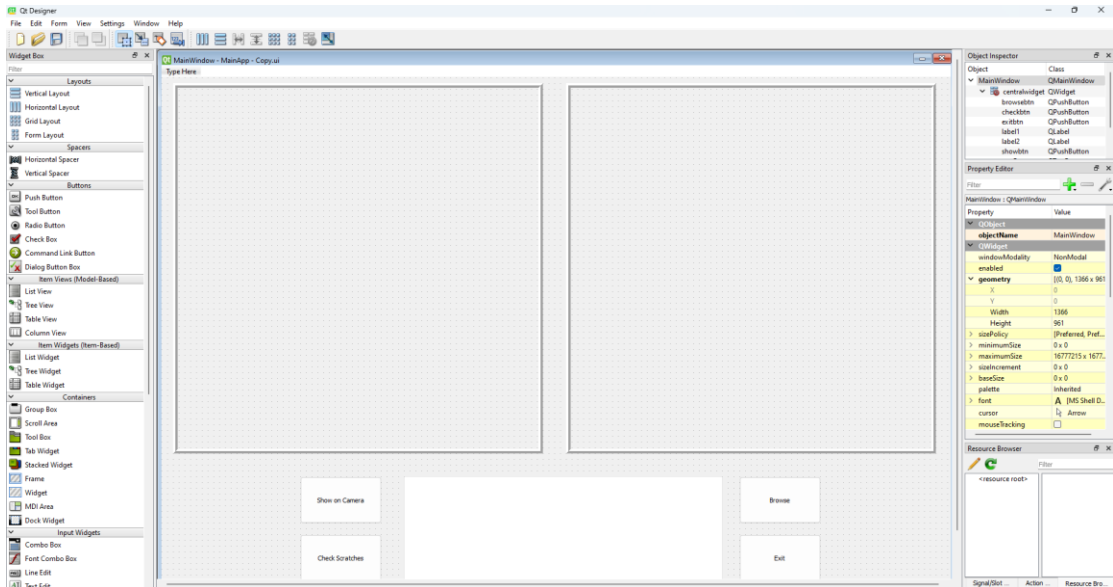


ภาพที่ 3.11 แผนผังการทำงานของระบบการตรวจจับรอยขีดข่วนบนผิวดูมึนเนียม

### 5.3 พัฒนาส่วนประสานผู้ใช้

การพัฒนาส่วนประสานผู้ใช้ (Graphic User Interface) ด้วยเครื่องมือ Visual Studio Code และออกแบบ GUI ด้วยโปรแกรม QT Designer ซึ่งเป็นเครื่องมือสำหรับออกแบบและสร้างตัวอินเทอร์เฟซกราฟิก (GUI) เป็นเครื่องมืออำนวยความสะดวกในการออกแบบและลดเวลาในการพัฒนา GUI เป็นอย่างมากเนื่องจากมีปุ่มหรือ object ในการสร้าง GUI มาให้ใช้งาน

ครบ เมื่อเทียบกับเครื่องมือในการออกแบบ GUI อย่างเช่น Tkinter ซึ่งต้องเขียนโค้ดทั้งหมด แสดงในภาพที่ 3.12



ภาพที่ 3.12 การออกแบบ User Interface ด้วยเครื่อง QT Designer

เมื่อทำการออกแบบจาก QT Design เสร็จแล้วทำการ Convert ไฟล์ของ QT Designer ให้เป็น ภาษา Python ด้วยคำสั่ง “convert : pyuic5 -x MainApp.ui -o MainApp.py”

#### 5.4 พัฒนาแอปพลิเคชัน

ทำการเขียนโค้ดคำสั่งโปรแกรมด้วยภาษา Python ด้วยโปรแกรม Visual Studio Code โดยมีลักษณะการพัฒนาเชิงวัตถุการเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) เป็นกระบวนการเขียนโปรแกรมที่ใช้แนวคิดของวัตถุเป็นศูนย์กลาง โดยตัวโปรแกรมจะถูกแบ่งเป็นองค์ประกอบหรือวัตถุที่มีคุณสมบัติและพฤติกรรม ดังภาพที่ 3.13

```

MainApp1.py > UI_MainApp2 > setting
..
MainApp2.setWindowTitle(_translate("MainApp2", "Aluminium Scratches Create by Mr.Nattee Malasarn STOU"))
self.btn1.setText(_translate("MainApp2", "Show on Camera"))
self.btn2.setText(_translate("MainApp2", "Check Scratches"))
self.btn3.setText(_translate("MainApp2", "Browse"))
self.btn4.setText(_translate("MainApp2", "Exit"))

self.btn1.clicked.connect(self.setImg)
self.btn2.clicked.connect(self.capture)
self.btn3.clicked.connect(self.browsefile)
self.btn4.clicked.connect(self.exit)

def cv2VDOcapture(self):
    self.capture = cv2.VideoCapture(0)

def setImg(self):
    # fname, _ = QtWidgets.QFileDialog.getOpenFileNames(None, "Select Image", "", "Image File (*.jpg *.png *.bmp)")
    # pixmap = QPixmap(fname)
    # pixmap = pixmap.scaled(self.label1.width(), self.label1.height())
    # self.label1.setPixmap(pixmap)
    self.cv2VDOcapture()
    self.timer = QTimer()

    self.timer.timeout.connect(self.update_frame)

    framerate = 30
    rate = int(1000/framerate)
    self.timer.start(rate)
    self.displayTxt.clear()
    self.btn2.setEnabled(True)

def update_frame(self):
    ret, frame = self.capture.read()
    if ret == True:
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        img = QImage(frame, frame.shape[1], frame.shape[0], QImage.Format_RGB888)
        pix = QPixmap(img)
        pix = pix.scaled(self.label1.width(), self.label1.height(), QtCore.Qt.KeepAspectRatio)
        pix2 = QPixmap.fromImage(img)
        self.label1.setPixmap(pix2)
        return frame

global cou
cou = ["result"]
def countmem(self):
    couA= len(cou)+ 1
    #print(couA)
    cou.append("result"+str(couA))

```

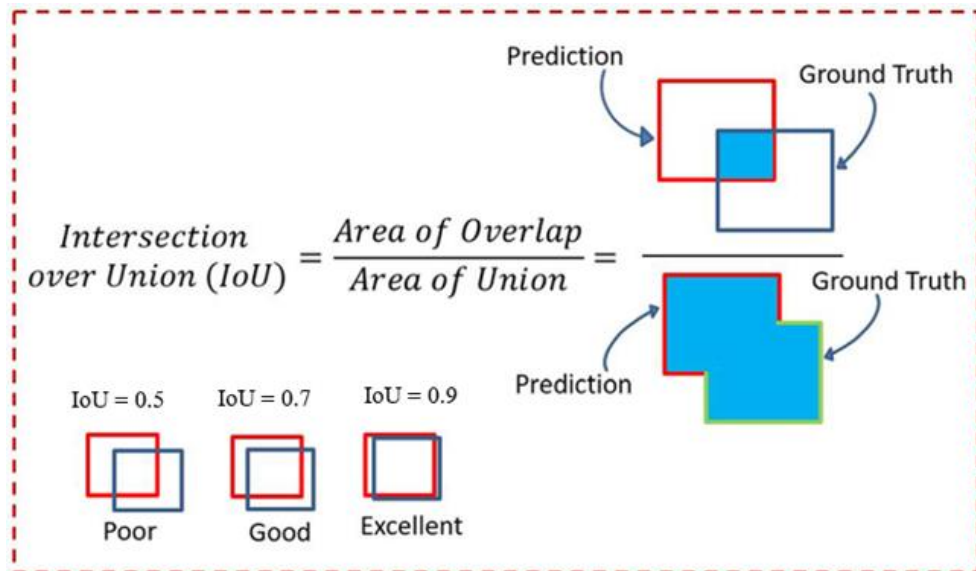
ภาพที่ 3.13 ภาพตัวอย่างการพัฒนาโปรแกรมด้วยภาษา Python

## 6. วิธีการประเมินประสิทธิภาพ

### 6.1 การประเมินประสิทธิภาพการตรวจจับวัตถุ

จะใช้วิธีประเมินความทับซ้อนของกรอบสี่เหลี่ยม Ground Truth เปรียบเทียบผลการทำนายกับผลเฉลย Bounding box โดยการหาค่า IoU (Intersection over Union) ผลลัพธ์ของค่า IoU สามารถบอกได้ว่าการตรวจจับนั้นถูกต้อง True Positive หรือไม่ถูกต้อง False Positive และ False Negative ซึ่งค่า IoU จะอยู่ในช่วง 0-1 จากภาพที่ 3.14 ค่าในกรอบสีแดงหมายถึงค่าผลเฉลย กรอบสีเทา หมายถึงผลการทำนาย จากภาพด้านซ้ายหากค่า IoU=0.5

หมายถึงผลการทำนายมีความถูกต้องไม่ดีมีผลการทำนายตรงกับผลเฉลยน้อย หากค่าเข้าใกล้ 1 มาก หมายถึงโมเดลมีการทำนายได้แม่นยำมากจึงใช้ค่านี้ในการวัดประสิทธิภาพของโมเดล



ภาพที่ 3.14 วิธีคำนวณค่า Intersection Over Union

## 6.2 การวัดประสิทธิภาพของโมเดล

ผู้ศึกษาได้ใช้เครื่องมือในการวัดประสิทธิภาพของโมเดลด้วยค่าดังต่อไปนี้

**6.2.1 ค่าความแม่นยำ (Accuracy)** เป็นตัวชี้วัดความแม่นยำในการตรวจจับวัตถุของโมเดล การคำนวณหาค่า Accuracy ได้จากสมการ

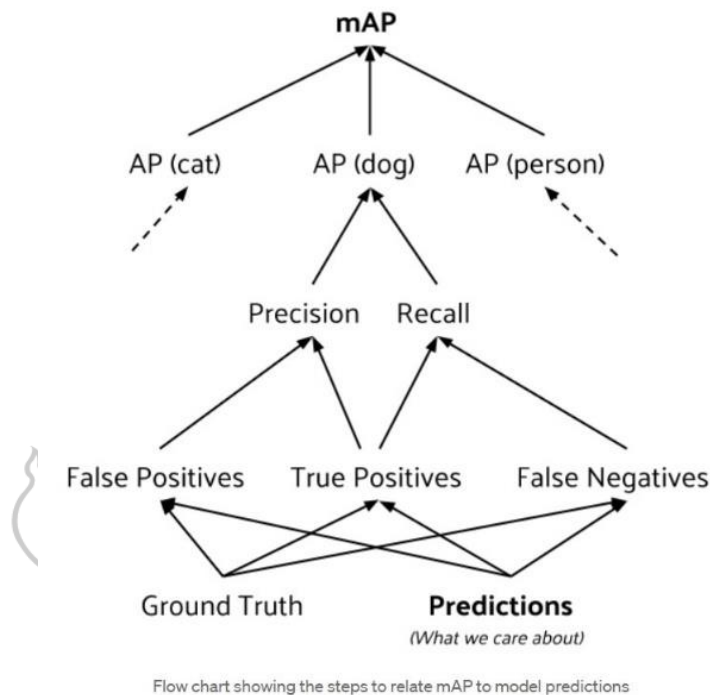
$Accuracy = (\text{จำนวนตัวอย่างที่ทำนายถูกต้อง}) / (\text{จำนวนตัวอย่างทั้งหมดในชุดทดสอบ})$

**6.2.2 ค่าความแม่นยำในคลาส (Precision)** เป็นตัววัดที่ใช้ในการประเมินความสามารถในการตรวจจับวัตถุของโมเดลว่ามีความแม่นยำในการตรวจจับวัตถุในคลาสนั้นมากน้อยเพียงใด โดยคำนวณจากสัดส่วนของจำนวนวัตถุที่โมเดลตรวจจับถูกต้อง เทียบกับจำนวนวัตถุที่โมเดลทำนายว่าเป็นวัตถุ ค่า Precision มีค่าเข้าใกล้ค่า 1 หมายถึงโมเดลยังมีประสิทธิภาพความแม่นยำสูง ดังสมการ  $Precision = (\text{จำนวนวัตถุที่ตรวจจับถูกต้อง}) / (\text{จำนวนวัตถุที่โมเดลทำนายว่าเป็นวัตถุ})$

**6.2.3 ค่าความครบถ้วน (Recall)** เป็นตัววัดที่ใช้ในการประเมินความสามารถในการตรวจจับวัตถุของโมเดลว่ามีความสามารถตรวจวัตถุครบถ้วนหรือไม่ โดยคำนวณจากสัดส่วนของจำนวนวัตถุที่โมเดลตรวจจับถูกต้อง เทียบกับจำนวนวัตถุที่เป็นจริง

ทั้งหมด ค่า Precision มีค่าเข้าใกล้ค่า 1 หมายถึงโมเดลยิ่งมีประสิทธิภาพความแม่นยำสูง ดังสมการ  $Recall = (\text{จำนวนวัตถุที่ตรวจจับถูกต้อง}) / (\text{จำนวนวัตถุที่เป็นจริงทั้งหมด})$

**6.2.4 ค่า Mean Average Precision (mAP50)** เป็นตัวชี้วัดประสิทธิภาพของโมเดลในการตรวจจับวัตถุที่กำหนดค่า threshold IoU (Intersection over Union) เท่ากับ 0.5 โดย mAP คือค่าเฉลี่ยของค่า Average Precision (AP) ของแต่ละคลาสของวัตถุที่ถูกตรวจจับโดยโมเดลโดยแสดงความสัมพันธ์ของค่าดังกล่าวที่ 3.15 โดยผลลัพธ์ที่โมเดลทำนายจะติกรอบ (bounding box) เทียบกับกรอบคำตอบ (Ground Truth) ด้วยค่า IoU = 0.5 โดยผลลัพธ์มีค่าเท่ากับหรือมากกว่า 0.5 จะถือว่าโมเดลทำนายได้ถูกต้อง ค่า mAP50 ยิ่งเข้าใกล้ 1 หมายถึงโมเดลยิ่งมีประสิทธิภาพตรวจจับวัตถุสูง



ภาพที่ 3.15 แสดงความสัมพันธ์ของค่า mAP

ที่มา: miro.medium.com (2023) Mean average precision (mAP).

## บทที่ 4

### ผลการวิเคราะห์ข้อมูล

#### 1. สรุปผลการพัฒนาโมเดล

จากการศึกษาการตรวจจับรอยขีดข่วนบนผิวอลูมิเนียมด้วยการเรียนรู้เชิงลึก CNN มีการฝึกสอนโมเดลโดยใช้ภาพในการฝึกสอน (Train) จำนวน 2,600 ภาพ สำหรับตรวจสอบ (Validation) จำนวน 246 ภาพ สำหรับทดสอบ (Testing) จำนวน 122 ภาพ ฝึกสอนบน Google Colab โดยการฝึกหยุดฝึกสอนโมเดลด้วยฟังก์ชัน patience Early Stopping ตั้งค่าความทนทานหากไม่มีการเปลี่ยนแปลงที่ดีที่สุดอยู่ที่ 50 epochs

ประเมินประสิทธิภาพโมเดลด้วยค่า mean Average Precision (mAP) เป็นค่าการหาค่าเฉลี่ยของในทุกคลาสของค่า Average Precision โดยจะมีการคำนวณค่า IoU ที่เป็นการนำค่าการทำนาย (Bounding box) กับค่าผลเฉลย (Ground Truth) โดยจะตั้งไว้ที่ IoU = 0.5 หากผลลัพธ์ mAP มีค่าเข้าใกล้ 1 มา แสดงถึงโมเดลประสิทธิภาพสูง โดยมีการเปรียบเทียบผลการทดลองของโมเดลทั้ง 3 แบบ ดังนี้

##### 1.1 เปรียบเทียบประสิทธิภาพของโมเดล

ทำการเปรียบเทียบประสิทธิภาพโมเดล YOLOv8n, YOLOv8s, YOLOv8x พบว่าค่าผลการประเมินประสิทธิภาพ mAP มีความใกล้เคียงกันของโมเดลทั้ง 3 แบบ แต่เวลาในการฝึกสอน YOLOv8x จะใช้เวลามาก การนำโมเดลไปใช้งานตรวจจับรอยขีดข่วนจากกล้องความละเอียดสูงในการศึกษาจะเลือกใช้ YOLOv8s เป็นแม่แบบเนื่องจากมีประสิทธิภาพในการตรวจจับและใช้เวลาในการตรวจจับอยู่ในเกณฑ์ที่เหมาะสม

ตารางที่ 4.1 เปรียบเทียบประสิทธิภาพของโมเดล



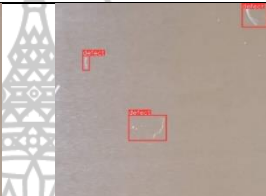
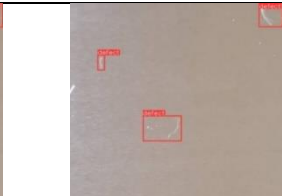

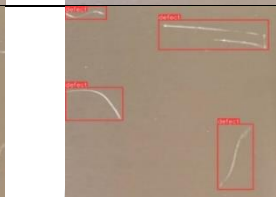
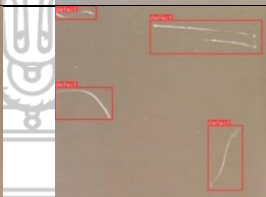





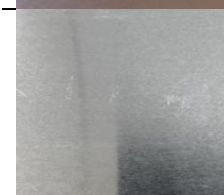


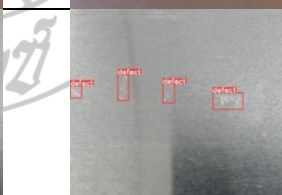

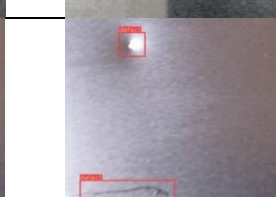

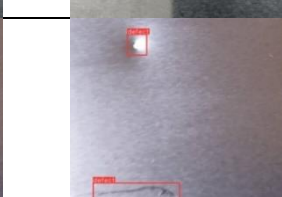
	YOLOv8n	YOLOv8s	YOLOv8x
Precision	91.9	89.5	89.3
Recall	81	87.7	87.3
mAP50	91.6	92.4	92.6
mAP50-90	58.9	59.5	61.5
LEANING TIME	2.5 Hour	2.8 Hour	10.8 Hour



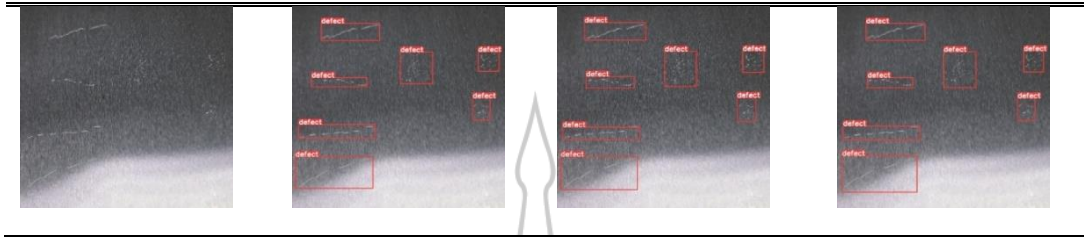
## 1.2 เปรียบเทียบการตรวจจับรอยขีดข่วนของโมเดลทั้ง 3 แบบ

ทำการเปรียบเทียบการตรวจจับรอยขีดข่วนของโมเดลทั้ง 3 แบบโดยกำหนดค่าพารามิเตอร์  $\text{conf} = 0.3$  ดังตารางที่ 4.2 ทำการตรวจจับรอยขีดข่วนทั้งหมด 6 ภาพ มีรอยขีดข่วน 23 รอย พบว่า YOLOv8n ตรวจจับได้ 21 รอย เปอร์เซนต์การตรวจจับเท่ากับ 91.3% YOLOv8s และ YOLOv8x มีประสิทธิภาพการตรวจจับเท่ากันคือตรวจจับได้ 22 รอย เปอร์เซนต์การตรวจจับเท่ากับ 95.6%

ตารางที่ 4.2 เปรียบเทียบผลลัพธ์การตรวจจับรอยขีดข่วนด้วยโมเดล YOLOv8 n,s,x

Image	YOLOv8n	YOLOv8s	YOLOv8x
			
			
			
			
			

ตารางที่ 4.2 (ต่อ)



### 1.3 เปรียบเทียบประสิทธิภาพการตรวจจับรอยขีดข่วนและเวลาที่ใช้ในการตรวจจับ

โดยทำการเปรียบเทียบการตรวจจับใช้ภาพทั้งหมด 20 ภาพ มีรอยขีดข่วนทั้งหมด 83 รอย แบ่งออกเป็น 2 ความละเอียดคือ ความละเอียด 640x640 pixel จำนวน 10 ภาพ มีรอยขีดข่วนทั้งหมด 29 รอย และความละเอียด 1200x1200 pixel จำนวน 10 ภาพ มีรอยขีดข่วนทั้งหมด 54 รอย จากตารางทดสอบผลลัพธ์ตัวอักษรสีแดง คือกรณีตรวจจับไม่ครบ False Negative (FN) และตัวอักษรสีน้ำเงิน คือตรวจจับซ้ำ การตรวจจับซ้ำสามารถแก้ไขโดยเพิ่มค่า IoU ในการทำ Non-Maximum Suppression เพื่อลดการตรวจจับซ้ำกันได้ T/A หมายถึงเวลาเฉลี่ยที่ใช้ในการประมวลผลของทุกภาพ ผลลัพธ์การเปรียบเทียบประสิทธิภาพแสดงในตารางที่ 4.3 พบว่าโมเดล YOLOv8n ตรวจจับไม่ได้ 4 รอย FN=4.82% เวลาเฉลี่ยการตรวจจับต่อหนึ่งภาพ 335.3ms, YOLOv8s ตรวจจับไม่ได้ 3 รอย FN=3.6% เวลาเฉลี่ยการตรวจจับต่อหนึ่งภาพ 679.9ms, YOLOv8x ตรวจจับไม่ได้ 2 รอย FN=2.4% เวลาเฉลี่ยการตรวจจับต่อหนึ่งภาพ 3,272.5ms, โมเดล YOLOv8x ใช้เวลาในการตรวจจับมากจึงไม่เหมาะกับการตรวจจับแบบเรียลไทม์

ตารางที่ 4.3 เปรียบเทียบประสิทธิภาพผลลัพธ์การตรวจจับรอยขีดข่วน

Image	Image size (pixel)	Total defect	YOLOv8n		YOLOv8s		YOLOv8x	
			detect	time	detect	time	detect	time
Image1/10	640 x 640	1	1	215ms	1	294.2ms	1	2132.8ms
Image2/10	640 x 640	1	1	136.6ms	1	290.2ms	1	1468.1ms
Image3/10	640 x 640	1	1	140.6ms	1	308.2ms	1	1641.6ms

## ตารางที่ 4.3 (ต่อ)


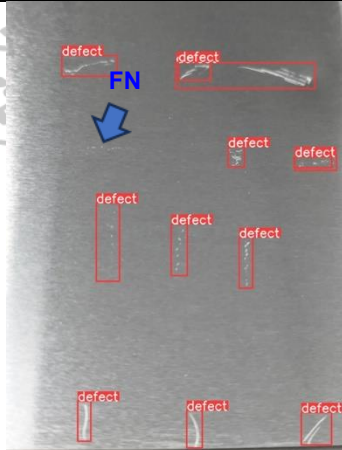
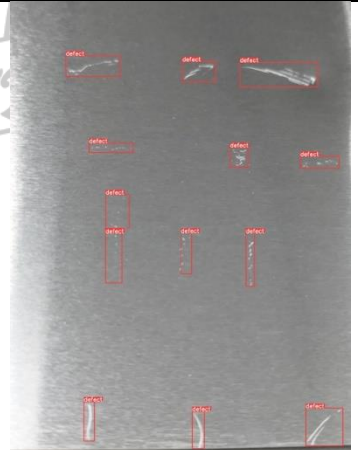
Image4/10	640 x 640	4	4	145.6ms	4	303.2ms	5	1497.6ms
Image5/10	640 x 640	4	4	126.7ms	4	297.2ms	5	1522.9ms
Image6/10	640 x 640	4	3	134.6ms	3	301.2ms	4	1647.6ms
Image7/10	640 x 640	2	2	135.6ms	2	289.2ms	2	1454.1ms
Image8/10	640 x 640	3	3	141.6ms	3	294.2ms	3	1801.2ms
Image9/10	640 x 640	5	4	141.6ms	5	294.2ms	5	1828.1ms
Image10/10	640 x 640	4	2	213.4ms	3	288.2ms	3	1828.1ms
Sum		29	FN 4=13.8%	T/A 153.1ms	FN 2=6.89%	T/A 296ms	FN 1=3.44%	T/A 1682.2ms

Image	Image size (pixel)	Total defect	YOLOv8n		YOLOv8s		YOLOv8x	
			detect	time	detect	time	detect	time
Image1/10	1200 x 1200	9	9	519.6ms	9	1104.2ms	9	4967.7ms
Image2/10	1200 x 1200	4	4	493.7ms	4	1149.9ms	4	4747.3ms
Image3/10	1200 x 1200	2	2	492.7ms	2	1044.2ms	2	4824.1ms
Image4/10	1200 x 1200	8	8	503.7ms	8	1032.2ms	8	4832.1ms
Image5/10	1200 x 1200	3	3	497.7ms	3	1024.3ms	3	4871ms
Image6/10	1200 x 1200	5	5	542.6ms	4	1142.9ms	4	5026.6ms
Image7/10	1200 x 1200	4	4	537.6ms	4	1044.2ms	4	4804.2ms
Image8/10	1200 x 1200	3	3	519.6ms	3	1035.2ms	3	4850.1ms
Image9/10	1200 x 1200	7	7	555.5ms	7	1020.3ms	7	4882ms
Image10/10	1200 x 1200	9	9	512.6ms	9	1041.2ms	9	4823ms
Sum		54	FN 0=100%	T/A 517.5ms	FN 1=1.85%	T/A 1063.8ms	FN 1=1.85%	T/A 4862.8ms
Summary		83	FN 4=4.82%	T/A 335.3ms	FN 3=3.6%	T/A 679.9ms	FN 2=2.4%	T/A 3272.5ms

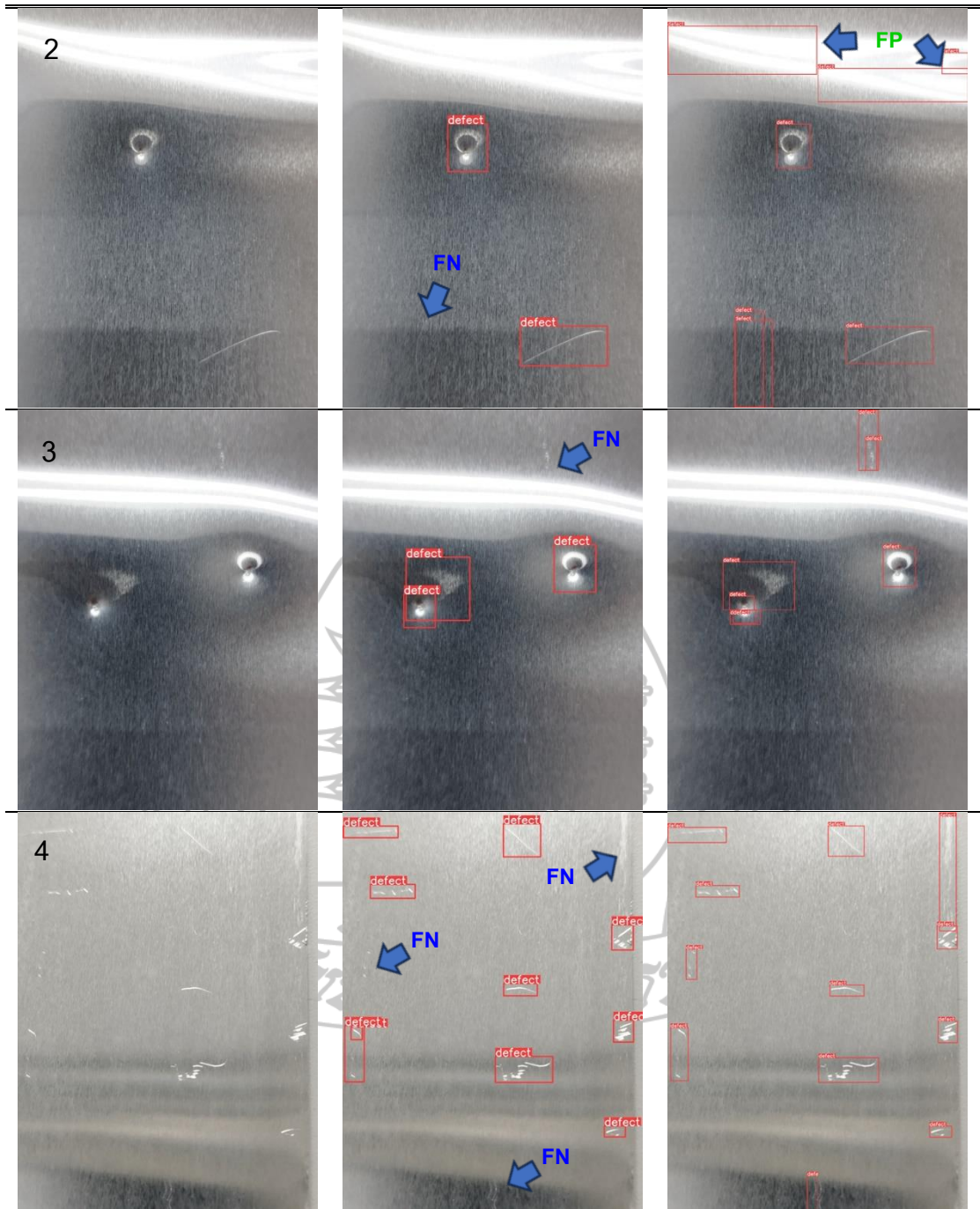
#### 1.4 เปรียบเทียบผลลัพธ์การตรวจจ็รอยขีดข่วนด้วยเทคนิคการแบ่งส่วนภาพ

โดยทำการแบ่งภาพเป็นส่วนย่อย 4 ส่วน (Subdivision Image) จากภาพอินพุตที่มีความละเอียดสูงขนาด 3,000 x 4,000 pixels แบ่งเป็น 4 ส่วนแต่ละส่วนมีขนาด 1,500 x 2,000 pixels เทียบกับภาพอินพุตที่ไม่ได้ใช้เทคนิคการแบ่งส่วนภาพ พบว่าโมเดลสามารถตรวจสอบรอยขีดข่วนได้ดีกว่าที่ไม่ได้ใช้เทคนิคการแบ่งภาพ โดยมีรอยขีดข่วนที่โมเดลที่ไม่ได้ใช้เทคนิคการแบ่งภาพตรวจไม่พบจำนวน 11 รอย จากจำนวน 49 รอย ซึ่งเป็นรอยขีดข่วนที่มีขนาดเล็กมากและไม่ชัดเจน โมเดลที่ใช้เทคนิคการแบ่งภาพสามารถตรวจพบเพิ่มขึ้นอีก 10 รอยจาก 11 รอย มีเปอร์เซ็นต์ตรวจพบเพิ่มขึ้นถึง 90.9% ดังผลลัพธ์ในตารางที่ 4.4 และทำการวิเคราะห์ผลลัพธ์จากการแบ่ง 4 ส่วน ยังตรวจไม่พบ 1 รอย จึงได้ทำการแบ่งส่วนเพิ่มเป็น 9,16,36,64 ส่วน พบว่าสามารถตรวจจ็รอยที่การแบ่ง 4 ส่วนตรวจไม่พบแต่ยังทำการแบ่งส่วนมากเวลาในการประมวลจะมากขึ้นตามไปด้วยและเมื่อนำภาพมารวมกันหลังจากโมเดลตรวจสอบเสร็จสิ้นพบว่ามีกรณีการตีกรอบ bounding box หลายกรอบใน 1 วัตถุ หากรอยขีดข่วนนั้นมีพื้นที่อยู่หลายส่วนจากการแบ่งส่วน มีผลลัพธ์ดังตารางที่ 4.5

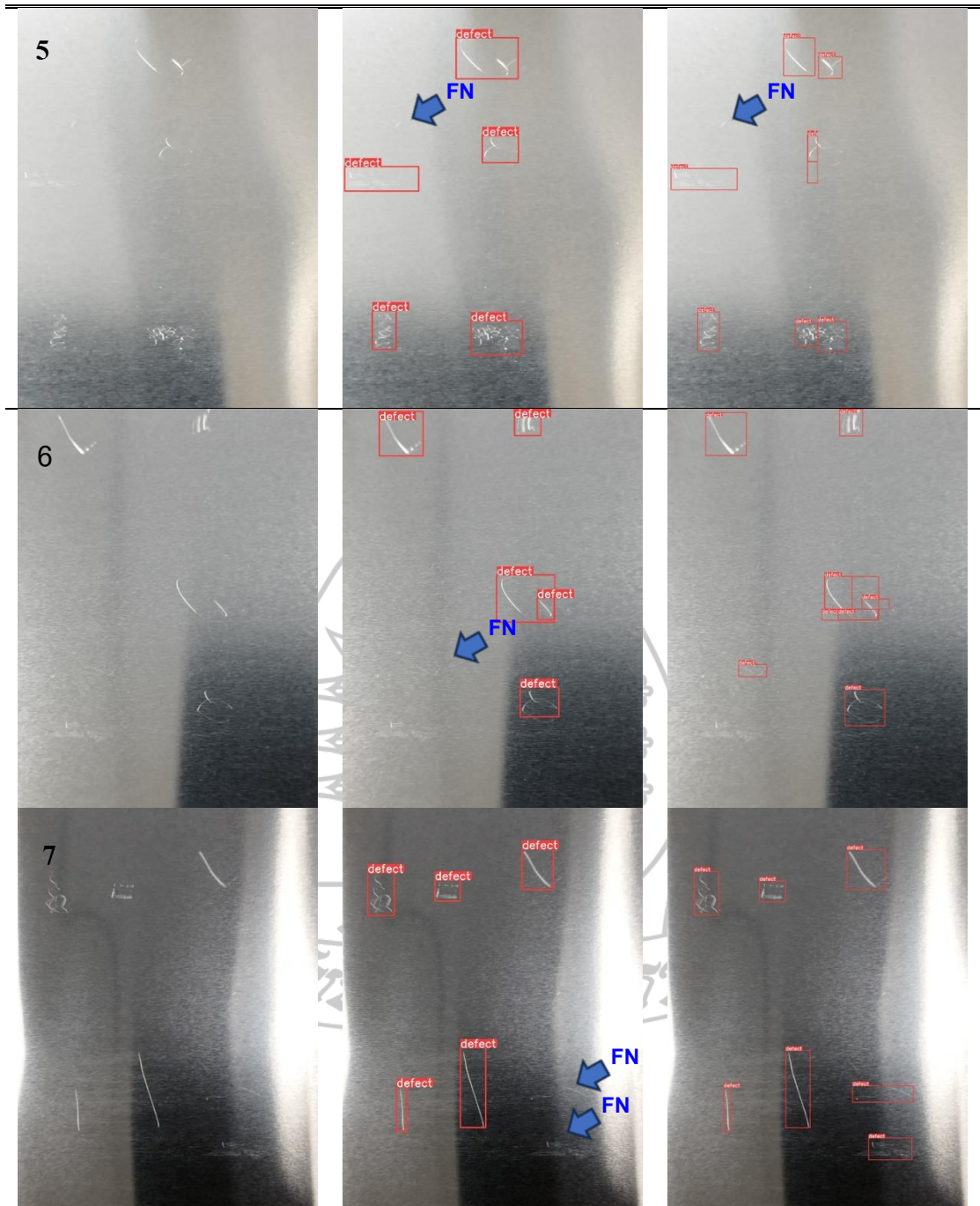
ตารางที่ 4.4 เปรียบเทียบผลลัพธ์การใช้เทคนิคการแบ่งส่วนภาพ

Scratches = 48 Image = 7	FN= 11	FN=1 , FP=3
Master image size = 3000x4000 pixel	Image size = 3000x4000 pixel	Split = 4 part Image size per picce 1500x2000 pixel
Master	Predict non Subdision	Predict Sundivision
		

ตารางที่ 4.4 (ต่อ)



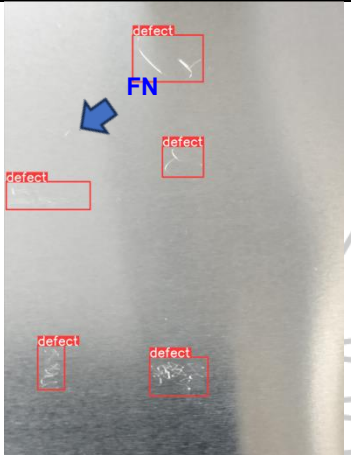
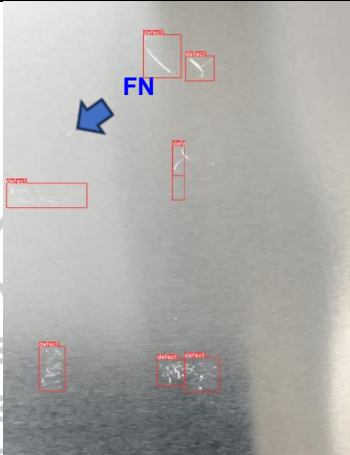
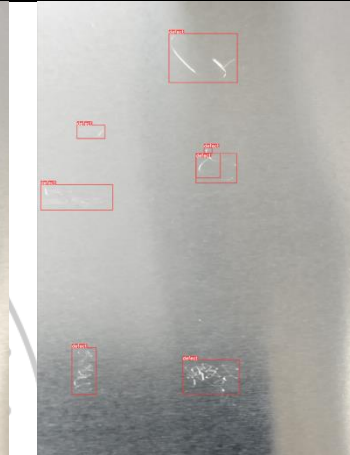
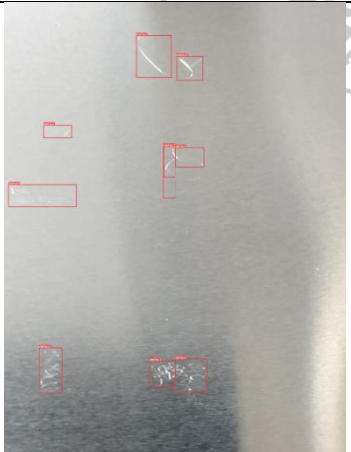


ตารางที่ 4.4 (ต่อ)



ตารางที่ 4.5 เปรียบเทียบผลลัพธ์การใช้เทคนิคการแบ่งส่วนภาพมากกว่า 4 ส่วน

จากผลการทดลองรูปที่ 5 เมื่อทำการแบ่ง 4 ส่วนยังไม่สามารถตรวจจับได้ จึงได้ทดลองแบ่งออกเป็น 4,9,16,36,64 ส่วน สรุปผลได้ดังนี้

- 1) เมื่อยิ่งแบ่งให้ภาพมีขนาดเล็กลงมากเท่าไรเวลาเฉลี่ยใช้ในการตรวจจับจะมากขึ้นตามไปด้วย
- 2) รอยขีดข่วนที่มีขนาดเล็กตรวจจับได้ดีขึ้นแต่เมื่อรวมภาพเข้าด้วยกันจะมี Bounding box ซ้อนกันหลายอัน

Predict non subdivision 3000x4000 pixel Time Average = 720 ms	Predict subdivision 4 part 1500x2000 pixel Time Average = 1043 ms	Predict subdivision 9 part 1000x1333 pixel Time Average = 2791 ms
		
Predict subdivision 16 part 750x1000 pixel Time Average = 5786 ms	Predict subdivision 36 part 500x667 pixel Time Average = 8364 ms	Predict subdivision 64 part 375x500 pixel Time Average = 14337 ms
		

## 1.5 การวิเคราะห์ผลลัพธ์จากการฝึกฝนโมเดล YOLOv8s

### 1.5.1 วิเคราะห์ผลลัพธ์ค่า mAP50

ซึ่งเป็นค่า Average Precision ของทุกคลาสมาคำนวณค่าเฉลี่ยที่ค่า IoU = 0.5 ยิ่งค่าผลลัพธ์เข้าใกล้ค่า 1 มากเท่าไรประสิทธิภาพของโมเดลยิ่งมีประสิทธิภาพมากขึ้นตามไปด้วยดังแสดงในภาพที่ 4.1

```

Ultralytics YOLOv8.0.143 Python-3.10.6 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8s summary (fused): 168 layers, 11125971 parameters, 0 gradients
val: Scanning /content/AluminiumDefect.Vi8 testset/Aluminium-Defect.vi8i.yolo/v8/valid/labels cache... 246 i
Class Images Instances Box(P R mAP50 mAP50-95): 100% ██████████
all 246 636 0.895 0.877 0.924 0.595

Speed: 3.8ms preprocess, 10.2ms inference, 0.0ms loss, 2.5ms postprocess per image
Results saved to runs/detect/val
ultralytics.utils.metrics.DetMetrics object with attributes:

ap_class_index: array([0])
box: ultralytics.utils.metrics.Metric object
confusion_matrix: <ultralytics.utils.metrics.ConfusionMatrix object at 0x7fc467744d30>
fitness: 0.627665154070227
keys: ['metrics/precision(B)', 'metrics/recall(B)', 'metrics/mAP50(B)', 'metrics/mAP50-95(B)']
maps: array([[ 0.59475]])
names: {0: 'defect'}
plot: True
results_dict: {'metrics/precision(B)': 0.895038589227972, 'metrics/recall(B)': 0.8773584905660378, 'metri
0.627665154070227}
save_dir: PosixPath('runs/detect/val')
speed: {'preprocess': 3.7869331313342585, 'inference': 10.20363288197091, 'loss': 0.004249859631546144, '

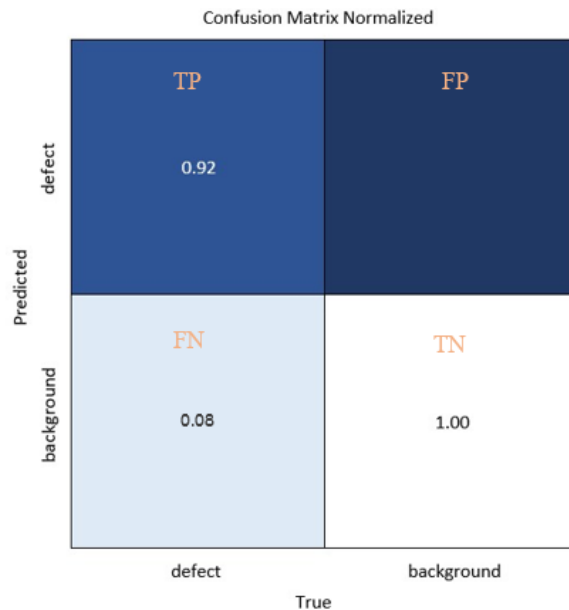
```

ภาพที่ 4.1 แสดงผลลัพธ์ของกระบวนการตรวจสอบโมเดล

### 1.5.2 วิเคราะห์ผลลัพธ์ตาราง Confusion Matrix

โดยดูที่ค่า TP (True Positive) ยิ่งค่าเข้าใกล้ค่า 1 มาก ก็ยิ่งมีประสิทธิภาพการตรวจจับวัตถุที่แม่นยำเข้าใกล้ 100 เปอร์เซ็นต์ จากผลลัพธ์ในตารางที่ 4.2 ค่า TP มีค่าเท่ากับ 0.92 ซึ่งเป็นค่าที่เข้าใกล้ค่า 1 ซึ่งแสดงให้เห็นว่าโมเดลมีประสิทธิภาพในการตรวจจับวัตถุสูง ค่า FP (False Positive) คือค่าการตรวจจับวัตถุที่ไม่ใช่วัตถุที่ต้องการแต่บอกว่าเป็นวัตถุที่ต้องการ ค่า FN (False Negative) คืออีก 1 ค่าที่ให้ความสำคัญเนื่องจากเป็นค่าที่โมเดลตรวจจับวัตถุไม่สามารถตรวจจับวัตถุที่ต้องการได้ ยังมีค่าน้อยหรือเข้าใกล้ค่า 0 โมเดลมีประสิทธิภาพในการตรวจจับวัตถุสูง





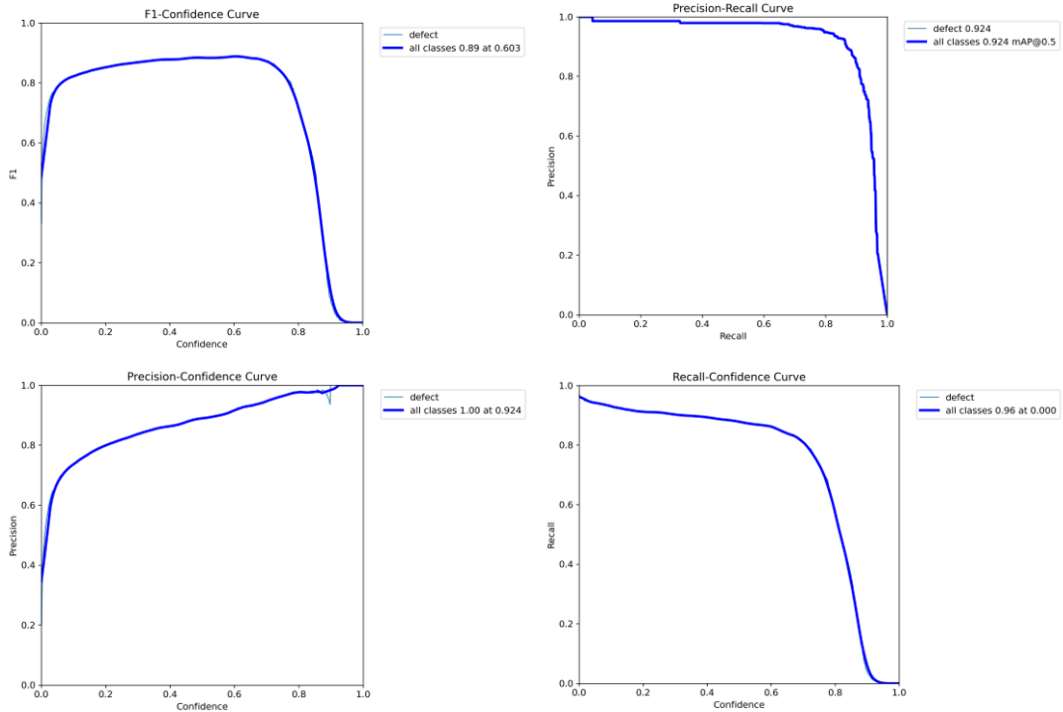
ภาพที่ 4.2 แสดงตาราง Confusion Matrix

### 1.5.3 กราฟแสดงความสัมพันธ์

เป็นกราฟที่ใช้ในการวิเคราะห์ความสัมพันธ์ระหว่างค่าสองค่าที่สอดคล้องกันดังภาพที่ 4.3

- 1) *F1-Confidence Curve* เป็นกราฟแสดงความสัมพันธ์ค่า F1 Score กับค่า Confidence Threshold ในการตรวจจับวัตถุ จากภาพที่ 4.3 ค่า Confidence สูงสุดอยู่ที่ 0.89 เมื่อเพิ่ม Threshold มากกว่า 0.89 ค่า F1 Score จะเริ่มลดลง
- 2) *Precision-Recall Curve* เป็นกราฟที่แสดงความสัมพันธ์ระหว่างค่า Precision และค่า Recall ของโมเดลตามค่า Threshold ที่ต่าง ๆ ในการตรวจจับวัตถุ การวัด Precision-Recall Curve ช่วยให้เห็นภาพรวมของประสิทธิภาพของโมเดลในการตรวจจับโดยไม่จำเป็นต้องตั้งค่า Threshold ที่แน่นอน
- 3) *Precision-Confidence Curve* เป็นกราฟที่แสดงความสัมพันธ์ระหว่างค่า Precision และค่า Confidence ในการตรวจจับวัตถุว่ามีวัตถุที่ตรวจจับอยู่หรือไม่ ซึ่งทั้งสองค่ามีความสัมพันธ์กันค่าความแม่นยำมีค่าสูงค่าความมั่นใจจะยิ่งสูงขึ้นเข้าใกล้ค่า 1 ตามไปด้วย
- 4) *Recall-Confidence Curve* เป็นกราฟที่แสดงให้เห็นถึงความสัมพันธ์ระหว่างความมั่นใจ Confidence ในการตรวจจับวัตถุของโมเดล และค่า

ความสามารถในการครอบคลุมวัตถุ Recall ที่แตกต่างกันตาม Confidence Threshold ที่กำหนด



ภาพที่ 4.3 แสดงกราฟความสัมพันธ์ของแต่ละค่าในการตรวจจับวัตถุ

#### 1.5.4 วิเคราะห์กราฟผลการฝึกฝนโมเดล

*Train/box\_loss* เป็นกราฟที่แสดงค่าความสูญเสียที่เกิดขึ้นในการฝึกโมเดลในขั้นตอนของการตรวจจับวัตถุ (object detection) โดยเฉพาะอย่างยิ่งในโมเดลที่ใช้เทคนิค YOLO (You Only Look Once) ซึ่งเป็นโมเดลที่สำหรับตรวจจับและจำแนกวัตถุในภาพที่มีความเร็วและความแม่นยำสูง และเป็นที่ยอมรับในวงการ Computer Vision ค่าความสูญเสียในกราฟ *train/box\_loss* จะแสดงให้เห็นถึงความคลาดเคลื่อนของการทำนายของกรอบสี่เหลี่ยม (bounding box) ที่ใช้ในการคาดคะเนตำแหน่งและขนาดของวัตถุในภาพ โดยตัวแปรที่กำหนดค่าความสูญเสียนี้จะระบุถึงความแตกต่างระหว่างตำแหน่งของกรอบคำตอบ (Ground Truth) กับตำแหน่งของกรอบที่โมเดลทำนาย (bounding box) การลดค่าความสูญเสียเป็นการเพิ่มความแม่นยำของการทำนายของกรอบสี่เหลี่ยมในการตรวจจับวัตถุ จากกราฟในภาพที่ 4.4 *train/box\_loss* จะเห็นได้ว่ามีค่าที่ลดลงอย่างมีนัยยะสำคัญจากรอบที่เริ่มต้นจนถึงรอบสุดท้าย

*val/box\_loss* เป็นกราฟค่าความสูญเสียที่เกิดขึ้นในขั้นตอนการตรวจสอบ จากกราฟที่ภาพที่ 4.4 แสดงให้เห็นถึงค่าความคลาดเคลื่อนลดลงเรื่อยๆ ในรอบการฝึกฝนที่สูงขึ้น หากกราฟไม่มีการเปลี่ยนแปลงที่ดีขึ้นหรือค่ากราฟเริ่มมีค่าความคลาดเคลื่อนเพิ่มขึ้น ควรหยุดการฝึกสอน

*val/cls\_loss* เป็นกราฟค่าความสูญเสียที่เกิดขึ้นในขั้นตอนการตรวจสอบ จะแสดงให้เห็นถึงความคลาดเคลื่อนของการจำแนกคลาสวัตถุ (object classification) จากกราฟในรอบการฝึกสอนที่สูงขึ้นความผิดพลาดในการจำแนกวัตถุลดลง ทำให้ประสิทธิภาพของโมเดลในการจำแนกวัตถุมีประสิทธิภาพสูงขึ้น

*metrics/precision* เป็นกราฟที่แสดงค่าความแม่นยำของการตรวจจับวัตถุ ในขั้นตอนของการทดสอบ (evaluation) โดยใช้ตัววัด Precision ซึ่งเป็นหนึ่งในตัววัดที่ใช้ในการประเมินความสามารถในการตรวจจับวัตถุของโมเดล ตัววัด Precision จะบอกถึงสัดส่วนของจำนวนวัตถุที่โมเดลตรวจจับถูกต้อง เทียบกับจำนวนวัตถุที่โมเดลทำนายว่าเป็นวัตถุ ยิ่งมีค่าเข้าใกล้ 1 แสดงถึงความแม่นยำสูงของโมเดลในการตรวจจับวัตถุ ดังสมการดังนี้

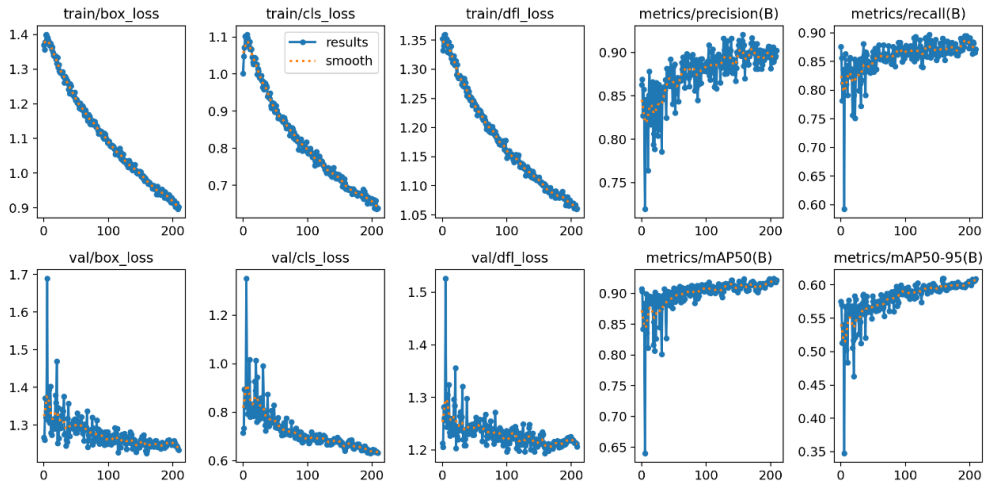
$$\text{Precision} = (\text{จำนวนวัตถุที่ตรวจจับถูกต้อง}) / (\text{จำนวนวัตถุที่โมเดลทำนายว่าเป็นวัตถุ})$$
 จากกราฟในภาพที่ 4.4 แสดงให้เห็นถึงค่าของ precision ประมาณ 8.95 ซึ่งเข้าใกล้ 1 มาก แสดงถึงประสิทธิภาพของโมเดลมีประสิทธิภาพในการตรวจวัตถุสูง

*metrics/recall* เป็นกราฟที่แสดงค่าความแม่นยำของการตรวจจับวัตถุ ในขั้นตอนการทดสอบของการทดสอบ (evaluation) โดยใช้ Recall ในการประเมินประสิทธิภาพของโมเดล ว่ามีความสามารถตรวจวัตถุครบถ้วนหรือไม่ โดยคำนวณจากสัดส่วนของจำนวนวัตถุที่โมเดลตรวจจับถูกต้อง เทียบกับจำนวนวัตถุที่เป็นจริงทั้งหมด ดังสมการ  $\text{Recall} = (\text{จำนวนวัตถุที่ตรวจจับถูกต้อง}) / (\text{จำนวนวัตถุที่เป็นจริงทั้งหมด})$  จากภาพที่ 4.4 พบว่าค่า *metrics/recall* มีค่า 8.77 เข้าใกล้ 1 ถือว่าโมเดลมีประสิทธิภาพที่ดี

*metrics/mAP50* เป็นกราฟที่แสดงค่า Mean Average Precision (mAP) ของการตรวจจับวัตถุที่ค่า threshold IoU (Intersection over Union) เท่ากับ 0.5 ในขั้นตอนการทดสอบ โดย mAP คือค่าเฉลี่ยของค่า Average Precision (AP) ของแต่ละคลาสของวัตถุที่ถูกตรวจจับโดยโมเดล โดยผลลัพธ์ที่โมเดลทำนายและตีกรอบ (bounding box) เทียบกับกรอบคำตอบ (Ground Truth) ด้วยค่า IoU = 0.5 โดยผลลัพธ์มีค่าเท่ากับหรือมากกว่า 0.5 จะถือว่าโมเดลทำนายได้ถูกต้อง จากภาพที่ 4.4 mAP มีค่าเท่ากับ 0.924 ซึ่งมีค่าเข้าใกล้ 1 ถือว่าโมเดลมีประสิทธิภาพดี

*metrics/mAP50-95* เป็นกราฟที่แสดงค่า Mean Average Precision (mAP) ของการตรวจจับวัตถุที่ค่า threshold IoU (Intersection over Union) 0.5 ถึง 0.95 โดยผลลัพธ์ที่โมเดลทำนายและตีกรอบ (bounding box) เทียบกับกรอบคำตอบ (Ground Truth) ด้วยค่า

IoU ในช่วง 0.5 ถึง 0.95 โดยผลลัพธ์มีค่าเท่ากับหรือมากกว่า 0.5-0.95 จะถือว่าโมเดลทำนายได้ถูกต้อง จากภาพที่ 4.4 mAP50-95 มีค่าเท่ากับ 0.595 ซึ่งอยู่ในเกณฑ์ที่พึงพอใจ



ภาพที่ 4.4 กราฟสรุปผลการฝึกฝนและตรวจสอบของโมเดล

ภาพตัวอย่างในการส่งข้อมูลเข้าไปฝึกฝนโมเดลโดยมีการกำหนดคำตอบคลาสของวัตถุ และส่งเข้าฝึกฝนด้วยขนาด batch size = 16 ตามภาพที่ 4.5 ด้านซ้าย และภาพตัวอย่างในการตรวจสอบ Validation แสดงในภาพที่ 4.5 ด้านขวา



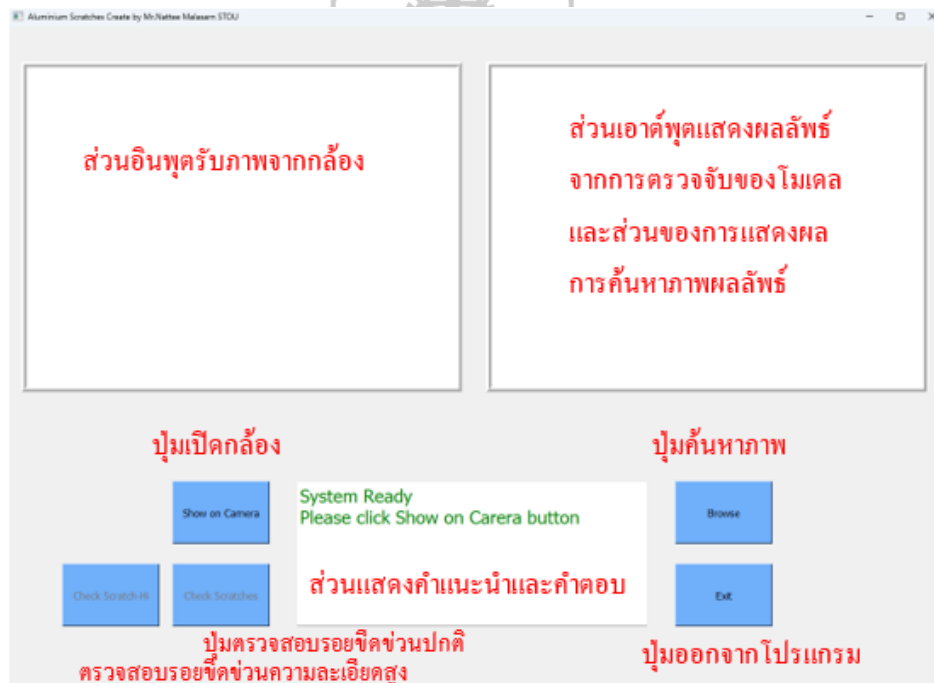
ภาพที่ 4.5 แสดงตัวอย่าง Train\_batch และ Validate\_batch

## 2. สรุปผลการพัฒนาส่วนประสานผู้ใช้ GUI

โปรแกรม UI ที่พัฒนาขึ้นจาก Python และ PyQt5 ดังที่กล่าวไว้ในวิธีดำเนินการศึกษา ข้อ 5.3 หน้า 41 พัฒนาส่วนประสานผู้ใช้ Graphic User Interface ด้วยเครื่องมือ Visual Studio Code และออกแบบ GUI ด้วยโปรแกรม QT Designer ซึ่งสามารถออกแบบและพัฒนา GUI ให้มีฟังก์ชันครบถ้วนตามความต้องการ โดยมีส่วนการทำงาน 4 ส่วนดังนี้

1. แสดงภาพอินพุตที่ได้จากกล้อง
2. ส่วนแสดงผลภาพผลลัพธ์ในการตรวจจับจากโมเดลและภาพที่ค้นหา
3. ส่วนแสดงคำตอบและคำแนะนำ
4. ส่วนปุ่มคำสั่งงานของโปรแกรม ปุ่มตรวจสอบรอยขีดข่วนจะถูกปิดการทำงานหากยังไม่ได้กดปุ่มเปิดกล้องหรือกล้องยังไม่พร้อมทำงาน

ผลลัพธ์ของโปรแกรมส่วนประสานผู้ใช้งาน GUI แสดงตามภาพที่ 4.6

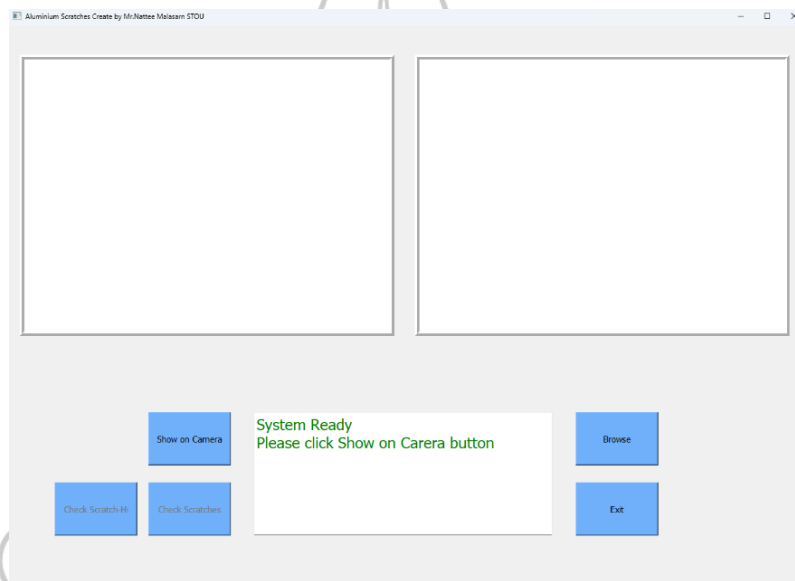


ภาพที่ 4.6 แสดงหน้าจอโปรแกรมตรวจสอบรอยขีดข่วนบนผิวโลหะอลูมิเนียม

### 3. ทดสอบและสรุปผลการทำงานของโปรแกรม

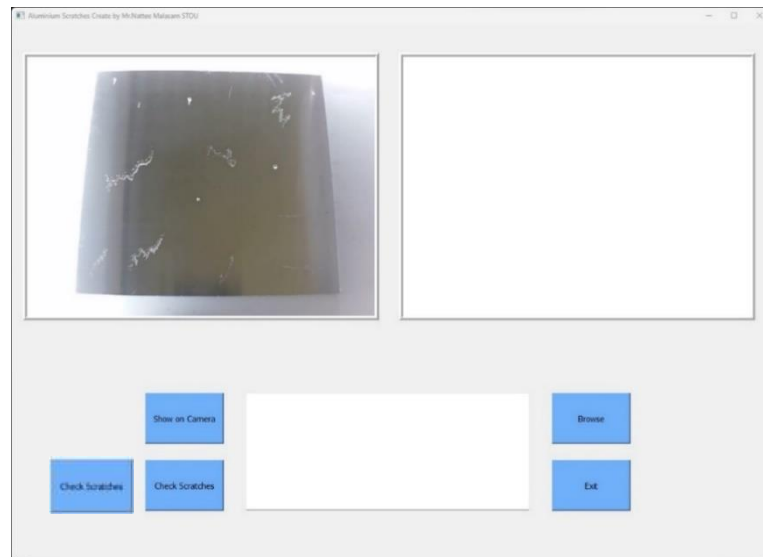
ทำการทดสอบการทำงานของโปรแกรมทั้งสองฟังก์ชันคือ ตรวจสอบชิ้นงานแบบแบ่งส่วนภาพสำหรับภาพที่มีความละเอียดสูงและตรวจสอบชิ้นงานแบบปกติ โดยทดสอบชิ้นงานที่มีรอยขีดข่วน และชิ้นงานไม่มีรอยขีดข่วน พบว่าสามารถแสดงผลลัพธ์ได้ถูกต้องและทำการระบุตำแหน่งของรอยขีดข่วนได้ถูกต้องตรงจุด ฟังก์ชันการค้นหาภาพสามารถค้นหาภาพในการตรวจสอบก่อนหน้าได้ โดยทำการทดสอบการทำงานของโปรแกรมมีขั้นตอนดังนี้

1. เริ่มจากเปิดโปรแกรมจะมีข้อความคำแนะนำ โปรแกรมพร้อมทำงานและให้กดปุ่มเปิดกล้อง หากต้องการออกจากโปรแกรมให้กดปุ่ม Exit หรือปุ่มกากบาทด้านบนขวามือ ดังภาพที่ 4.7



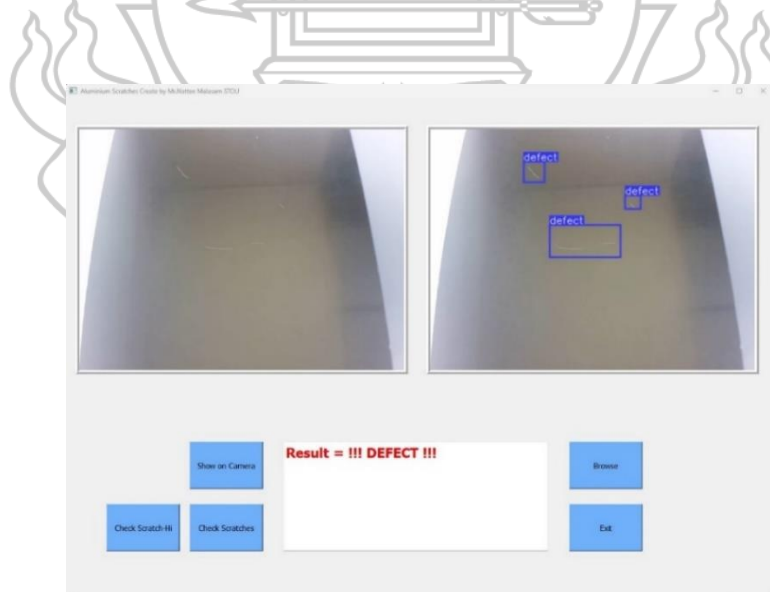
ภาพที่ 4.7 แสดงการเริ่มต้นการทำงานของโปรแกรม

2. เปิดกล้องเพื่อเริ่มทดสอบ (Show on Camera) จะปรากฏภาพจากกล้องขึ้นบนกรอบแสดงภาพอินพุตด้านซ้ายมือ ดังภาพที่ 4.8



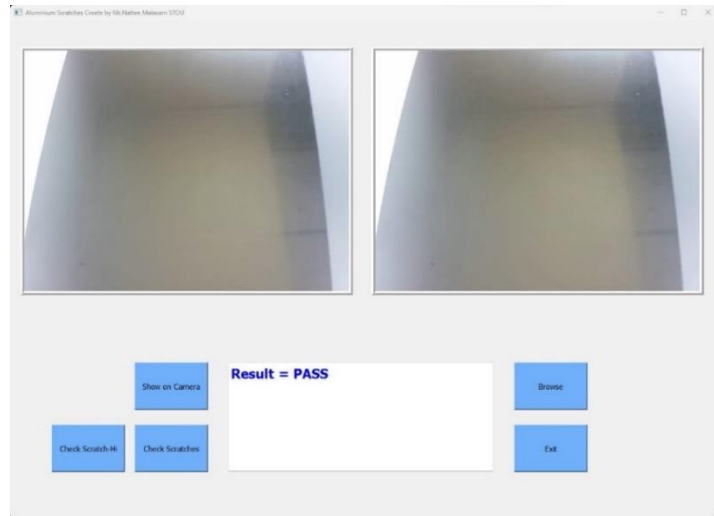
ภาพที่ 4.8 แสดงหน้าจอโปรแกรมเริ่มทำการประมวลผล

3. เมื่อทำการกดปุ่มเปิดกล้องภาพจะปรากฏที่กรอบอินพุตด้านซ้ายและปุ่มตรวจสอบรอยขีดข่วนจะเปิดการทำงาน (Check Scratches) และเมื่อกดปุ่มตรวจสอบรอยขีดข่วนภาพการตรวจจับจะแสดงยังกรอบเอาต์พุตด้านขวาดังในภาพที่ 4.9 แสดงผลลัพธ์ตรวจพบรอยขีดข่วนและจะมีข้อความ **Result = !!! DEFECT !!!** แจ้งในส่วนของช่องแสดงผลลัพธ์ที่ตั้งแสดงในภาพที่ 4.9



ภาพที่ 4.9 แสดงหน้าจอโปรแกรมตรวจจับรอยขีดข่วนบนชิ้นงานเสีย

ทดสอบชิ้นงานดีแสดงในภาพที่ 4.10



ภาพที่ 4.10 แสดงหน้าจอโปรแกรมตรวจจับรอยขีดข่วนบนชิ้นงานดี





## บทที่ 5

### สรุปผลการศึกษา และข้อเสนอแนะ

#### 1. สรุปผลการศึกษา

จากการศึกษาพัฒนาโมเดลและประเมินประสิทธิภาพ ระบบการตรวจจับรอยขีดข่วนบนผิวอลูมิเนียม ได้ทำการฝึกสอนโมเดล YOLOv8 ด้วยข้อมูลการฝึกสอนจำนวน 2,600 ภาพ สำหรับตรวจสอบ 246 ภาพ และสำหรับทดสอบ 122 ภาพ โดยฝึกสอนโมเดล YOLOv8 3 แบบที่มีการออกแบบสถาปัตยกรรมที่มีความซับซ้อนแตกต่างกัน คือ YOLOv8n , YOLOv8s , YOLOv8x ประเมินค่าประสิทธิภาพโมเดลด้วยค่า mean Average Precision (mAP50) มีค่าเท่ากับ 0.916, 0.924, 0.926 ตามลำดับ ทดสอบประสิทธิภาพโมเดลด้วยตรวจจับรอยขีดข่วนทั้งหมด 83 รอย บนภาพ 20 ภาพ โดยมีความละเอียด 640x640 pixel จำนวน 10 ภาพ และความละเอียด 1200x1200 pixel จำนวน 10 ภาพ พบว่า YOLOv8n ตรวจจับไม่ได้ 4 รอย ใช้เวลาเฉลี่ยในการตรวจจับเท่ากับ 333.5ms ค่า Accuracy = 95.1% YOLOv8s ตรวจจับไม่ได้ 3 รอย ใช้เวลาเฉลี่ยในการตรวจจับเท่ากับ 679.3ms ค่า Accuracy = 96.3% YOLOv8x ตรวจจับไม่ได้ 2 รอย ใช้เวลาเฉลี่ยในการตรวจจับเท่ากับ 3,272.5ms ค่า Accuracy = 97.5% ซึ่ง YOLOv8x มีค่า mAP50 สูงที่สุดแต่ใช้เวลาในการทำนายผลมากที่สุด ส่วนโมเดล YOLOv8s มีค่า mAP50 เท่ากับ 0.924 ที่ใกล้เคียงกับโมเดล YOLOv8x แต่ใช้เวลาเฉลี่ยในการตรวจจับรอยขีดข่วนลดลงถึง 79.24% และมีประสิทธิภาพการตรวจจับลดลงเพียง 1.2% จากผลการทดสอบข้างต้นของ YOLOv8s พบว่าตรวจจับรอยขีดข่วนไม่ได้อยู่จำนวน 3 รอย คิดเป็นเปอร์เซ็นต์เท่ากับ 3.6%

สำหรับการใช้เทคนิคการแบ่งภาพอินพุตที่มีความละเอียดสูงเป็นเทคนิคที่ให้ผลลัพธ์ในการตรวจจับรอยขีดข่วนได้เพิ่มขึ้นด้วยการแบ่งขนาดภาพอินพุตเป็น 4 ส่วน จากผลการทดสอบกับภาพอินพุตที่มีความละเอียดขนาด 3000x4000 pixel โมเดลที่ไม่ได้ใช้เทคนิคการแบ่งภาพตรวจจับรอยขีดข่วนไม่พบจำนวน 11 รอย เนื่องจากเป็นรอยขีดข่วนที่มีขนาดเล็กและไม่ชัดเจน แต่เมื่อใช้เทคนิคการแบ่งภาพก่อนนำเข้าโมเดล CNN สามารถตรวจพบรอยเพิ่มขึ้นอีก 10 รอยจากทั้งหมด 11 รอย คิดเป็นร้อยละ 90.9% และรอยที่ตรวจไม่พบ 1 รอย เป็นรอยขีดข่วนที่มีขนาดเล็กมาก สามารถตรวจพบได้ด้วยวิธีการแบ่งภาพให้เป็นภาพย่อยลงอีกแต่ต้องไม่

เล็กไปกว่าความละเอียดของภาพที่โมเดลต้องการ เมื่อทำการแบ่งย่อยภาพมากเท่าใดจะใช้เวลาในการตรวจจับมากขึ้นตามไปด้วย

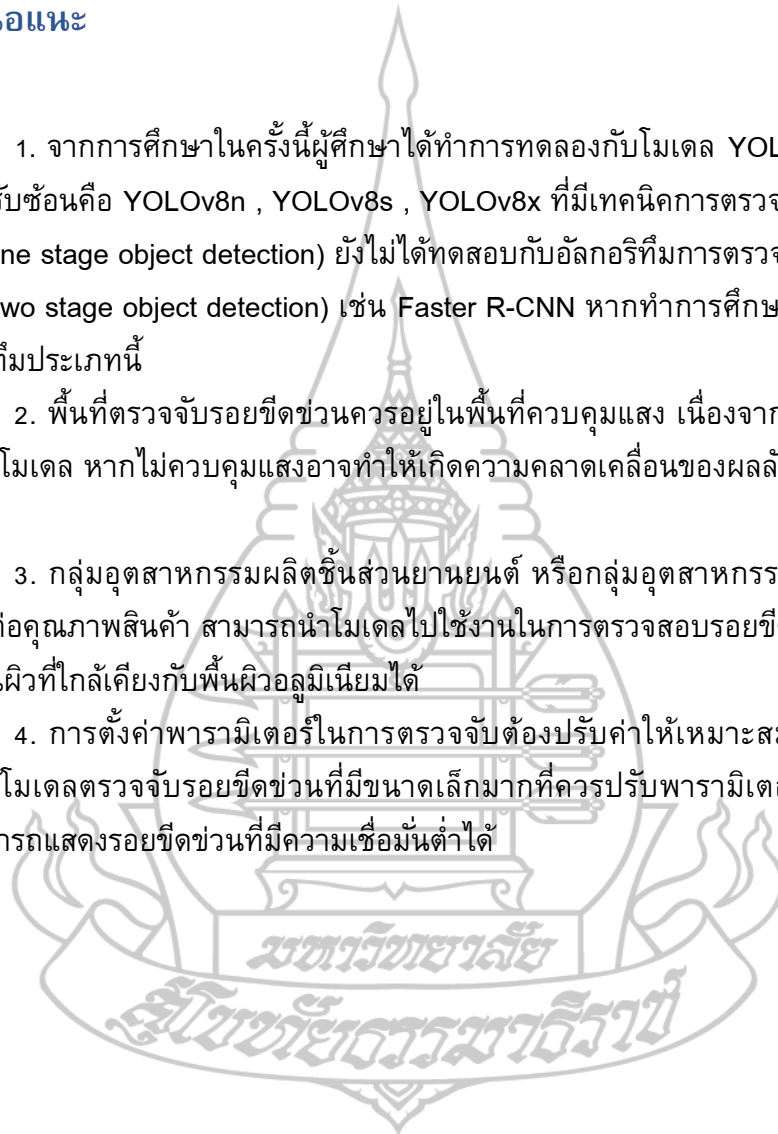
## 2. ข้อเสื่อนอแนะ

1. จากการศึกษาในครั้งนี้ผู้ศึกษาได้ทำการทดลองกับโมเดล YOLOv8 ใน 3 ระดับของความซับซ้อนคือ YOLOv8n , YOLOv8s , YOLOv8x ที่มีเทคนิคการตรวจจับเป็นแบบหนึ่งขั้นตอน (One stage object detection) ยังไม่ได้ทดสอบกับอัลกอริทึมการตรวจจับวัตถุแบบสองขั้นตอน (Two stage object detection) เช่น Faster R-CNN หากทำการศึกษาต่อควรทดสอบกับอัลกอริทึมประเภทนี้

2. พื้นที่ตรวจจับรอยขีดข่วนควรอยู่ในพื้นที่ควบคุมแสง เนื่องจากแสงมีผลกับการทำงานของโมเดล หากไม่ควบคุมแสงอาจทำให้เกิดความคลาดเคลื่อนของผลลัพธ์ในการทำงานของโมเดล

3. กลุ่มอุตสาหกรรมผลิตชิ้นส่วนยานยนต์ หรือกลุ่มอุตสาหกรรมอื่นๆ ที่รอยขีดข่วนส่งผลต่อคุณภาพสินค้า สามารถนำโมเดลไปใช้งานในการตรวจสอบรอยขีดข่วนกับวัสดุที่มีลักษณะพื้นผิวที่ใกล้เคียงกับพื้นผิวอลูมิเนียมได้

4. การตั้งค่าพารามิเตอร์ในการตรวจจับต้องปรับค่าให้เหมาะสมกับชิ้นงานหากต้องการให้โมเดลตรวจจับรอยขีดข่วนที่มีขนาดเล็กมากที่ควรปรับพารามิเตอร์ความมั่นใจลงเพื่อให้สามารถแสดงรอยขีดข่วนที่มีความเชื่อมั่นต่ำได้



## บรรณานุกรม

- กรณีการ์ มุลโพธิ์. (2565). หุ่นยนต์คู่กายคัดแยกผลจากโดยใช้อัลกอริทึม Yolo และบรรจุภัณฑ์ในกระบวนการผลิตน้ำดื่ม. มหาวิทยาลัยเทคโนโลยีมหานคร, กรุงเทพมหานคร.
- ขวัญกมล ดิฐกัญจน์, ปิยะวัฒน์ หนูเล็ก และกรวิทย์ พฤษชัยนิมมิต. (2565). วินิจฉัยมะเร็งผิวหนัง  
จากภาพถ่ายโทรศัพท์ด้วยโครงข่ายประสาทเทียมสังวัตนาการ.  
มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต, ภูเก็ต.
- จักรกฤษ อ่อนชื่นจิตร. (2564). การตรวจสอบข้อบกพร่องวงจรมินิปต์ด้วยเครื่องมือทางการประมวลผลภาพ. วิทยานิพนธ์ปริญญาคุชฎบัณฑิต. มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี, ปทุมธานี.
- ชวิต ภูมิพัฒน์. (2563). ประสิทธิภาพของระบบตรวจจับวัตถุเคลื่อนที่กรณีศึกษา ฟาร์มนกแอ่นกินรัง  
ปริญญาวิทยาศาสตรมหาบัณฑิต. สถาบันเทคโนโลยีไทย-ญี่ปุ่น, กรุงเทพมหานคร.
- เชิด คงห้อย และ นิคม สุวรรณวร. (2563). เทคนิคการคัดเลือกคุณภาพของตัวอ่อนปูแสมด้วยวิธีการ  
ตรวจจับการเคลื่อนที่. วารสารวิจัย มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย, สงขลา.
- ตะวัน ขุนอาสา. (2558). ระบบตรวจจับและคัดแยกป้ายจราจร. มหาวิทยาลัยเทคโนโลยีราชมงคล  
สุวรรณภูมิ, พระนครศรีอยุธยา.
- พิมพ์า ชีวาประกอบกิจ. (2562). การปรับปรุงประสิทธิภาพในการจำแนกภาพด้วยโครงข่ายประสาทแบบคอนโวลูชันโดยใช้เทคนิคการเพิ่มภาพ. มหาวิทยาลัยนานาชาติเอเชีย-แปซิฟิก, กรุงเทพมหานคร.
- พุดิพงษ์ จันท์แจ่ม. (2563). การปรับปรุงประสิทธิภาพของกระบวนการบริการลูกค้าโดยใช้การ

วิเคราะห์จากกล้องวงจรปิด (วิทยานิพนธ์ วิทยาศาสตร์มหาบัณฑิต). สถาบันบัณฑิต  
พัฒน บริหารศาสตร์, กรุงเทพมหานคร.

มณีนุช นาทาทอง และ นิวัตร์ อังควิศิษฐพันธ์. (2563). การจำแนกชิ้นส่วนโทรศัพท์มือถือด้วย  
เทคนิคการประมวลผลภาพ. มหาวิทยาลัยมหาสารคาม, กรุงเทพมหานคร.

รัตน์โชติ พันธุ์วิล. (2562). การตรวจต้นไม้เป็นโรคอติโนมิติด้วยภาพถ่ายมุมสูงจากโดรนและ  
การ

เรียนรู้เชิงลึก (วิทยานิพนธ์ ปริญญาวิศวกรรมศาสตรมหาบัณฑิต). มหาวิทยาลัยธุรกิจ  
บัณฑิตย์, กรุงเทพมหานคร.

ราตรี คำโห่ง. (2565). การตรวจจับมัจจุคด้วยการเรียนรู้เชิงลึก. มหาวิทยาลัยราชภัฏ  
อุตรดิตถ์, อุตรดิตถ์.

อัฐพงศ์ สงข์เพชร และ วีระ สอิ่ง. (2566). ระบบตรวจนับและจำแนกประเภทยานพาหนะอติ  
โนมิติ

สำหรับถนนในเขตเมืองด้วยหลักการโครงข่ายแบบคอนโวลูชัน (วิทยานิพนธ์ ปริญญา  
วิทยาศาสตรมหาบัณฑิต). มหาวิทยาลัยศรีนครินทรวิโรฒ, กรุงเทพมหานคร.

*A Gentle Introduction to YOLOv8.* (2023, July 8). Wandb.ai.

[https://wandb.ai/mukilan/wildlife-yolov8/reports/A-Gentle-Introduction-to-YOLOv8-  
-Vmllldzo0MDU5NDA2](https://wandb.ai/mukilan/wildlife-yolov8/reports/A-Gentle-Introduction-to-YOLOv8-Vmllldzo0MDU5NDA2)

Aviral Bhardwaj. (2021, Jul 6). *An introduction to machine learning algorithms.*

[https://iaviral.medium.com/what-are-neural-networks-an-introduction-to-machine-  
learning-algorithms-6b73383c9089](https://iaviral.medium.com/what-are-neural-networks-an-introduction-to-machine-learning-algorithms-6b73383c9089)

Bryan Tan. (2020, May 3). *Guide to Car Detection using YOLO.*

<https://towardsdatascience.com/guide-to-car-detection-using-yolo-48caac8e4ded>

Deval Shah. (2022, March 7). *Mean Average Precision (mAP) Explained: Everything You  
Need to Know.* <https://www.v7labs.com/blog/mean-average-precision>

- Documents Ultralytics YOLOv8*. (2023). Ultralytics. <https://docs.ultralytics.com>
- Khathinhorm, N., Samankitesakul, R., Laesanklang, W., Horbanluekit, B., & Amornsamankul, S. (2022). Deep Learning and Image Processing for Disc Brake Pad Identification: A Case Study of Brake Pads Company. *Journal of Engineering and Digital Technology (JEDT)*, 10(2), 21-30. <https://ph01.tci-thaijo.org/index.php/TNIJournal/article/view/247946>
- Liao, D., Cui, Z., Zhang X., Li, J., Li, W., Zhu, Z., & Wu, N. (2022). Surface defect detection and classification of Si<sub>3</sub>N<sub>4</sub> turbine blades based on convolutional neural network and YOLOv5 *Advances in Mechanical Engineering*. *Advances in Mechanical Engineering*, 14(2), 1-12. <https://journals.sagepub.com/doi/pdf/10.1177/16878132221081580>
- Jearanaitanakij, K., Iamthammarak, K., & Wangcharoen, N. (2021, January 1). Fast Classifying Non-helmeted Motorcyclists by Using Convolutional Neural Networks. *SNRU Journal of Science and Technology*, 13(1), 1-8. [https://ph01.tci-thaijo.org/index.php/snru\\_journal/article/view/242823/165705](https://ph01.tci-thaijo.org/index.php/snru_journal/article/view/242823/165705)
- Jiao, L., Zhang, F., Liu, F., & Yang, S. (2019, September 5). *A Survey of Deep Learning-Based Object Detection*. [https://commons.wikimedia.org/wiki/File:Object\\_detector\\_1stage\\_vs\\_2\\_stage.png](https://commons.wikimedia.org/wiki/File:Object_detector_1stage_vs_2_stage.png)
- PradyaSin. (2019, August 16). *Convolution Neural Network*. <https://medium.com/@pradyasin/what-is-convolution-neural-network-bf2e525089f5>
- Dwyer, B. & Gallagher, J. (2023, March 16). *Getting Started with Roboflow*. <https://blog.roboflow.com/getting-started-with-roboflow>
- Seri Lee. (2021, August 13). *Padding in Neural Network*. <https://dev.to/sally20921/padding-in-neural-network-o8m>

Sharma, N., Baral, S., Paing, M. L., & Chawuthai, R. (2023). Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms. *Sensors*, 23(13), 6-12.

<https://doi.org/10.3390/s23135843>

Sumit Singh. (2023, January 5). *Leveraging YOLO Object Detection for Accurate and Efficient Visual Recognition*. <https://www.labellerr.com/blog/why-is-the-yolo-algorithm-important>

*Ultralytics YOLOv8: The State-of-the-Art YOLO Model*. (2023). YOLOv8.

<https://ultralytics.com/yolov8>

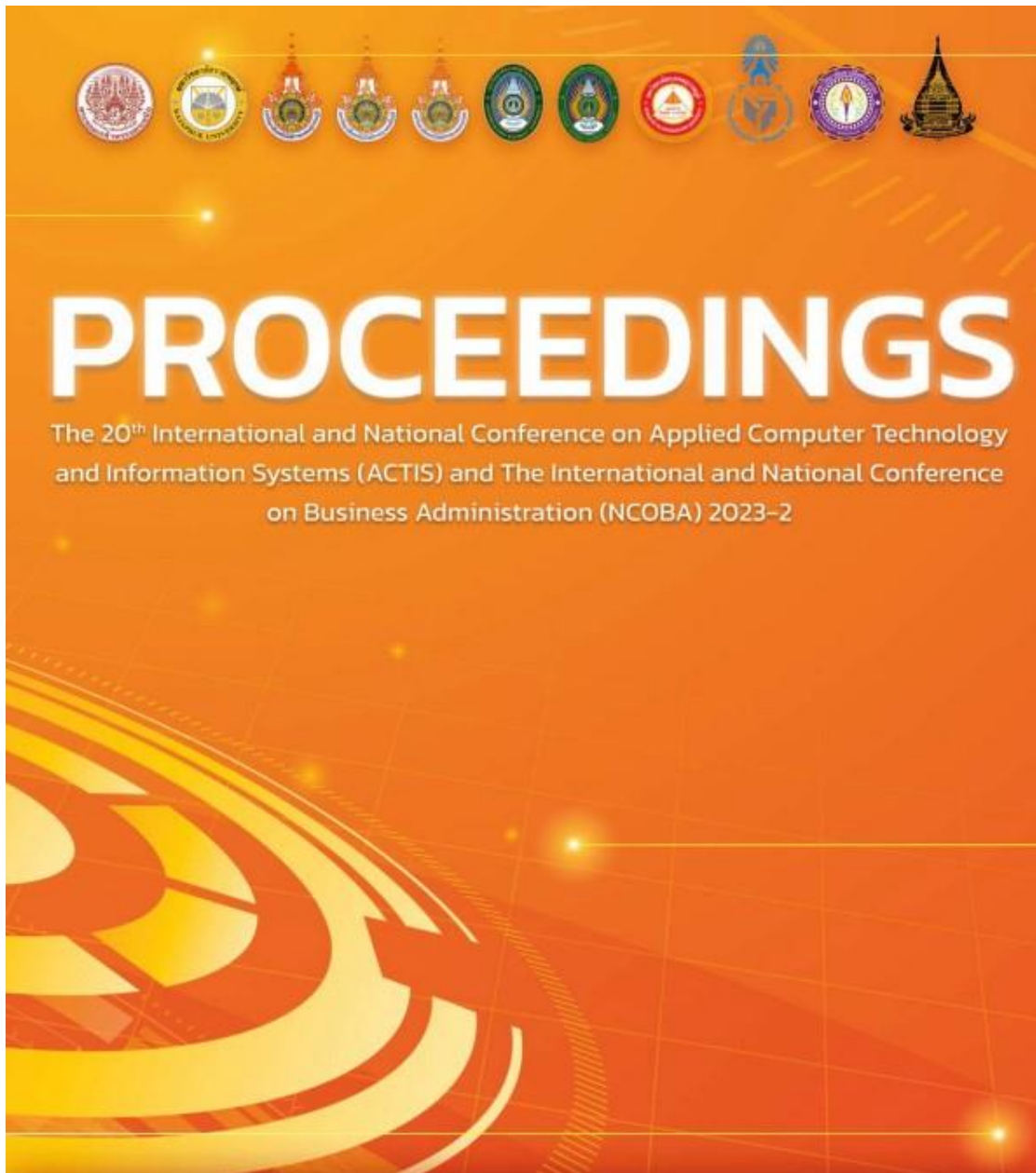


## ภาคผนวก

### บทความวิจัยที่ได้รับการตีพิมพ์งานประชุมวิชาการระดับชาติและระดับนานาชาติ

นัฒธี มละสาร (2566) “การตรวจจับพื้นผิวรอยขีดข่วนของวัสดุโลหะโดยใช้วิธีการประมวลผลการแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน” ใน งานประชุมวิชาการระดับชาติและระดับนานาชาติด้านเทคโนโลยีคอมพิวเตอร์ประยุกต์และระบบสารสนเทศ ครั้งที่ 20 วันที่ 25 สิงหาคม 2566 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ และมหาวิทยาลัยราชพฤกษ์ กรุงเทพมหานคร หน้า 46-50





**AiAT**  
ARTIFICIAL INTELLIGENCE ASSOCIATION OF THAILAND

**Online conference**  
**25 August 2023**

<http://conference.rpu.ac.th/actis2023>



**การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผลภาพ  
และโครงข่ายประสาทแบบคอนโวลูชัน**  
**Scratched Surface Detection of Metal Objects Using Processing Methods of Image Subdivision  
and Convolution Neural Network**

นัฒธี มละสาร<sup>1</sup>, เตชศุทธิ์ ฟิลิปป์ เพ็ญชัย<sup>2</sup>

สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช  
9/9 หมู่ 9 ต.บางพูน อ.ปากเกร็ด จ.นนทบุรี 11120 โทรศัพท์ : 0-2504-8281  
E-mail: <sup>1</sup>2649600257@stou.ac.th, <sup>2</sup>Tejtasin.phil@stou.ac.th

#### บทคัดย่อ

บทความนี้เสนอการตรวจจับรอยขีดข่วนบนพื้นผิวของวัตถุโลหะที่เป็นผลิตภัณฑ์มูลค่าสูงเพื่อทำการคัดชิ้นงานเสียออกก่อนส่งเข้าผลิต โดยการตรวจจับพื้นผิวที่มีรอยขีดข่วนใช้วิธีการประมวลผลภาพเพื่อตรวจจับรอยขีดข่วนด้วยวิธีการโครงข่ายประสาทแบบคอนโวลูชัน (CNN) อย่างไรก็ตาม การนำภาพพื้นผิวของวัตถุโลหะที่มีความละเอียดสูงและขนาดใหญ่ผ่านกระบวนการของโครงข่ายประสาทแบบคอนโวลูชันอาจทำให้รอยขีดข่วนบนภาพอินพุตหายและทำให้ไม่สามารถตรวจจับรอยขีดข่วนได้ เพื่อแก้ปัญหาดังกล่าว จึงทำการแบ่งภาพอินพุตออกเป็นส่วนย่อยก่อนนำแต่ละภาพย่อยเข้ากระบวนการของ CNN ตามลำดับ การฝึกสอนใช้ชุดข้อมูลรอยขีดข่วนจำนวน 2,600 ภาพ ผลการทดลองพบว่าวิธีการที่เสนอสามารถตรวจจับรอยขีดข่วนเพิ่มขึ้นประมาณร้อยละ 90

คำสำคัญ: การตรวจจับรอยขีดข่วน, โครงข่ายประสาทแบบคอนโวลูชัน, การแบ่งภาพ

#### Abstract

This paper proposes scratched surface detection of metal objects that are luxury products to reject them before producing the products. The scratched surface detection uses a processing method by the Convolution Neural Network (CNN)

method. However, surface images that are high resolution and big size to input the process of CNN may lose the small scratches and cannot detect that region. To solve this problem, an image subdivision method is applied for dividing the input image into the sub-images before inputting each sub-image to the CNN. The training data set of scratches is 2,600 images. The experimental results found that the proposed method can detect the scratched surfaces that increase by approximately 90 percent.

Keywords: Scratch Detection, Convolution Neural Network, Image Subdivision

#### 1. บทนำ

กระบวนการด้านการตรวจสอบคุณภาพของวัตถุดิบเป็นกระบวนการที่สำคัญในการผลิตสินค้า การผลิตกล่องผลิตภัณฑ์ของสินค้าแบรนด์เนมที่มีราคาสูง มีการตรวจสอบสินค้าอย่างเข้มงวดจากลูกค้าหรือองค์กรที่เป็นตัวแทนลูกค้า ฉะนั้นวัตถุดิบในการผลิตสินค้าที่มีส่วนประกอบเป็นโลหะอลูมิเนียม เป็นชิ้นส่วนที่พบว่าเกิดรอยขีดข่วนได้ง่าย ทำให้สินค้าไม่ผ่านมาตรฐาน และการตรวจสอบด้วยสายตาอาจทำให้เกิดความเมื่อยล้าส่งผลทำให้เกิดข้อผิดพลาดในการตรวจสอบได้ บทความนี้จึงเสนอวิธีการประมวลผลภาพด้วย

เทคนิค CNN ช่วยในกระบวนการตรวจสอบคุณภาพของสินค้า โดยเสนอการออกแบบและพัฒนาโมเดลด้วย YOLOv8 (You Only Look Once) เป็นปัญญาประดิษฐ์การเรียนรู้เชิงลึก (Deep Learning) ในการตรวจจับวัตถุแบบเรียลไทม์ และพัฒนาแอปพลิเคชันในการใช้งานการตรวจสอบรอยขีดข่วนบนผิวอลูมิเนียม โดยมีงานวิจัยที่เกี่ยวข้องในการตรวจจับวัตถุ 2 แนวทาง คือ CNN และการประมวลผลภาพ ดังนี้

แนวทางการตรวจจับวัตถุด้วย CNN ได้แก่งานวิจัยของ ขวัญกมล คีฎกัญจน์ และคณะ [1] ที่ได้ใช้ CNN ในการวินิจฉัยมะเร็งผิวหนังจากภาพถ่ายจากโทรศัพท์มือถือ มีการใช้เทคนิคปรับปรุงภาพ และเทคนิคการโอนถ่ายการเรียนรู้ และงานวิจัยของ กรณิการ์ นูลโพธิ์ [2] ที่ได้นำ CNN YOLOv5 มาทำการตรวจจับผลจากบนขวดน้ำดื่ม โดยทำงานร่วมกับหุ่นยนต์คั่ว โดยมีผลลัพธ์ค่าความแม่นยำ 97.23% งานวิจัยของ Nicha and Somkid [3] ที่ได้นำ CNN YOLOv5 และ R-CNN มาตรวจจับ ตรวจสอบระบุผ้าใบครกชนิดในแต่ละรุ่น เพื่อส่งมอบให้ลูกค้าได้ถูกต้อง ซึ่งผลการวิจัยพบว่า CNN YOLOv5 มีประสิทธิภาพดีกว่าทั้งในด้านประสิทธิภาพการตรวจจับและความเร็ว งานวิจัยของ Kietikul, et al. [4] กล่าวถึงการจำแนกผู้ไม่สวมหมวกนิรภัย ด้วย CNN Resnet-v2, Inception-v2, Inception-v3 ผลประเมินประสิทธิภาพของแบบจำลองพบว่า Resnet-v2 และ Inception-v3 มีประสิทธิภาพดีใกล้เคียงกันจึงได้นำ Resnet-v2 ตรวจสอบคนและรถจักรยานยนต์ และใช้ Inception-v3 ตรวจสอบการสวมใส่หมวกนิรภัย งานวิจัยของ พุฒิพงศ์ จันทร์แจ่ม [5] ได้กล่าวถึงการนำ CNN YOLOv4 มาใช้ในการตรวจสอบการรอคอยการได้รับบริการของลูกค้าในร้านอาหาร โดยการตรวจสอบภาพจากกล้องวงจรปิด และได้ใช้เทคนิคการโอนถ่ายความสามารถของโมเดล ทำให้ได้ค่าประเมินประสิทธิภาพสูงถึง 96% งานวิจัยของ ราตรี คำโหวง [6] ได้นำ CNN และ Faster R-CNN มาตรวจสอบผลสุกของมังคุดบนต้นซึ่ง Faster R-CNN มีประสิทธิภาพที่ดีกว่า

แนวทางการตรวจจับวัตถุด้วย การประมวลผลภาพ ได้แก่งานวิจัยของ วรุตม์ บุญเยี่ยม และ ศิริเรือง พัฒน์ช่วย [7] ได้นำการประมวลผลภาพมาตรวจตรวจสอบแบบปรนัย

ด้วยวิธีการฝนบนกระดาษ A4 ด้วยวิธีการแปลงภาพเป็นไบนารี การตัดภาพ การหาขอบภาพ งานวิจัยของมณีนุช นาคาทอง และ นิวัตร อังควิสิษฐพันธ์ [8] ได้นำวิธีการประมวลผลภาพมาตรวจสอบชิ้นส่วนอิเล็กทรอนิกส์ของโทรศัพท์มือถือที่มีการประกอบไม่สมบูรณ์ งานวิจัยของเชิด คงห้อย และนิคม สุวรรณวร [9] ได้นำวิธีการประมวลผลภาพมาตรวจจับการเคลื่อนไหวของลูกโป่งลมในกระเบื้องเพื่อหาค่าเฉลี่ยการเคลื่อนที่เพื่อประเมินความแข็งแรงของลูกโป่งลม งานวิจัยของ ตะวัน ขุนอาสา [10] ได้นำวิธีการประมวลผลภาพมาตรวจจับป้ายจราจรเพื่อช่วยเหลือผู้ขับขี่ยานพาหนะ โดยการใช้ไลบรารี Haar Like Feature detection ของ OpenCV มีความถูกต้อง 96.95 %

บทความนี้ได้แบ่งหัวข้อสำหรับการนำเสนอประกอบไปด้วย 4 ส่วน ได้แก่ ส่วนบทนำ ส่วนวิธีการพัฒนาโมเดลและวิธีการนำไปใช้งาน ส่วนวิเคราะห์ผลการทดลอง และส่วนสรุปผลการทดลองและข้อเสนอแนะ

## 2. วิธีการพัฒนาโมเดลและวิธีการนำไปใช้งาน

กระบวนการในการตรวจจับรอยขีดข่วนด้วย CNN ประกอบไปด้วย 5 ขั้นตอนดังต่อไปนี้

### 2.1 วิเคราะห์ปัญหาและศึกษาเทคโนโลยีที่นำมาแก้ไข

ผู้วิจัยพบว่าวิธีการประมวลผลภาพโดยใช้ OpenCV กับวิธีการ CNN และได้ทำการสร้างแบบทดสอบ พบว่า CNN ให้ผลลัพธ์ที่ดีที่สุดกับการตรวจจับรอยขีดข่วนบนผิวอลูมิเนียม ซึ่งมีการสะท้อนแสงและเงาสูง

### 2.2 รูปแบบและวิธีการศึกษา

ขอบเขตการออกแบบและพัฒนากระบวนการประกอบด้วย (ก) พัฒนาโมเดลด้วยอัลกอริทึม CNN โดยใช้โมเดล YOLOv8 เป็นแม่แบบ (ข) เตรียมข้อมูลสำหรับการฝึกสอนโมเดล (ค) ฝึกสอนโมเดล (ง) วัดประสิทธิภาพโมเดล และ (จ) สร้างส่วนประสานผู้ใช้ GUI

### 2.3 ขั้นตอนการเก็บรวบรวมข้อมูล และการสร้าง Dataset

ในขั้นตอนนี้ประกอบด้วย (ก) เตรียมข้อมูลที่นำเข้ามาสำหรับการฝึกสอนโมเดล โดยการสร้างพื้นที่ควบคุมแสงเพื่อถ่ายภาพรอยขีดข่วนตามลักษณะที่กำหนดด้วยขนาดภาพขนาด

640x640 pixel (ข) ทำการกำหนดประเภทของวัตถุที่ต้องการตรวจจับที่ <https://roboflow.com/> (ค) ทำการเพิ่มภาพแบบ Image Augmentation เพื่อเพิ่มรูปภาพหรือปรับเปลี่ยนรูปแบบของภาพเพื่อให้ภาพมีความหลากหลายรูปแบบมากขึ้นส่งผลให้การเรียนรู้ของโมเดลมีประสิทธิภาพการทำนายได้ดีขึ้นและลดปัญหาการเกิด Overfitting (ง) ใช้จำนวนชุดข้อมูลฝึกสอนจำนวน 2,600 ภาพและสำหรับการตรวจสอบ 246 ภาพ และสำหรับทดสอบ 122 ภาพ

**2.4 ขั้นตอนการสร้างโมเดล การตรวจจับวัตถุ**

ในขั้นตอนนี้ได้ทำการฝึกสอน โมเดลการตรวจจับวัตถุทั้ง 3 แบบคือ YOLOv8s, YOLOv8n, YOLOv8x โดยใช้ YOLOv8n, YOLOv8x เป็นตัวเปรียบเทียบในการประเมินประสิทธิภาพเพื่อให้ได้โมเดลที่เหมาะสมในการนำไปใช้งานฝึกสอน โมเดลด้วย Google Colab ที่ใช้ทรัพยากรฮาร์ดแวร์ GPU T4 และ GPU Memory 16GB โดยการตั้งค่าพารามิเตอร์สำหรับฝึกสอน ได้แก่ number of class = 1, epochs = 300, imgsz = 640, batch = 16, patience = 50, optimizer = auto, lr0 = 0.001 และ lrf = 0.001

**2.5 การนำโมเดลไปใช้งาน**

ขั้นตอนการนำโมเดลไปใช้งานประกอบด้วย (ก) เมื่อฝึกฝนโมเดลเสร็จ ทำการโหลดโมเดลที่ได้ฝึกฝนบน Google Colab ชื่อไฟล์ best.pt และติดตั้ง YOLOv8 (ข) ออกแบบและพัฒนาแอปพลิเคชันด้วย Python และ PyQt5 โดยมีการออกแบบระบบแสดงดังภาพที่ 1

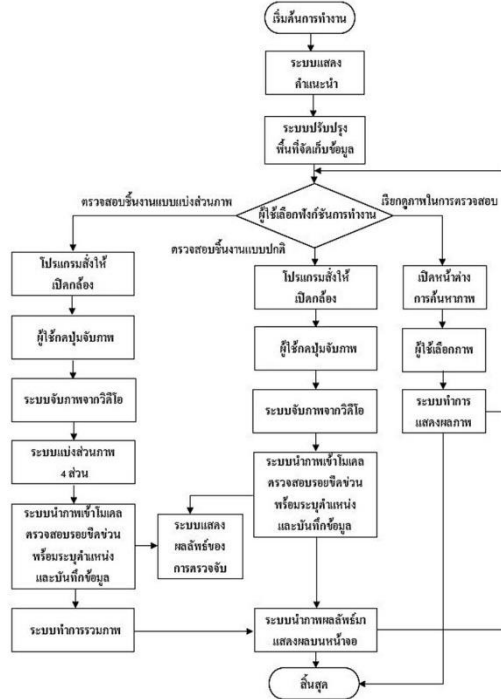
**3. ผลการทดลอง**

การทดลองประกอบด้วยรายละเอียดดังนี้

**3.1 การเปรียบเทียบประสิทธิภาพของโมเดล CNN**

การทดลองได้ทำการเปรียบเทียบประสิทธิภาพของโมเดล CNN แต่ละแบบ ได้แก่แบบ YOLOv8n, YOLOv8s และ YOLOv8x ด้วยค่า Mean Average Precision (mAP50) ที่มีการกำหนดค่า IoU = 0.5 พบว่าค่าผลการประเมินมีความใกล้เคียงกัน ในโมเดลทั้ง 3 แบบ แต่ใช้เวลาในการฝึกสอนแตกต่างกันมาก ดังผลการทดลองในตารางที่ 1

การประชุมวิชาการระดับชาติและนานาชาติด้านเทคโนโลยีคอมพิวเตอร์และระบบสารสนเทศประยุกต์ (ACTIS) และการประชุมวิชาการระดับชาติและนานาชาติด้านบริหารธุรกิจ (NCOBA) ครั้งที่ 20, 2566



ภาพที่ 1 : แผนผังการทำงานของระบบการตรวจจับรอยขีดข่วนบนผิวอูมิเนียม

ตารางที่ 1: เปรียบเทียบประสิทธิภาพโมเดล CNN แต่ละแบบ

	YOLOV8N	YOLOV8S	YOLOV8X
PRECISION	91.9	89.5	89.3
RECALL	81	87.7	87.3
MAP50	91.6	92.4	92.6
MAP50-90	58.9	59.5	61.5
LEANING TIME	2.5 Hour	2.8 Hour	10.8 Hour

ผลการเปรียบเทียบประสิทธิภาพการตรวจจับด้วยการนับจำนวนรอยขีดข่วนที่ตรวจจับได้ และค่าเวลาที่ใช้ในการตรวจจับ ค่า FN (False Negative) คือค่าจำนวนที่ไม่สามารถตรวจจับรอยขีดข่วนได้ และค่า T/A (Time Average) คือค่าของเวลาเฉลี่ยที่ใช้ในการประมวลผลของทุกภาพ ดังตารางที่ 2

**3.2 การเปรียบเทียบผลลัพธ์การตรวจจับรอยขีดข่วนด้วยเทคนิคการแบ่งภาพ และเทคนิค CNN**

การทดลองนี้ใช้เทคนิคการแบ่งภาพที่มีความละเอียดสูงก่อนนำเข้าโมเดล CNN โดยเทคนิคนี้ ภาพอินพุตไม่ถูกลดความละเอียดลงมากจนทำให้คุณลักษณะสำคัญของวัตถุสูญหายหรือไม่ชัดเจน ภาพอินพุตขนาด 3,000x4,000 pixel ถูกแบ่งให้เป็นภาพย่อย 4 ส่วนแต่ละภาพย่อยมีขนาด 1,500x2,000 pixel และผลการทดลองพบว่า วิธีที่เสนอสามารถตรวจสอบรอยขีดข่วนได้ดีกว่าที่ไม่ได้ใช้เทคนิคการแบ่งภาพอินพุตก่อนนำเข้าโมเดล CNN โดยจากภาพทั้งหมด 7 ภาพมีรอยรวมทั้งหมด 48 รอย วิธีเทคนิค CNN อย่างเดียวตรวจไม่พบ 11 รอย (FN=11) และเมื่อเพิ่มกระบวนการการแบ่งภาพอินพุตทำให้สามารถตรวจพบรอยได้ 10 รอยจาก 11 รอยหรือตรวจพบเพิ่มขึ้นร้อยละ 90.9 ดังตารางที่ 3 ที่แสดงตำแหน่งของจุดที่ตรวจไม่พบ (FN) 1 จุดบนภาพอินพุตภาพที่หนึ่ง

**ตารางที่ 2:** ผลลัพธ์การตรวจจับรอยขีดข่วนและเวลาเฉลี่ยที่ใช้

Image size (pixel)	Total of defect	YOLOv8n		YOLOv8s		YOLOv8x	
		detect	time	detect	time	detect	time
10 Image 640 x 640	29	4	215ms	2	294.2ms	1	2132.8ms
10 Image 1200 x 1200	54	0	512ms	1	1041.2ms	1	4823ms
Summary	83	FN=4 4.82%	T/A 335.3ms	FN=3 3.6%	T/A 679.3ms	FN=2 2.4%	T/A 3272.5ms

**ตารางที่ 3:** ผลลัพธ์การตรวจพบรอยเพิ่มขึ้นด้วยวิธีที่เสนอ

Scratches = 48 Image = 7	FN= 11	FN=1 , FP=3
Master image size = 3000x4000 pixel	Image size = 3000x4000 pixel	Split = 4 part Image size per pice = 1500x2000 pixel
Master	Predict non subdivision	Predict with subdivision

**3.3 การแสดงหน้าต่างแอปพลิเคชันของการตรวจ**

หน้าต่างแสดงผลลัพธ์การตรวจจับรอยขีดข่วนพัฒนาด้วย Python และ PyQT5 โดยหากตรวจพบรอยขีดข่วน

จะแสดงหน้าต่างผลลัพธ์ดังภาพที่ 2 และหากตรวจไม่พบรอยขีดข่วนบนชิ้นงานจะแสดงหน้าต่างผลลัพธ์ดังภาพที่ 3



**ภาพที่ 2 :** ผลลัพธ์การตรวจสอบชิ้นงานที่มีรอยขีดข่วน



**ภาพที่ 3 :** ผลลัพธ์การตรวจสอบชิ้นงานที่ไม่มีรอยขีดข่วน

**4. สรุปผลการทดลอง และข้อเสนอแนะ**

จากผลการทดลองสามารถสรุป และมีข้อเสนอแนะดังนี้

**4.1 สรุปผลการทดลอง**

จากการทดลองของวิธีการที่นำเสนอและการวัดประสิทธิภาพของแต่ละรูปแบบโมเดลด้วยข้อมูลการฝึกสอนจำนวน 2,600 ภาพ โดยโมเดลทั้ง 3 แบบมีการออกแบบสถาปัตยกรรมที่มีความซับซ้อนแตกต่างกัน ผลลัพธ์ในการประเมินค่าประสิทธิภาพ mean Average Precision (mAP50) เท่ากับ 0.916, 0.924, 0.926 ตามลำดับ ซึ่ง YOLOv8x มีค่า mAP50 สูงที่สุดแต่ใช้เวลาในการทำนายผลมากที่สุด ส่วนโมเดล YOLOv8s ได้ให้ค่า mean Average Precision เท่ากับ

0.924 ที่ใกล้เคียงกับโมเดล YOLOv8x แต่ใช้เวลาเฉลี่ยในการตรวจจับรอยขีดข่วนลดลงถึง 79.24% และให้ค่าในการตรวจจับลดลงเพียง 1.2% จากผลการทดสอบโดยใช้ภาพถ่ายทดสอบจำนวน 20 ภาพ โดยมีความละเอียด 640x640 pixel จำนวน 10 ภาพ และความละเอียด 1200x1200 pixel จำนวน 10 ภาพ โดยมีรอยขีดข่วนรวมทั้งหมด 83 รอย และทดสอบด้วยโมเดล YOLOv8s พบว่ามีรอยขีดข่วนที่ตรวจจับไม่ได้อยู่จำนวน 3 รอย คิดเป็นเปอร์เซ็นต์เท่ากับ 3.6%

สำหรับการใช้เทคนิคการแบ่งภาพอินพุตที่มีความละเอียดสูงเป็นเทคนิคที่ให้ผลลัพธ์ในการตรวจจับรอยขีดข่วนได้เพิ่มขึ้น จากผลการทดสอบที่เปรียบเทียบกับโมเดลที่ไม่ได้ใช้เทคนิคการแบ่งภาพจะตรวจไม่พบรอยขีดข่วนจำนวน 11 รอย เนื่องจากเป็นรอยขีดข่วนที่มีขนาดเล็กและไม่ชัดเจน แต่เมื่อใช้เทคนิคการแบ่งภาพก่อนนำเข้าโมเดล CNN ทำให้สามารถตรวจพบรอยเพิ่มขึ้นอีก 10 รอยจากทั้งหมด 11 รอย คิดเป็นร้อยละ 90.9% และรอยที่ตรวจไม่พบ 1 รอย เป็นรอยขีดข่วนที่มีขนาดเล็กมากซึ่งอาจต้องทำการแบ่งภาพให้เป็นภาพย่อยลงอีกแต่ต้องไม่เล็กไปกว่าความละเอียดของภาพที่โมเดลต้องการ

#### 4.2 ข้อเสนอแนะ

ข้อเสนอแนะจากผลการทดลองได้แก่ 1) บทความนี้ได้ทำการทดลองกับโมเดล YOLOv8 ใน 3 ระดับความซับซ้อนของสถาปัตยกรรมที่แตกต่างกันได้แก่ YOLOv8n, YOLOv8s และ YOLOv8x ที่มีเทคนิคการตรวจจับเป็นแบบ One stage object detection ซึ่งยังไม่ได้ทำการทดสอบกับอัลกอริทึมที่เป็นแบบ Two stage object detection เช่น Faster R-CNN เป็นต้น 2) พื้นที่ตรวจจับรอยขีดข่วนควรอยู่ในพื้นที่ควบคุมแสงสว่าง เนื่องจากแสงสว่างมีผลกับการทำงานของโมเดลที่ทำให้เกิดความคลาดเคลื่อนของผลลัพธ์ในการทำงานของโมเดล 3) กลุ่มอุตสาหกรรมผลิตชิ้นส่วนยานยนต์ หรือกลุ่มอุตสาหกรรมอื่นๆ ที่รอยขีดข่วนส่งผลกระทบต่อคุณภาพสินค้า สามารถนำโมเดลไปใช้งานในการตรวจสอบรอยขีดข่วนกับวัสดุที่มีลักษณะพื้นผิวที่ใกล้เคียงกับพื้นผิวอลูมิเนียมได้

#### เอกสารอ้างอิง

- [1] ขวัญกมล คิฐักัญจน์ และคณะ, “วินิจฉัยมะเร็งผิวหนังจากภาพถ่ายโทรศัพท์ด้วยโครงข่ายประสาทเทียมสั่งวัฒนาการ”, มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต, 2565.
- [2] กรณิการ์ มูลโพธิ์, “หุ่นยนต์คู่เท้าคัดแยกถั่วโดยใช้อัลกอริทึม Yolo และบรรจุก้อนในกระบวนการผลิตน้ำดื่ม”, มหาวิทยาลัยเทคโนโลยีมหานคร, 2565.
- [3] Nicha, et al., “Deep Learning and Image Processing for Disc Brake Pad Identification: A Case Study of Brake Pads Company”, Mahidol University, 2022.
- [4] Kietikul, et al., “Classifying Non-helmeted Motorcyclists by Using Convolutional Neural Networks”, King Mongkut's Institute of Technology Ladkrabang, 2022.
- [5] พุฒิพงษ์ จันทรแจ่ม, “การปรับปรุงประสิทธิภาพของกระบวนการบริการลูกค้าโดยใช้การวิเคราะห์จากกล้องวงจรปิด” สถาบันบัณฑิตพัฒนบริหารศาสตร์, 2563.
- [6] ราตรี คำโหวง, “การตรวจจับมัจจุสดูด้วยการเรียนรู้เชิงลึก”, มหาวิทยาลัยราชภัฏอุดรดิตต์, 2565.
- [7] วรุตม์ บุญเยี่ยม และ ศิริเรือง พัฒนาช่วย, “ระบบตรวจข้อสอบปรนัยแบบวิธีการฝนด้วยหลักการประมวลผลภาพ”, มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์, 2564.
- [8] มณีนุช นาคทอง และ นิวัฒน์ อังควิทยูพันธ์, “การจำแนกชิ้นส่วนโทรศัพท์มือถือด้วยเทคนิคการประมวลผลภาพ”, มหาวิทยาลัยมหาสารคาม, 2563.
- [9] เชิด คงห้อย และ นิคม สุวรรณวร, “เทคนิคการคัดเลือกคุณภาพของตัวอ่อนปูแสมด้วยวิธีการตรวจจับการ”, มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย, 2563.
- [10] ตะวัน ขุนอาสา, “ระบบตรวจจับและคัดแยกป้ายจราจรฐานข้อมูล”, มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ, 2558.

## ประวัติผู้ศึกษา

ชื่อสกุล	นายณัฏฐ์ มละสาร
วัน เดือน ปี เกิด	29 กุมภาพันธ์ 2523
สถานที่เกิด	อำเภอวัฒนานคร จังหวัดสระแก้ว
ที่อยู่ปัจจุบัน	วิทยาศาสตร์บัณฑิต (เทคโนโลยีพิวเตอร์) มหาวิทยาลัยเทคโนโลยีราชมงคล ธัญบุรี 2554
ประวัติการศึกษา	บริษัทไพรม์บ็อกซ์ เอ็มเอฟจี จำกัด จังหวัดสมุทรปราการ
ประวัติการทำงาน	IT SUPERVISOR

