การประยุกต์ใช้ระบบปัญญาประดิษฐ์ในการตรวจจับลักษณะข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์

นายณัฐชัย รุ่งเชษฐ์

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

การประยุกต์ใช้ระบบปัญญาประดิษฐ์ในการตรวจจับลักษณะข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์

นายณัฐชัย รุ่งเชษฐ์

การค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตร
วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมการจัดการ
บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปีการศึกษา 2567

ใบรับรองโครงร่างการค้นคว้าอิสระ

บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

เรื่อง การประยุกต์ใช้ระบบปัญญาประดิษฐ์ในการตรวจจับลักษณะข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์

โดย นายณัฐชัย รุ่งเชษฐ์

ได้รับอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมการจัดการ

คณบดีบัณฑิตวิทยาลัย / หัวหน้าภาควิชา

คณะกรรมการสอบการค้นคว้าอิสระ

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.กฤษฎากร บุดดาจันทร์)

..... อาจารย์ที่ปรึกษา

(อาจารย์ ดร.นฤมล สัมฤทธิ์)

..... กรรมการภายนอก

(รองศาสตราจารย์ ดร.สถาพร อมรสวัสดิ์วัฒนา)

ชื่อ : นายณัฐชัย รุ่งเชษฐ์
ชื่อการค้นคว้าอิสระ : การประยุกต์ใช้ระบบปัญญาประดิษฐ์ในการตรวจจับลักษณะข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์
สาขาวิชา : วิศวกรรมการจัดการ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก : อาจารย์ ดร.นฤมล สัมฤทธิ์
ปีการศึกษา : 2567

บทคัดย่อ

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาระบบปัญญาประดิษฐ์ในการตรวจจับข้อบกพร่องด้านความสมบูรณ์ของตัวอักขระบนชิ้นส่วนอิเล็กทรอนิกส์ประเภท 16-position rotary switch ซึ่งเป็นปัญหาที่มีผลต่อคุณภาพและความน่าเชื่อถือในอุตสาหกรรม Electronics Manufacturing Services (EMS) โดยเฉพาะในกรณีศึกษาที่แม้จะมีการตรวจสอบแบบ 100% โดยพนักงานแล้ว ยังพบการหลุดรอดของงานเสียไปยังลูกค้า การใช้ระบบปัญญาประดิษฐ์จึงเป็นทางเลือกที่เหมาะสมในการลดข้อผิดพลาดของมนุษย์และยกระดับการควบคุมคุณภาพ

งานวิจัยนี้ใช้โมเดล YOLOv11 ในการตรวจจับข้อบกพร่อง โดยทดลองทั้งรุ่น Yolov11n และ Yolov11s ภายใต้เงื่อนไขการฝึกสอนแบบแบ่งคลาส 1 คลาส และ 18 คลาส รวมถึงจำนวนรอบการฝึกสอนที่แตกต่างกัน ผลการทดลองพบว่า โมเดล Yolov11s แบบ 1 คลาส ฝึกสอน 200 รอบ ให้ผลลัพธ์ดีที่สุด โดยได้ค่า Precision 97.5%, Recall 98.2%, mAP50 98.3% และเมื่อทดสอบกับภาพจำลองงานเสีย 32 ภาพ พบว่ามี ค่า Accuracy สูงถึง 93.8% รองลงมาคือ Yolov11n แบบ 1 คลาส Accuracy 87.5% ในขณะที่การแบ่งคลาสแบบ 18 คลาสให้ผลลัพธ์ต่ำกว่าอย่างชัดเจน โดย Yolov11n และ Yolov11s มี Accuracy เพียง 46.9% และ 37.5% ตามลำดับ

(มีจำนวนทั้งสิ้น 60 หน้า)

คำสำคัญ : ปัญญาประดิษฐ์, YOLOv11, การตรวจจับข้อบกพร่อง, อิเล็กทรอนิกส์

_____ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลัก

Name : Mr. Nattachai Rungchate
Independent Study Title : Applying an artificial intelligence to detect the defect on electronics component
Major Field : Engineering Management
King Mongkut's University of Technology North Bangkok
Independent Study Advisor : Dr. Naruemon Sumrith
Academic Year : 2024

ABSTRACT

This independent study aimed to develop an artificial intelligence (AI) system for detecting character integrity defects on 16-position rotary switches—components critical to the quality of products in the competitive Electronics Manufacturing Services (EMS) industry. Despite 100% human visual inspection, defective parts were still delivered to customers. Therefore, AI-based detection is explored as a promising alternative to minimize human error and enhance quality assurance.

The study applied YOLOv11 object detection models (Yolov11n and Yolov11s) with different training conditions, including single-class and 18-class annotations and multiple training epochs. The results showed that the Yolov11s model with single-class annotation trained for 200 epochs performed best, achieving a Precision of 97.5%, Recall of 98.2%, and mAP50 of 98.3%. When tested with 32 simulated defective images, this model reached an Accuracy of 93.8%. The Yolov11n model with the same conditions achieved a slightly lower Accuracy of 87.5%. In contrast, the 18-class annotation produced lower performance across both models, with Accuracy of only 46.9% for Yolov11n and 37.5% for Yolov11s.

(Total 60 Pages)

Keywords: Artificial Intelligence, YOLOv11, Defect Detection, Electronics

_____ Advisor

กิตติกรรมประกาศ

การค้นคว้าอิสระฉบับนี้สามารถดำเนินการจนสำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความกรุณาและการสนับสนุนจากหลายท่านที่มีส่วนสำคัญในการให้คำแนะนำ แบ่งปันความรู้ และอำนวยความสะดวกในด้านต่างๆ ตลอดระยะเวลาของการศึกษา

ข้าพเจ้าขอกราบขอบพระคุณ ดร.นฤมล สัมฤทธิ์ อาจารย์ที่ปรึกษาการค้นคว้าอิสระหลักที่ได้กรุณาให้คำปรึกษา แนะนำ และเสนอแนะแนวทางในการวิจัยอย่างใกล้ชิดตลอดระยะเวลาการดำเนินงาน อีกทั้งยังเป็นแรงสนับสนุนที่สำคัญให้การค้นคว้าอิสระนี้สำเร็จตามเป้าหมาย

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.กฤษฎากร บุดดาจันทร์ ประธานกรรมการสอบ และ รองศาสตราจารย์ ดร.นิพนธ์ ภูวเกียรติกำจร กรรมการสอบ ที่ได้ให้คำแนะนำในการปรับปรุงและพัฒนาผลงานวิจัยให้มีความสมบูรณ์มากยิ่งขึ้น รวมถึงแนวทางการต่อยอดในการประยุกต์ใช้ความรู้ในอนาคต

ข้าพเจ้าขอแสดงความขอบคุณอย่างยิ่งต่อ ผู้ช่วยศาสตราจารย์ ดร.สวนิต อิชยาวณิชย์ ที่ได้ให้คำแนะนำด้านระเบียบวิธีวิจัยอย่างละเอียดรอบคอบ อันเป็นแนวทางสำคัญในการกำหนดโครงสร้างและวางแผนการวิจัยให้ดำเนินไปอย่างถูกต้องและมีประสิทธิภาพ

ขอขอบพระคุณ บริษัท แอลไพน์ เทคโนโลยี แมนูแฟคเจอริ่ง (ประเทศไทย) จำกัด ที่ให้ความอนุเคราะห์ในการเก็บรวบรวมข้อมูล สนับสนุนอุปกรณ์ และจัดสรรสถานที่เพื่อการทดลองและถ่ายภาพสำหรับการฝึกสอนปัญญาประดิษฐ์ ซึ่งเป็นองค์ประกอบสำคัญของการศึกษาครั้งนี้

นอกจากนี้ ข้าพเจ้ายังขอขอบคุณครอบครัว เพื่อนร่วมงาน และเพื่อนร่วมรุ่นในระดับบัณฑิตศึกษาทุกท่าน ที่มอบกำลังใจและความช่วยเหลืออย่างจริงใจตลอดระยะเวลาการศึกษา

ณัฐชัย รุ่งเชษฐ์

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูปภาพ

บทที่ 1
บทนำ

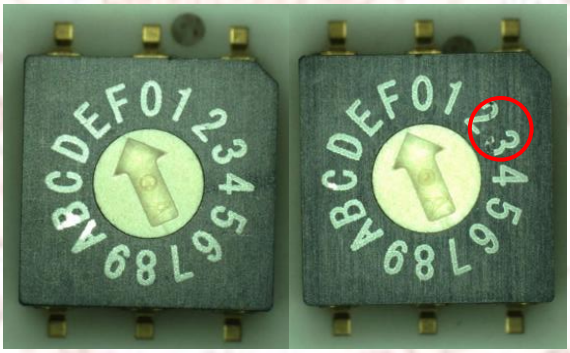
1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันธุรกิจ Electronics Manufacturing Services (EMS) ในประเทศไทยนั้นมีผู้ประกอบการอยู่เป็นจำนวนมาก ซึ่งมีทั้งผู้ประกอบการเดิมและผู้ประกอบการใหม่อันเป็นกลุ่มทุนจากประเทศจีนที่ต้องการใช้ประเทศไทยเป็นฐานการผลิตเพื่อหลบเลี่ยงมาตรการกีดกันทางการค้าของทางสหรัฐอเมริกา (รศ.ดร.จุฑาทิพย์, 2566) ลูกค้าของธุรกิจ EMS จึงมีทางเลือกในการคัดเลือกบริษัทผู้ส่งมอบให้มาเป็นผู้ประกอบแผงวงจรอิเล็กทรอนิกส์ของสินค้าของตนเองมากขึ้น ทำให้เกิดการแข่งขันที่มีความเข้มข้นสูงมากในธุรกิจนี้ โดยปัจจัยหนึ่งที่มีส่วนสำคัญในการตัดสินใจเลือกผู้ส่งมอบก็คือคุณภาพของการประกอบ ดังนั้นหากไม่สามารถรักษามาตรฐานในด้านคุณภาพของการประกอบได้ก็อาจทำให้ผู้ประกอบการแข่งขันในธุรกิจนี้ได้ยากลำบากมากยิ่งขึ้น

โรงงานที่ผู้วิจัยทำการศึกษาในสารนิพนธ์ฉบับนี้เป็นหนึ่งในผู้ประกอบการในธุรกิจ EMS สัญชาติญี่ปุ่นที่ดำเนินธุรกิจอยู่ในประเทศไทยมาเป็นระยะเวลายาวนานกว่า 30 ปี และมีลูกค้าหลักเป็นบริษัทสัญชาติญี่ปุ่นที่อยู่ทั้งภายในประเทศและต่างประเทศ ซึ่งแผงวงจรอิเล็กทรอนิกส์ที่รับประกอบนั้นถูกใช้อยู่ในเครื่องใช้ไฟฟ้าประเภทต่างๆ ได้แก่ เครื่องปริ้นเตอร์, เครื่องปรับอากาศ, เครื่องอัลตราซาวด์, โมดูลเซ็นเซอร์ต่าง และเครื่องเสียงรถยนต์ เป็นต้น โดยมีกำลังการผลิตเกินกว่า 10,000 ชิ้นต่อวัน ทั้งนี้ขึ้นอยู่กับขนาดและความยากง่ายของการประกอบแผงวงจร

ปัญหาที่หยิบยกมาทำการศึกษาในครั้งนี้คือการหลุดรอดของงานเสียไปยังลูกค้ารายหนึ่ง โดยมีลักษณะข้อบกพร่องของของเสียนั้นในด้านความสมบูรณ์ของอักขระ (Character) บนชิ้นส่วนอิเล็กทรอนิกส์ประเภท 16-position rotary switch ซึ่งข้อบกพร่องที่เกิดขึ้นนี้เกิดจากกระบวนการของผู้ผลิตชิ้นส่วนอิเล็กทรอนิกส์รายหนึ่ง และแม้ว่าทางผู้ผลิตชิ้นส่วนรายนี้จะแจ้งว่าได้ทำการปรับปรุงกระบวนการผลิตของตนแล้ว แต่ทางบริษัทยังตรวจพบปัญหานี้ที่กระบวนการประกอบอยู่เป็นครั้งคราว ทางโรงงานจึงจำเป็นต้องเพิ่มหัวข้อปัญหาข้อบกพร่องนี้กระบวนการตรวจสอบขั้นสุดท้าย ด้วยวิธีการตรวจสอบด้วยสายตา (Visual Inspection) ผ่านโคมไฟเลนส์ขยาย (Magnify Lamp) โดยพนักงานฝ่ายผลิตที่มีทักษะในการตรวจสอบและเป็นการตรวจสอบแบบ 100 เปอร์เซ็นต์

อย่างไรก็ดีก็ยังมีโอกาสที่ชิ้นงานที่มีข้อบกพร่องจะหลุดรอดการตรวจสอบของพนักงานไปได้ จากสาเหตุดังต่อไปนี้ ความเหนื่อยล้าของพนักงานจากการทำงานเป็นระยะเวลายาวนาน, ความเร่ง-รีบของการผลิตเพื่อให้ทันต่อการส่งมอบ, สภาวะทางอารมณ์หรือความมั่นคงทางจิตใจของผู้ทำการตรวจสอบชิ้นงานในขณะทำการตรวจสอบ ซึ่งการหลุดรอดของชิ้นงานนี้จะส่งผลทำให้บริษัทต้องเสียค่าใช้จ่ายในการส่งคืนชิ้นงานจากประเทศญี่ปุ่นกลับมายังประเทศไทยเพื่อทำการคัดแยกชิ้นงานดีและงานเสีย ต้องใช้ชั่วโมงแรงงานของพนักงานในการตรวจสอบซึ่งรวมถึงการทำงานนอกเวลา ทำให้ต้นทุนต่อหน่วยของชิ้นงานสูงขึ้น ส่งผลให้ยอดขายและกำไรโดยภาพรวมลดลงในบางไตรมาส ดังนั้นเพื่อเป็นการป้องกันการหลุดรอดของของเสียที่มีลักษณะข้อบกพร่องนี้ทางโรงงานจึงจำเป็นต้องดำเนินการหาวิธีการทางเลือกอื่นๆ มาใช้ในการตรวจจับข้อบกพร่องดังกล่าว ซึ่งมีข้อพิจารณาด้านต้นทุนของเครื่องมือหรืออุปกรณ์ที่นำมาใช้ในการตรวจจับปัญหานี้เข้ามาเกี่ยวข้องด้วย ทั้งนี้ระบบ-วิชชั่น (Vision System) ที่ใช้ในการตรวจจับในอุตสาหกรรมมีต้นทุนอยู่ที่ราว 3 – 4 แสนบาท ซึ่งนับเป็นมูลค่าที่สูงในการลงทุน เนื่องจากการลงทุนดังกล่าวไม่สามารถคิดเพิ่มในราคาค่าประกอบได้



ภาพที่ 1-1 ภาพตัวอย่างชิ้นส่วนที่ดีและส่วนชิ้นที่มีข้อบกพร่อง

ลักษณะของงานดี คือจะต้องสามารถมองเห็นเป็นตัวอักขระอย่างชัดเจน ไม่มีส่วนใดส่วนหนึ่งของตัวอักขระขาด, แหว่งหรือเลือนลางจนไม่สามารถอ่านได้ว่าเป็นอักขระใด ดังเช่นภาพตัวอย่างชิ้นส่วนทางด้านซ้ายในภาพที่ 1-1 ส่วนลักษณะของงานเสีย คือมีส่วนใดส่วนหนึ่งของตัวอักขระขาด, แหว่งหรือเลือนลางจนไม่สามารถเห็นได้ชัดเจนว่าเป็นอักขระใดดังเช่นเลข 3 ของภาพที่ 1-1 ของชิ้นส่วนทางด้านขวามือ

แนวคิดหนึ่งที่ถูกนำมาพิจารณาคือการประยุกต์ใช้ระบบปัญญาประดิษฐ์ (Artificial Intelligence : AI) เนื่องจากระบบปัญญาประดิษฐ์ในปัจจุบันนี้ได้ถูกนำมาใช้อย่างแพร่หลายในด้านต่างๆ รวมถึงในระบบการมองเห็น (Vision System) ซึ่งความสามารถแยกแยะความแตกต่างของวัตถุประเภทต่างๆ นั้นสามารถประยุกต์ใช้ในงานอุตสาหกรรมแทนการตรวจสอบโดยมนุษย์ได้ ซึ่งความสามารถนั้นเกิดขึ้นได้ผ่านการที่ผู้ใช้งานป้อนชุดข้อมูล (Data Set) เข้าไปในระบบการเรียนรู้ของเครื่อง (Machine Learning) อันมีโครงข่ายระบบประสาทเทียม (Artificial Neural Networks - ANN) จำลองการทำงานของสมองมนุษย์เพื่อเรียนรู้จากชุดข้อมูลภาพของวัตถุประเภทต่างๆ เป็นจำนวนมาก และการเรียนรู้นั้นเป็นการเรียนรู้ในเชิงลึก (Deep Learning) ซึ่งมีความซับซ้อนส่งผลให้เกิดการสั่งสมเป็นฐานความรู้ (Knowledge base) และเป็นการเลียนแบบการแยกแยะและจดจำลักษณะของวัตถุของมนุษย์ (ยุวเรศมคฐ์, 2564)ดังนั้นจึงมีความเป็นไปได้ที่จะนำระบบปัญญาประดิษฐ์มาประยุกต์ใช้ในการตรวจจับข้อบกพร่องของข้อบกพร่องดังที่กล่าวมาข้างต้นนี้ได้ และนอกจากนี้ปัญญาประดิษฐ์นั้นยังมีข้อดีที่สามารถชดเชยส่วนที่เป็นข้อด้อยด้านต่างๆ ในการทำงานของมนุษย์ได้อีกด้วย เช่น ความเหนื่อยล้า,ความเครียดหรือบีบคั้นในการทำงาน, สภาวะทางอารมณ์ หรือการถูกรบกวนทางจิตใจ ซึ่งปัจจัยเหล่านี้ล้วนแล้วแต่มีความเป็นไปได้ที่จะส่งผลต่อความสามารถในการตัดสินใจของมนุษย์ การตรวจสอบชิ้นงานโดยปัญญาประดิษฐ์ที่เป็นการทำงานโดยเครื่องจักรจึงให้ผลลัพธ์ที่มีความถูกต้องและแม่นยำมากกว่ามนุษย์ (Li et al., 2023)

จากข้อมูลข้างต้น ผู้วิจัยจึงสนใจที่จะศึกษาการนำระบบปัญญาประดิษฐ์มาประยุกต์ใช้ในการตรวจจับข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์แทนการใช้พนักงานในการตรวจสอบด้วยสายตา เพื่อลดโอกาสการในการหลุดรอดซ้ำของของเสียไปยังลูกค้า โดยใช้ต้นทุนต่ำกว่าระบบวิชชั่นโดยทั่วไป เพื่อเป็นการยกระดับการรับประกันคุณภาพของการประกอบแผงวงจรอิเล็กทรอนิกส์ให้ดียิ่งขึ้น และนอกจากนี้ยังใช้ต่อยอดในการสร้างแบบจำลองปัญญาประดิษฐ์เพื่อใช้ในการตรวจจับข้อบกพร่องลักษณะอื่นๆ อนาคตได้อีกด้วย

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาปัญญาประดิษฐ์ในการตรวจจับข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์ ด้านความสมบูรณ์ของตัวอักขระ

1.2.2 เพื่อทราบว่าการแบ่งจำนวนคลาส, จำนวนครั้งการฝึกสอน, และโมเดลที่ใช้ฝึกสอนที่ต่างกันมีผลต่อประสิทธิภาพในการตรวจจับหรือไม่ และหากมีผล จะมีผลอย่างไร

1.3 ขอบเขตการศึกษา

ศึกษาและพัฒนาแบบจำลองปัญญาประดิษฐ์ YOLO v11 ในการตรวจจับข้อบกพร่องของอักขระบนชิ้นส่วนอิเล็กทรอนิกส์ประเภท 16-position rotary switch โดยแบ่งได้เป็นหัวข้อดังต่อไปนี้

1.3.1 การจำลองภาพงานเสียเพื่อใช้เป็นชุดข้อมูล (Dataset) ในการฝึกสอนโมเดลปัญญาประดิษฐ์

1.3.2 การสร้างป้ายกำกับ (Label) หรือ คำอธิบาย (Annotation) ด้วยเว็บไซต์ Roboflow โดยแบ่งคลาส (Class) ของป้ายกำกับ ออกเป็น 2 รูปแบบ คือ 1 คลาส และ 18 คลาส

1.3.3 การฝึกสอนปัญญาประดิษฐ์ด้วยแบบจำลองรุ่นต่างๆ ได้แก่ Yolov11n, Yolov11s และเงื่อนไขต่างๆ ได้แก่จำนวนครั้งในการฝึกสอน 50, 100, 150, 200

1.3.4 เปรียบเทียบประสิทธิภาพของแบบจำลองแต่ละรุ่น และแต่ละเงื่อนไขโดยประเมินดังต่อไปนี้

1.3.4.1 ค่าPrecision จากการฝึกสอน

1.3.4.2 ค่าRecall จากการฝึกสอน

1.3.4.3 ค่าmAP50 จากการฝึกสอน

1.3.4.4 ความแม่นยำ (Accuracy) จากการทดสอบ

1.3.5 การพัฒนาส่วนติดต่อผู้ใช้งาน (User Interface) ด้วยภาษา python

1.4 นิยามศัพท์เฉพาะ

1.4.1 Vision System หมายถึง เทคโนโลยีการใช้กล้องถ่ายภาพในการตรวจสอบและวิเคราะห์ตามเงื่อนไขที่ถูกตั้งค่าไว้ในตัวควบคุมให้ตรวจจับภาพคุณลักษณะใดคุณลักษณะหนึ่ง

1.4.2 Artificial Intelligence หมายถึง คอมพิวเตอร์ที่ถูกนำมาใช้ในการจำลองการคิดให้ใกล้เคียงกับมนุษย์โดยอาศัยการคำนวณทางคณิตศาสตร์ที่มาความซับซ้อนมาประยุกต์เพื่อให้เกิดการเรียนรู้สิ่งที่ได้รับการฝึกสอนเข้าไป และสามารถแสดงออกซึ่งความสามารด้านใดด้านหนึ่งเสมือนการคิดหรือการกระทำของมนุษย์

1.4.3 Machine Learning หมายถึง ส่วนย่อยหนึ่งของระบบปัญญาประดิษฐ์ที่มีส่วนทำให้คอมพิวเตอร์สามารถเรียนรู้และปรับปรุงรูปแบบได้ด้วยตนเอง ซึ่งความสามารถนี้เพิ่มขึ้นทุกๆ ครั้งที่มีการฝึกสอน (Train)

1.4.4 Deep Learning หมายถึง เทคโนโลยี Machine Learning ในการสร้างปัญญาประดิษฐ์ โดยใช้โครงข่ายประสาทเทียมหรือข่ายงานประสาทเทียมหลายๆ ชั้นเหมือนแบบจำลองของสมองมนุษย์ในการกระมวลผลข้อมูลที่มีความซับซ้อนและชุดข้อมูลขนาดใหญ่

1.4.5 YOLO v11 หมายถึง โมเดลอัลกอริทึ่มในการตรวจจับวัตถุ ซึ่ง YOLO ย่อมาจาก “You Only Look Once” และเป็นนับรุ่นที่ 11 ที่มีการพัฒนาต่อเนื่องมาจนถึงปัจจุบัน (เมษายน 2568) ซึ่งมีประสิทธิภาพสูงสุดเมื่อเทียบกับรุ่นก่อนๆ ที่ผ่านมา

1.4.6 Class หมายถึง ลำดับชั้นหรือประเภทของวัตถุในภาพที่ถูกสร้างป้ายชื่อ (label) หรือป้ายกำกับ (Annotation) เพื่อให้คอมพิวเตอร์ทำการเรียนรู้ ทดสอบ หรือทำนาย

1.4.7 Accuracy หมายถึง ความแม่นยำของการตรวจจับวัตถุที่ตรวจพบ

1.4.8 Precision หมายถึง ความเที่ยงตรงของการตรวจจับวัตถุที่ตรวจพบ

1.4.9 Recall หมายถึง ความสามารถของโมเดลในการระบุจำนวนครั้งทั้งหมดของวัตถุแต่ละประเภทหรือลำดับชั้นปรากฏในภาพ

1.4.10 Mean Average Precision (mAP50) หมายถึง ค่าเฉลี่ยของการตรวจจับวัตถุ โดยมีเกณฑ์ค่า Intersection over Union (IoU) ที่ 0.50 ขึ้นไป

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ลดภาระในการตรวจสอบชิ้นงานที่มีข้อบกพร่องของพนักงาน

1.5.2 ลดโอกาสหลุดรอดของชิ้นงานที่มีปัญหาคุณภาพไปยังลูกค้า

1.5.3 ได้ระบบปัญญาประดิษฐ์ในการตรวจจับข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์ด้านความสมบูรณ์ของตัวอักขระ

1.5.4 ได้แนวทางการฝึกสอนปัญหาประดิษฐ์ YOLO v11 เพื่อสร้างโมเดลที่มีความถูกต้องแม่นยำที่สร้างมาไปใช้ต่อยอดในการตรวจจับข้อบกพร่องอื่นๆ ในอนาคตได้

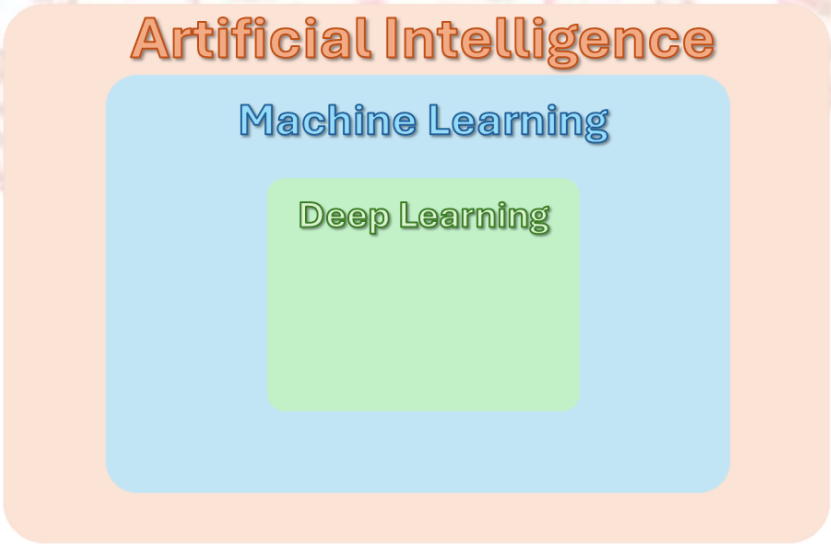
บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

การพัฒนาปัญญาประดิษฐ์เป็นหัวข้อที่ได้รับความสนใจเพิ่มมากขึ้นในช่วงหลายปีที่ผ่านมา เนื่องจากความสามารถของปัญญาประดิษฐ์ที่ช่วยทำให้มนุษย์สามารถทำงานได้อย่างสะดวกมากขึ้น และมีการนำไปใช้งานมากมายหลายด้านรวมถึงในอุตสาหกรรมการผลิต เพื่อให้การวิจัยนี้สามารถดำเนินการให้เกิดผลได้ การศึกษาทฤษฎีและแนวคิดที่เกี่ยวข้องจึงเป็นสิ่งสำคัญอย่างยิ่ง แม้ว่าปัญญาประดิษฐ์จะมีเครื่องมือต่างๆ ที่สามารถใช้งานได้อย่างง่ายดายแต่การศึกษาทำความเข้าใจหลักการทำงานของปัญญาประดิษฐ์ก็เป็นสิ่งที่ไม่สามารถมองข้ามได้ ซึ่งในบทนี้จะกล่าวถึงแนวคิดหลักและทฤษฎีที่สนับสนุนการทำงานของปัญญาประดิษฐ์ (Artificial Intelligence) โดยจะเน้นที่พื้นฐานด้านการประมวลผลภาพ (Image Processing), การเรียนรู้ของเครื่อง (Machine Learning), การเรียนรู้เชิงลึก (Deep Learning) และการตรวจจับวัตถุ (Object Detection) อันจะนำไปสู่การต่อยอดพัฒนาในบทถัดไปตามจุดประสงค์ของการวิจัยในครั้งนี้

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1.1 การประมวลผลภาพ (Image Processing) เป็นส่วนสำคัญหนึ่งในการวิจัยนี้เนื่องจากปัญญาประดิษฐ์จะต้องเรียนรู้ลักษณะของชิ้นงานทั้งชิ้นงานดีและชิ้นงานที่มีข้อบกพร่อง โดยการเรียนรู้นั้นจะต้องฝึกสอนผ่านรูปภาพที่ได้ทำการเตรียมไว้ ซึ่งรูปภาพเหล่านั้นก็จะต้องผ่านการประมวลผลภาพมาก่อน เช่น การปรับขนาดขนาดของภาพ การปรับปรุงคุณภาพของภาพ หรือการปรับความละเอียดของภาพ เป็นต้น และจากการค้นคว้าพบว่ามีผู้ให้คำนิยามเกี่ยวกับการประมวลผลภาพไว้ดังนี้ การประมวลผลภาพ (Image processing) คือ กระบวนการวิเคราะห์ข้อมูลรูปภาพในลักษณะของ ข้อมูลดิจิทัล โดยการวิเคราะห์จะประมวลผลผ่าน คอมพิวเตอร์ด้วยวิธีการคำนวณต่าง ๆ ที่ทำให้ได้ คุณสมบัติทั้งในเชิงคุณภาพและปริมาณ (กวิน, และคนอื่นๆ, 2564) ซึ่งสอดคล้องกับงานวิจัยของ (เอกรัตน์, 2566) โดยในการวิจัยนี้จะใช้กล้องอุตสาหกรรมพร้อมเลนส์ที่มีใช้งานอยู่แล้วภายในโรงงานในการถ่ายภาพเพื่อนำมาเข้าสู่กระบวนการประมวลผลภาพ เพื่อให้เกิดต้นทุนในการสร้างระบบปัญญาประดิษฐ์โดยใช้งบประมาณน้อยที่สุด ซึ่งจะกล่าวถึงรายละเอียดของอุปกรณ์ในหัวข้อ 2.1.6 เป็นลำดับต่อไป

2.1.2 ปัญญาประดิษฐ์ (Artificial Intelligence) คือ เทคโนโลยีทางด้านคอมพิวเตอร์ที่ถูกทำให้สามารถเรียนรู้และเข้าใจวิธีการคิดของมนุษย์ได้เพื่อให้คอมพิวเตอร์สามารถทำงานได้ด้วยตัวของมันเองเสมือนว่ามี “ปัญญา” ในการคิด วิเคราะห์ หรือ สังเคราะห์ข้อมูล หรือดำเนินการตามที่มนุษย์ต้องการได้ ซึ่งสอดคล้องกับแนวคิดของ (กฤษฎา และ สายรุ้ง, 2567) ซึ่งกล่าวว่า ปัญญาประดิษฐ์ (Artificial Intelligence) หรือ AI คือ ศาสตร์แขนงหนึ่งของวิทยาการคอมพิวเตอร์ที่มุ่งเน้นพัฒนาให้คอมพิวเตอร์สามารถกระทำการต่างๆ ได้อย่างชาญฉลาด คล้ายกับที่มนุษย์กระทำได้ นอกจากนี้ผู้วิจัยได้ศึกษาเกี่ยวกับปัญญาประดิษฐ์จากงานวิจัยของ (ยุวเรศมคฐ์, 2564) ซึ่งได้กล่าวถึงการพัฒนาปัญญาประดิษฐ์ไว้ว่ามีวิวัฒนาการมาตั้งแต่ปี พ.ศ. 2499 (ค.ศ. 1956) ในการประชุมที่วิทยาลัยดาร์ทเมาท์ (Dart-mouth College) รัฐนิวแฮมเชียร์ ประเทศสหรัฐอเมริกาโดย John McCarthy และทีมได้กำหนดคำว่า Artificial Intelligence หรือปัญญาประดิษฐ์ขึ้นหลังจากนั้นได้มีการพัฒนาการเรียนรู้ของเครื่อง (Machine Learning) ขึ้นในปี พ.ศ. 2523 และเกิดรูปแบบการเรียนรู้ผ่านเครือข่ายใยประสาทเทียม (Artificial neural network หรือ ANN) แบบเรียนรู้เชิงลึก (Deep Learning) ในปี พ.ศ.2553 เป็นต้นมา ซึ่งเป็นหัวใจสำคัญที่ทำให้ AI มีความสามารถในการคิดวิเคราะห์ได้ใกล้เคียงสมองของมนุษย์และมีความฉลาดขึ้น ซึ่งจากงานวิจัยดังกล่าวทำให้สามารถแบ่งระดับการทำงานของปัญญาประดิษฐ์ได้ 3 ระดับ ได้แก่ระดับบนสุดคือ Artificial Intelligence ระดับถัดไปคือMachine Learning และระดับลึกที่สุดคือ Deep Learning ดังภาพที่ 2-1



ภาพที่ 2-1 การทำงานในระดับต่างๆ ของปัญญาประดิษฐ์

2.1.3 การเรียนรู้ของเครื่อง (Machine Learning) คือ กระบวนการเรียนรู้ของคอมพิวเตอร์โดยอาศัยชุดข้อมูล (Dataset) เพื่อสร้างเป็นโมเดลการเรียนรู้ ทำให้คอมพิวเตอร์สามารถเข้าใจ คิด และตัดสินใจอย่างใดอย่างหนึ่งได้ด้วยตัวมันเอง ตามงานวิจัยของ (เวณิกา , พัชรี , ศุภโชค, และ ทุนพัฒนานักวิจัยระดับบัณฑิตศึกษา ประจำปี 2565, 2023) ให้ความหมายไว้ว่าหมายถึง การทำให้ระบบคอมพิวเตอร์สามารถประมวลผลคาดการณ์ ตัดสินปัญหาต่าง ๆ ด้วยตนเองผ่านการเรียนรู้ชุดข้อมูลที่ป้อนเข้าไปโดยอาศัยการเรียนรู้จากโมเดลของข้อมูลนำเข้าเป็นตัวอย่างทำให้ Machine Learning สามารถเรียนรู้ เข้าใจ และจะสามารถทำนายข้อมูลได้ ซึ่งการพัฒนาการเรียนรู้ของเครื่อง (Machine Learning) สามารถแบ่งประเภทไว้ดังต่อไปนี้ (พิพัฒน์, 2563)

1. Supervised Learning เป็นการสอนเครื่องจักรให้เรียนรู้การแบ่งชุดข้อมูลเพื่อสร้างโมเดลการอนุมาน โดยทำตามแบบจากชุดข้อมูลในอดีตที่มีการระบุ Input และ Output ไว้อย่างชัดเจน
2. Unsupervised Learning เป็นการสอนเครื่องจักรให้เรียนรู้จากชุดข้อมูลที่ไม่มีการแบ่งกลุ่ม หรือระบุความสัมพันธ์ของข้อมูลไว้ชัดเจน เพราะฉะนั้นการเรียนรู้แบบนี้เครื่องจักรมีหน้าที่ต้องหาความสัมพันธ์และแบ่งกลุ่มของข้อมูลก่อนที่จะสร้างโมเดลการอนุมานขึ้นมา
3. Reinforcement Learning เป็นการสอนให้เครื่องจักรเรียนรู้ที่จะคิดหากลยุทธ์ที่ดีที่สุดจากสภาพแวดล้อม เพื่อที่จะได้รับ “รางวัล” หรือ สิ่งตอบแทนที่กำหนดไว้

2.1.4 การเรียนรู้เชิงลึก (Deep Learning) คือ การจำลองการทำงานของคอมพิวเตอร์ให้เหมือนกับการทำงานของสมองมนุษย์ที่มีความซับซ้อนของการทำงาน ซึ่งการทำงานในระดับนี้มีการเชื่อมโยงการทำงานย่อยๆ (node) เข้าด้วยกัน จึงมีความแตกต่างจากการทำงานในขั้น Machine Learning เนื่องจากการเรียนรู้เชิงลึกอาศัยโครงข่ายประสาทเทียม (Artificial Neural Network) มีหลักการทำงานโดยจำลองมาจากสมองของสิ่งมีชีวิตที่มีความซับซ้อน โดยใช้โมเดลทางคณิตศาสตร์ซึ่งแนวคิดเริ่มต้นได้มาจากการศึกษาโครงข่ายไฟฟ้าชีวภาพในสมอง โครงข่ายประสาทเกิดจากการเชื่อมต่อระหว่างเซลล์ประสาทจนเป็นเครือข่ายที่ทำงานร่วมกัน (ปฏิญญา, ไตรรัตน์, ดัชกรณ์, ณัฐชัย, และ ศีตภา, 2564) การเรียนรู้ในเชิงลึกเป็นเทคนิคการวิเคราะห์ข้อมูลสมัยใหม่ซึ่งเริ่มเป็นที่นิยมอย่างแพร่หลายหลังจากการเอาชนะการแข่งขัน ImageNet Large Scale Visual Recognition

Challenge (ILSVRC) ในปี ค.ศ. 2012 ส่งผลให้การวิเคราะห์รูปภาพได้รับการพัฒนาอย่างต่อเนื่อง สิ่งสำคัญอย่างหนึ่งที่ทำให้ Deep Learning ให้ผลลัพธ์ที่เหนือกว่าเทคนิคการวิเคราะห์แบบดั้งเดิมคือ เป็นเทคนิคที่มีการสร้างวิธีดึงคุณลักษณะสำคัญของรูปภาพให้โดยอัตโนมัติ (Automatic Feature Extraction) ทำให้ตัวแบบ (Model) สามารถเรียนรู้คุณลักษณะต่าง ๆ ของรูปภาพได้หลากหลาย ส่งผลให้ตัวแบบมีความยืดหยุ่นในการนำไปใช้งานจริงได้มากกว่าเทคนิคการวิเคราะห์แบบดั้งเดิม ซึ่งได้ถูกกล่าวไว้ในงานวิจัยของ (ปัญจวรรษ และ ฐิติรัตน์, 2566) ซึ่งเราสามารถแยกความแตกต่างของการทำงานในระดับ Machine Learning และ Deep Learning ได้ดังตารางที่ 2-1

ตารางที่ 2-1 ความแตกต่างระหว่าง Machine Learning และ Deep Learning

Machine Learning	Deep Learning
เป็นส่วนย่อยของปัญญาประดิษฐ์	เป็นส่วนย่อยของ Machine Learning
สามารถฝึกสอนได้ด้วยข้อมูลไม่มาก	จำเป็นต้องใช้ข้อมูลจำนวนมากในการฝึกสอน
ต้องใช้คนในการแทรกแซงเพื่อแก้ไขให้ถูกและเรียนรู้	เรียนรู้ได้ด้วยตัวเองจากสภาพแวดล้อมและความผิดพลาดในอดีต
การฝึกสอนรวดเร็วแต่ความแม่นยำต่ำ	การฝึกสอนใช้เวลานานแต่ความแม่นยำสูง
สามารถฝึกสอนโดยใช้หน่วยประมวลผลกลาง (CPU)	จำเป็นต้องใช้หน่วยประมวลผลกราฟิก (GPU) เฉพาะในการฝึกสอน

2.1.5 การตรวจจับวัตถุ (Object Detection) คือ การทำงานของคอมพิวเตอร์ในค้นหาสิ่งที่เป็นเป้าหมายซึ่งถูกโปรแกรมเอาไว้จากภาพที่ทำการป้อนเข้าไปในกระบวนการตรวจจับ หากคอมพิวเตอร์รู้จักสิ่งนั้นก็จะสามารถที่จะตรวจจับได้ ซึ่งสอดคล้องกับงานวิจัยของ (สิริทัศน์, 2563) ซึ่งให้คำอธิบายไว้ว่า "เทคโนโลยีตรวจจับวัตถุ" (Object Detection) คือ หนึ่งในฟีเจอร์หลักของ AI (Artificial Intelligence) ที่ใช้กับกล้องหรือกล้องวิดีโอ ซึ่งสามารถค้นหาสิ่งของโดยนำ AI มาวิเคราะห์ข้อมูล จากการมองเห็นของคอมพิวเตอร์ (Computer Vision) และการประมวลผลภาพ (Image Processing) เพื่อตรวจจับวัตถุที่อยู่ในรูปหรือวิดีโอ เช่น มนุษย์ สัตว์ สิ่งของ รถยนต์ อาคาร และวัตถุอื่น ๆ ที่อยู่ในรูปภาพ หรือวิดีโอ

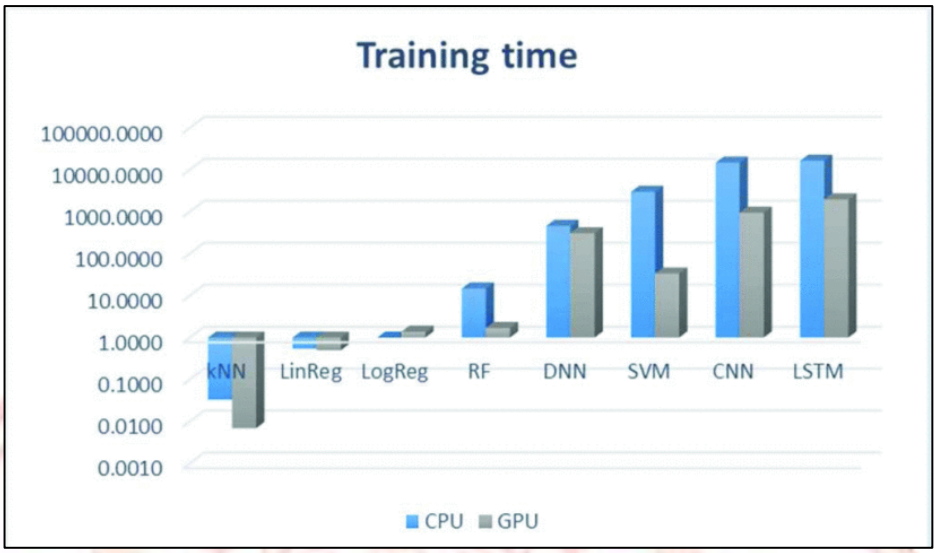
2.1.6 อัลกอริทึม YOLO (You Only Look Once) เป็นอัลกอริทึมสำหรับฝึกสอนปัญญาประดิษฐ์ในการจดจำลักษณะของชิ้นงานและใช้ในตรวจจับวัตถุตามการเรียนรู้ที่ได้สร้างขึ้นถูกพัฒนาขึ้นตั้งแต่ปี ค.ศ. 2015 โดย Joseph Redmon และ Ali Farhadi แห่งมหาวิทยาลัยวอชิงตัน (Ultralytics, n.d.) และมีการพัฒนาอย่างต่อเนื่องมาจนถึงปัจจุบันซึ่งนับเป็นรุ่นที่ 11 YOLO มีโมเดลการเรียนรู้หลากหลายประเภทที่เปิดให้ผู้พัฒนาสามารถนำไปพัฒนาต่อได้อย่างไม่มีค่าใช้จ่าย ซึ่งหากต้องการใช้งานในเชิงพานิชย์ก็สามารถที่จะซื้อสิทธิ์การใช้งาน (License) ได้อีกด้วยโดยในการวิจัยนี้จะพัฒนาภายใต้สิทธิ์การใช้งานแบบ AGPL-3.0 ซึ่งเป็นสิทธิ์การใช้งานเพื่อการวิจัย

โมเดลการทำงานของ YOLO นั้นมีหลากหลายรูป ได้แก่ การตรวจจับ (Detection), การแบ่งส่วน (Segmentation), การจัดแบ่งประเภท (Classification) การคาดการณ์ท่าทางการเคลื่อนไหว (Pose Estimation), การตรวจจับแบบกรอบที่มีทิศทาง (Oriented Bounding Boxes) และการติดตาม (Tracking) จากงานวิจัยของ (กรณิการ์, 2565) ได้ให้ข้อมูลไว้ว่า YOLO เป็นอัลกอริทึมที่สามารถตรวจจับวัตถุได้จากการส่งผ่านรูปภาพเข้าไปในระบบเพียงครั้งเดียว โดยใช้หลักการโครงข่ายประสาทแบบคอนโวลูชัน ที่นำแนวคิดของการทำนายตำแหน่งของวัตถุ (Localization) และจัดจำแนกวัตถุว่าเป็นชนิดอะไร (Classification) โดยใช้กรอบล้อมวัตถุ (Bounding Box) โดยจะทำนายทั้งกรอบล้อมวัตถุ และความน่าจะเป็นของวัตถุบางส่วนที่อยู่ในกรอบออกมาพร้อมกันทีเดียว จึงทำให้ YOLO มีความสามารถที่ทำงานได้แบบเวลาจริง (Real-time) และมีความเร็วที่เหนือกว่าอัลกอริทึมอื่น ๆ จึงถูกนำไปพัฒนาอย่างแพร่หลาย

ภาพที่ 2-2 ตัวอย่างการตรวจจับ (Detection) ยานพาหนะประเภทต่างๆ ของ YOLO

2.1.7 ซอฟต์แวร์ CUDA (Compute Unified Device Architecture) คือ แพลตฟอร์มสำหรับการประมวลผลแบบคู่ขนานและ Application Programing Interface (API) พัฒนาโดยบริษัท Nvidia เพื่อให้นักพัฒนาและวิศวกรซอฟต์แวร์สามารถดึงศักยภาพในการประมวลผลแบบขนานของ GPU (Graphic Processing Unit) สำหรับการประมวลผลในงานต่างๆ หรือที่เรียกว่า GPGPU (General-Purpose computing on Graphics Processing Units) ซึ่งผู้วิจัยได้ศึกษาเกี่ยวกับการพัฒนาโมเดลปัญญาประดิษฐ์จะต้องทำการฝึกสอนคอมพิวเตอร์ให้จดจำภาพของสิ่งที่ต้องการให้คอมพิวเตอร์จำแนกหรือแยกแยะได้เป็นจำนวนมาก และกระบวนการฝึกสอนนี้ใช้เวลายาวนานหลายชั่วโมง แต่หากมีการใช้งานซอฟต์แวร์ CUDA จะสามารถทำการฝึกสอนได้เร็วยิ่งขึ้น ซึ่งการจะใช้งาน CUDA ต้องมีฮาร์ดแวร์การ์ดแสดงผลหรือ GPU ของ Nvidia รุ่นที่รองรับจึงสามารถใช้งานได้

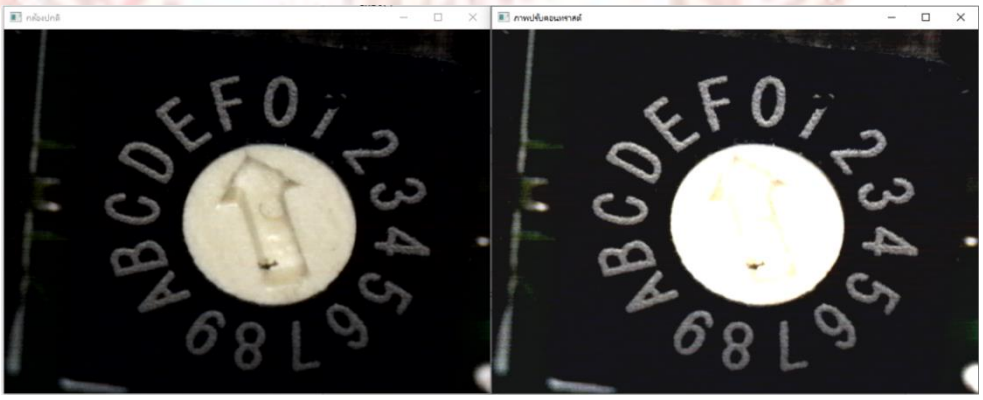
โดยปกติแล้วในเครื่องคอมพิวเตอร์หน้าที่สำหรับการประมวลผลจะเป็นหน้าที่ของ CPU (Central Processing Unit) ซึ่งเป็นหน่วยประมวลผลกลางในการทำงานของคอมพิวเตอร์ ในสมัยก่อนการประมวลผลทางด้านกราฟฟิกจะใช้ CPU เป็นหลัก ส่งผลให้การทำงานของ CPU หนักขึ้นและไม่เพียงพอต่อการทำงานในด้านอื่นๆ GPU จึงเข้ามามีบทบาทในการประมวลผลทางด้านกราฟฟิกแทน เพื่อลดการทำงานของ CPU ทำให้การงานด้านกราฟฟิกมีความลื่นไหลมากยิ่งขึ้น และเมื่อ CPU ทำงานน้อยลง ความร้อนภายในเครื่องก็ลดลงด้วยเช่นกัน (วชิดชนก, 2558) นอกจากนี้จากงานวิจัยที่ศึกษาเรื่องสมรรถนะของ CPU และ GPU ในการทำงานในการฝึกสอนชุดข้อมูลให้โครงข่ายประสาทเทียม ก็พบว่า GPU ใช้เวลาในการฝึกสอนน้อยกว่า CPU ในหลายโมเดลที่ทำการวิจัย โดยเฉพาะโมเดลประเภท SVM, CNN และ kNN นั้น GPU สามารถทำได้รวดเร็วกว่า CPU อย่างมีนัยสำคัญ โดยเวลาที่ใช้ในการฝึกสอนด้วยโมเดลประเภท CNN ซึ่งเป็นประเภทการทำงานเดียวกับอัลกอริทึมอย่าง YOLO นั้นใช้เวลาเกิน 15 นาทีไปเล็กน้อยสำหรับการฝึกสอนด้วย GPU ในขณะที่ CPU ใช้เวลาถึงกว่า 4 ชั่วโมงเมื่อเปรียบเทียบกันพบว่า GPU ทำการฝึกสอนได้เร็วกว่า CPU อยู่ราว 16 เท่า (Štaka, Mišić, & Tomašević, 2025) ซึ่งชุดข้อมูลที่ใช้ในการฝึกสอนนั้นมีชื่อว่า “Stanford Dogs” ประกอบด้วยรูปสุนัขจำนวน 120 สายพันธุ์ ซึ่งหมายความว่ามีการแบ่ง class ไว้จำนวน 120 class และมีจำนวนรูปภาพทั้งสิ้น 20,580 ภาพด้วยกัน ดังนั้นจากผลของการวิจัยนี้ทำให้เห็นได้ว่าในการฝึกสอนการตรวจจับด้วยโมเดล YOLO หากใช้คอมพิวเตอร์ที่มี GPU ในการฝึกสอนก็น่าที่จะทำได้รวดเร็วกว่าคอมพิวเตอร์ที่ทำงานด้วย CPU เพียงอย่างเดียว



ภาพที่ 2-3 เวลาในการฝึกสอนของCPU และ GPU (Štaka, Mišić, & Tomašević, 2025)

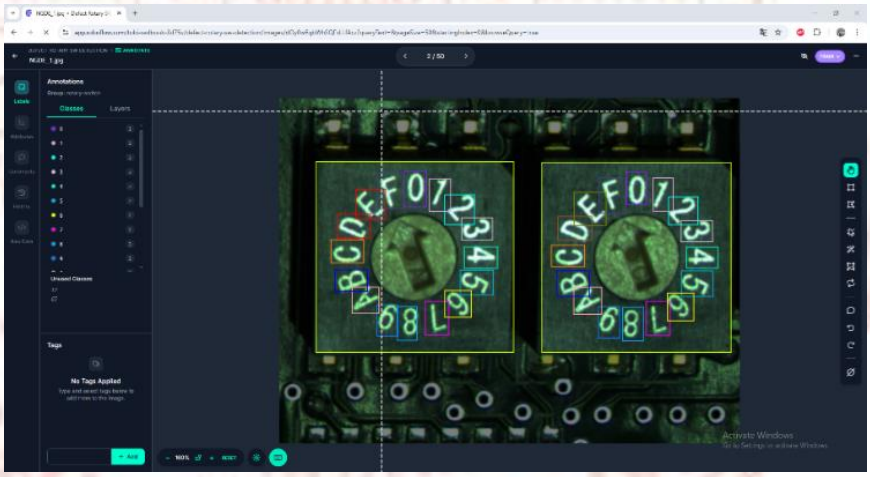
2.1.8 ไลบรารี OpenCV (Open-Source Computer Vision) เป็นไลบรารีแบบเปิดเผยโค้ด (Open Source) ที่รวมอัลกอริทึมการมองเห็นด้วยคอมพิวเตอร์ (Computer Vision) หลายร้อยรายการเข้าไว้ด้วยกัน ทำให้สะดวกแก่การใช้งาน และมีโครงสร้างแบบโมดูลาร์ใช้สำหรับการสั่งงานกล้องเพื่อประมวลภาพ (Image Processing) เช่น การเบลอภาพ, การกลับภาพ, การลดหรือเพิ่มขนาดภาพ, การปรับสีของภาพ เป็นต้น

OpenCV มีการพัฒนาอย่างต่อเนื่องซึ่งในปัจจุบันเป็นรุ่น 4.10 (OpenCV, 2024) ซึ่งตัวอย่างรูปภาพที่ผ่านการประมวลผลภาพด้วย OpenCV ดูได้จากภาพที่ 2-4 โดยรูปฝั่งซ้ายคือรูปต้นฉบับที่ได้จากกล้องและรูปทางขวามีการปรับความเข้มของแสง (Contrast) เพิ่มขึ้นทำให้ส่วนที่เป็นสีขาวของภาพมีความชัดเจนขึ้น



ภาพที่ 2-4 รูปภาพที่ผ่านการประมวลผลภาพด้วย OpenCV

2.1.9 Roboflow เป็นแอพพลิเคชั่นออนไลน์ที่มีเครื่องมือสำหรับการสร้างป้ายกำกับ (Label) หรือการใส่คำอธิบายภาพ (Annotation) ซึ่งมีความจำเป็นสำหรับการฝึกสอนโมเดลปัญญาประดิษฐ์ โดยการใส่คำอธิบายนี้เป็นการกำหนดขอบเขตของสิ่งที่ต้องการให้ปัญญาประดิษฐ์ทำการเรียนรู้ภายในขอบเขตของรูปภาพที่เราเตรียมไว้สำหรับการฝึกสอนด้วยการตีกรอบล้อมรอบส่วนที่เราต้องการจะให้โมเดลเรียนรู้หรือจดจำ นอกจากนี้ Roboflow ยังสามารถทำการนำข้อมูลที่ถูกใส่คำอธิบายไว้ออกมาเป็นชุดข้อมูล (Dataset) ที่สามารถใช้งานร่วมกับ YOLO รุ่นต่างๆ ได้อีกด้วย ข้อดีของ Roboflow อย่างหนึ่งคือสามารถใช้งานผ่าน Browser ได้โดยไม่ต้องติดตั้งแอพพลิเคชั่นลงในคอมพิวเตอร์เพียงเข้าไปที่ **https://roboflow.com** ลงทะเบียนผู้ใช้งานจากนั้นก็จะสามารถใช้งานได้ทันที และยังบันทึกภาพที่ได้ทำการอัพโหลดขึ้นไปเพื่อการสร้างป้ายกำกับไว้บน Cloud และส่งออก (Export) ชุดข้อมูล (Dataset) ออกมาเป็นรูปแบบตามที่เราต้องการได้ ซึ่งในการวิจัยนี้คือชุดข้อมูลที่มีรูปแบบเข้ากันได้กับ Yolo v11



ภาพที่ 2-5 การสร้างป้ายกำกับด้วย Roboflow

2.1.10 ไลบรารี PyTorch เป็นไลบรารีสำหรับสร้างการเรียนรู้ของเครื่อง (Machine Learning) ร่วมกับอัลกอริทึม YOLO ซึ่งต้องเขียนคำสั่งผ่านภาษา Python เช่นเดียวกับไลบรารี OpenCV จากการสืบค้นบนอินเตอร์เน็ตพบว่า คือ Library ที่ถูกพัฒนาขึ้นโดยมีต้นแบบจาก Library ชื่อว่า Torch มักถูกนำไปใช้งานในด้าน Machine Learning และ Deep Learning PyTorch คือ open-source machine learning ที่ถูกพัฒนาขึ้นจากทีม Facebook’s AI Research lab และเริ่มปล่อยให้ใช้งาน

ในปี 2016 (TensorGym, n.d.) ทั้งนี้การทำงานของ PyTorch รองรับการประมวลผลด้วย GPU ซึ่งจะทำให้การประมวลผลรวดเร็วกว่าการประมวลผลด้วย CPU ดังที่กล่าวมาแล้วข้างต้น

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยของ วุฒิชัย วัชรารัตน์ เจษฎา ตัณฑนุช เรื่องการจำแนกพันธุ์ข้าวจากภาพของเมล็ดข้าวสารด้วยวิธีการตรวจหาวัตถุ ซึ่งได้ทำการประยุกต์ใช้วิธีการตรวจจับวัตถุด้วยโมเดล YOLO version 5 ให้สามารถจำแนกพันธุ์ข้าวจากภาพของเมล็ดข้าวสารพันธุ์คาราก้าดาก หอมมะลิ ยิปซาลา บาสมาติ และอาโบริโอ เพื่อใช้เป็นต้นแบบในการทำวิจัยเกี่ยวกับการจำแนกเมล็ดข้าวสารซึ่งอาจมีการปลอมปนของเมล็ดข้าวอื่น ผลการวิจัยพบว่าโมเดล YOLO version 5 สามารถจำแนกพันธุ์ข้าวด้วยเมล็ดข้าวสารทั้ง 5 สายพันธุ์ได้ดี จากการประเมินความแม่นยำของโมเดลที่ค่าขีดแบ่ง 0.6 ซึ่งภาพเมล็ดข้าวสารมีการซ้อนทับกันแตกต่างกันคิดเป็นร้อยละ 5, 10, 15, 20 และ 25 ซึ่งโปรแกรมสามารถจำแนกได้ถูกต้องร้อยละ 99.13, 99.00, 98.62, 98.19, 97.56 และ 96.89 ตามลำดับ (วุฒิชัย และเจษฎา, 2566)

2.2.2 การพัฒนาโปรแกรมจำแนกประเภทหน้ากากและตรวจจับการสวมใส่ด้วยเทคโนโลยีประมวลผลภาพ งานวิจัยของ วรวิทย์ ฝันคำอ้าย ศุภชัย วงศ์นภาวิเศษ สหวรรษ กิติยะ และนงนุช เกตุ้ย ได้ทำการศึกษาในหัวข้อ การพัฒนาโปรแกรมจำแนกประเภทหน้ากากและตรวจจับการสวมใส่ด้วยเทคโนโลยีประมวลผลภาพ เพื่อจำแนกประเภทหน้ากากและตรวจจับการสวมใส่หน้ากาก ซึ่งได้ทำการพัฒนาโปรแกรมจำแนกประเภทหน้ากากและตรวจจับการสวมโดยใช้อัลกอริทึม YOLO version 4 และ 5 ซึ่งเมื่อทำการฝึกสอนปัญญาประดิษฐ์และนำมาใช้งานจริงพบว่ามีความถูกต้องของการจำแนกหน้ากากผ้าได้ถูกต้องสูงสุด จำนวน 11 คนจาก 12 คน หรือคิดเป็นร้อยละ 91 (วรวิทย์, ศุภชัย, สหวรรษ, และ นงนุช, 2566)

2.2.3 การศึกษาการตรวจจับสิ่งเจือปนในผักโขมแช่แข็งด้วยการเรียนรู้เชิงลึกโมเดล YOLOv4 จากงานวิจัยของ สุวรรณกาญจน์ สุพมาตรา ศุภกิตต์ สายสุนทร และอนุสรณ์ เชื้อสามารถ ได้ทำการศึกษาการตรวจจับสิ่งเจือปนในผักโขมแช่แข็งด้วยการเรียนรู้เชิงลึกโมเดล YOLO version 4 เพื่อเพิ่มความแม่นยำการตรวจจับสิ่งเจือปนในอาหาร ซึ่งชุดข้อมูลที่นำมาฝึกสอนปัญญาประดิษฐ์ได้แก่ภาพผักโขม, ไม้, แก้ว, เชือก, พลาสติก และหิน โดยการใช้แบบจำลอง YOLO version 4 พบว่ามีประสิทธิภาพในการตรวจจับพลาสติกสูงสุด รองลงมาคือ เชือก แก้ว หิน และไม้ตามลำดับ โดยค่าความถูกต้อง (Accuracy Score) อยู่ที่ร้อยละ 52.22, ค่าความแม่นยำ(Precision Score) อยู่ที่ร้อยละ 52.94, ค่ารีคอล (Recall) อยู่ที่ร้อยละ 55.38 และค่าเอฟ-1สกอร์ (F-1 Score) อยู่ที่ร้อยละ 54.14 โดยกลุ่มผู้วิจัยได้อภิปรายถึงสาเหตุที่การทำนายเกิดมาค่าต่างๆ ต่ำว่าเกิดจากการเตรียมชุด

ข้อมูลรูปภาพที่ใช้ในการฝึกสอนนั้นไม่ได้มีการรวมสิ่งเจือปนแต่ละประเภทในรูปเดียวกันกับรูปผักโขม (สุวรรณกาญจน์, ศุภกิตต์, และ อนุสรณ์, 2566)

2.2.4 จากการศึกษางานวิจัยของ อิสราภรณ์ อมรสวัสดิ์วัฒนา ศุภฤกษ์ จันทร์ศุภเสน และ อลงกรณ์ นมะหุต เรื่องเครื่องนับลูกไก่ ซึ่งได้ทำการพัฒนาเครื่องนับลูกไก่เพื่อแทนการใช้แรงงานคนในการนับเนื่องจากอาจเกิดความผิดพลาดได้เนื่องจากความเหนื่อยล้า โดยผู้วิจัยได้ใช้การประมวลผลแบบเรียลไทม์ด้วย Jetson Nano Developer ซึ่งทำงานกับซอฟต์แวร์ CUDA และอัลกอริทึม YOLO ทั้งนี้กลุ่มผู้วิจัยได้ทำการทดสอบการนับลูกไก่บนสายพานที่ความเร็ว 0.2 เมตรต่อวินาที และ 0.5 เมตรต่อวินาที ซึ่งหากมีลูกไก่บนสายพานเพียง 1 ตัวผลของการนับนั้นไม่แตกต่างกันมากนัก กล่าวคือมีค่าความถูกต้องอยู่ที่ราวร้อยละ 90 แต่กรณีที่มีลูกไก่บนสายพาน 2-3 ตัว พบว่าค่าความถูกต้องลดลงเหลือร้อยละ 60 ที่ความเร็ว 0.2 เมตรต่อวินาที และร้อยละ 33 ที่ความเร็ว 0.5 เมตรต่อวินาที และกรณีที่มีลูกไก่บนสายพาน 5 ตัว ค่าความถูกต้องจะลดลงเหลือเพียงร้อยละ 28 ความเร็ว 0.2 เมตรต่อวินาที และร้อยละ 14 ที่ความเร็ว 0.5 เมตรต่อวินาที จากผลการทดลองนี้ผู้วิจัยได้สรุปผลการทดลองว่าควรทำการปล่อยลูกไก่บนสายพานครั้งละ 1 ตัวและใช้ความเร็วสายพานที่ 0.2 เมตรต่อวินาทีเพื่อให้ได้ความถูกต้องมากที่สุด (อิสราภรณ์, ศุภฤกษ์, และ อลงกรณ์, 2566)

2.2.5 Automatic Industry PCB Board DIP Process Defect Detection with Deep Ensemble Method เป็นงานวิจัยของ Li Yu-Ting, Paul Kuo และ Guo Jiun-In ซึ่งได้ทำการศึกษาเกี่ยวกับการตรวจจับด้วยวิธี Deep Ensemble ในกระบวนการผลิตแผงวงจร (PCB) ซึ่งหมายถึงการคือการผสมผสานโมเดลการเรียนรู้เชิงลึก (Deep Learning) หลายๆ โมเดลเข้าด้วยกัน เพื่อปรับปรุงประสิทธิภาพในการทำงาน โดยการรวมโมเดลเหล่านี้จะช่วยให้ได้ผลลัพธ์ที่มีความแม่นยำมากขึ้นและลดข้อผิดพลาดที่เกิดขึ้นจากการใช้โมเดลเดียว เพื่อประยุกต์ใช้ในการตรวจสอบข้อบกพร่องในกระบวนการบัดกรีแบบ DIP (Dual Inline Process) ซึ่งการตรวจจับแบบดั้งเดิมนั้นสิ้นเปลืองทั้งเวลาและแรงงาน การวิจัยนี้จึงได้นำอัลกอริทึม YOLO version 2 และResNet-101 มาใช้ในการพัฒนาเพื่อใช้ตรวจจับข้อบกพร่องในการบัดกรี PCB แทนการตรวจสอบด้วยแรงงาน เพื่อให้ได้อัตราการตรวจจับที่สูงและอัตราการแจ้งเตือนผิดพลาดที่ต่ำ ซึ่งผลจากการทดสอบจริงมีอัตราการตรวจจับอยู่ที่ร้อยละ 96.73 และ อัตราการตรวจจับผิดพลาด (False Alarm Rate) อยู่ที่ร้อยละ 19.73 เวลาที่ใช้ในการตรวจจับน้อยกว่า 15 วินาที ได้รับการพิสูจน์ว่ามีประสิทธิภาพในการช่วยชี้แนะพนักงานและซ่อมแซมข้อบกพร่องของการบัดกรี อีกทั้งยังสามารถลดความต้องการแรงงานในกระบวนการผลิตลงได้ร้อยละ 33 (Li, Kuo, & Guo, 2021)

2.2.6 ระบบตรวจจับการพกอาวุธปืนสั้นด้วย YOLOv8 โดยจตุพร ศรีสอาดรักษ์ผู้ดำเนินงานวิจัยนี้ได้กล่าวถึงการพัฒนาระบบตรวจจับอาวุธปืนสั้นด้วย YOLOV8 โดยการทดลองสร้างแบบจำลองของการตรวจจับอาวุธปืนสั้นขึ้นมาใน ลักษณะพกซ่อน 3 รูปแบบ ได้แก่ แบบมีซองพก

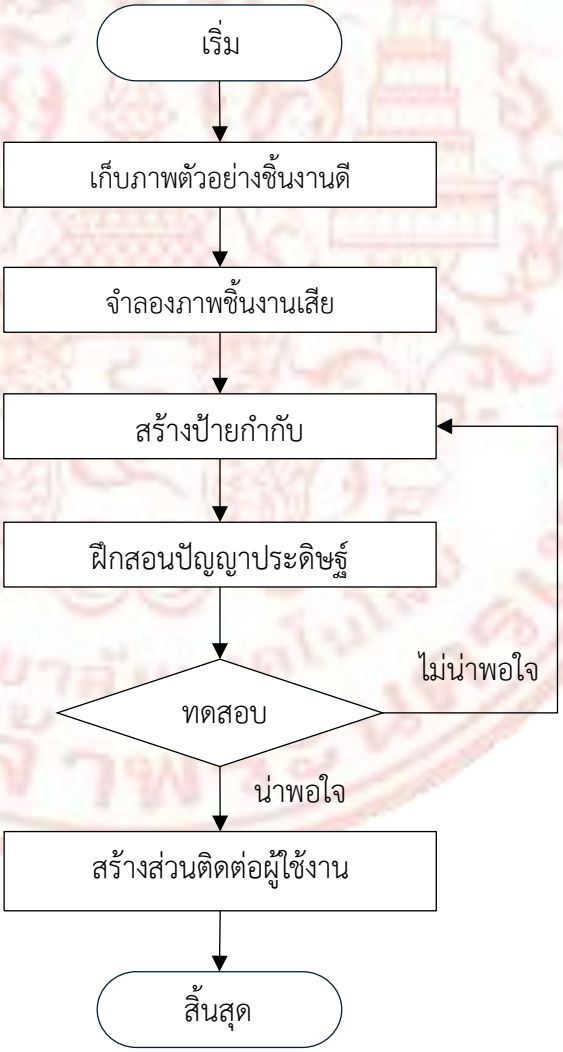
ด้านข้าง, พกซ่อนด้านหน้า-หลัง และแบบถืออาวุธปืนสั้น โดยใช้ชุดข้อมูลรูปภาพในการฝึกสอนแบบจำลองปัญญาประดิษฐ์ที่แบ่งออกเป็น 3 Class ได้แก่ 1. แบบมีซองพกด้านข้าง จำนวน 700 ภาพ 2. แบบพกซ่อนด้านหน้า-หลัง จำนวน 700 ภาพ และ 3. แบบถืออาวุธปืนสั้น จำนวน 700 ภาพ โดยรวมทั้งหมด 2,100 ภาพ ผลจากการฝึกสอนปัญญาประดิษฐ์ได้ค่าความแม่นยำ (Precision) อยู่ที่ 82% และค่าความถูกต้องของแบบจำลองหรือค่าการเรียกคืน (Recall) อยู่ที่ 81% ซึ่งในงานวิจัยนี้ได้แนะนำโมเดล YOLOv8 ฝึกสอนโมเดลจำนวน 50 รอบ โดยใช้รูปภาพที่มีความละเอียดอยู่ที่ 1560 พิกเซล ซึ่งคำแนะนำดังกล่าวเพียงพอต่อการตรวจสอบอาวุธปืนสั้นเท่านั้น และมุมมองภาพมีส่วนสำคัญของการตรวจจับที่มีผลต่อความแม่นยำและค่าความถูกต้องของโมเดล กล่าวคือ มุมมองของภาพที่เป็นมุมก้มและมุมเงย ต้องไม่เกิน 15 องศา และมีระยะในการตรวจจับที่แม่นยำอยู่ที่ 2 – 3 เมตร งานวิจัยนี้ได้แนะนำให้ใช้งานแบบประมวลผลแบบทันทีร่วมกับกล้อง CCTV ซึ่งจะต้องมีระยะในการตรวจจับอยู่ที่ 2 – 3 เมตร เช่นกัน (จตุพร , 2566)

2.2.7 การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผลการแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน โดยนัฒธี มละสาร ได้ทำการวิจัยโดยมีเป้าหมายเพื่อพัฒนาและศึกษาวิธีการตรวจจับรอยขีดข่วนบนพื้นผิวโลหะอะลูมิเนียมโดยใช้เทคนิคแบ่งภาพร่วมกับโครงข่ายประสาทเทียมแบบคอนโวลูชัน (CNN) โดยกระบวนการเริ่มจากการแยกภาพความละเอียดสูงออกเป็นส่วนย่อยๆ ในหลายรูปแบบ ได้แก่ 4, 9, 16, 36 และ 64 ส่วน เพื่อรักษารายละเอียดของภาพและช่วยให้ CNN สามารถวิเคราะห์ได้แม่นยำยิ่งขึ้น โมเดลที่ใช้ในการประมวลผลภาพย่อย คือ YOLOv8 ในเวอร์ชันต่างๆ ได้แก่ YOLOv8n, YOLOv8s และ YOLOv8x

ผลการทดลองชี้ให้เห็นว่า การแบ่งภาพก่อนนำเข้าสู่โครงข่ายประสาทช่วยเพิ่มอัตราการตรวจจับรอยขีดข่วนเมื่อเทียบกับการใช้ภาพเต็มโดยตรง โดยโมเดล YOLOv8s ทำงานได้ดีสุดในทุกกรณีที่ทดสอบ ชุดข้อมูลสำหรับฝึกสอนประกอบด้วยภาพรอยขีดข่วนจำนวน 2,600 ภาพ สำหรับตรวจสอบ 246 ภาพ และสำหรับทดสอบ 122 ภาพ ส่วนชุดข้อมูลทดสอบหลังการฝึกสอนประกอบด้วย 7 ภาพ ซึ่งมีรอยขีดข่วนรวม 48 รอย จากการทดสอบพบว่า เมื่อไม่แบ่งภาพ โมเดลที่เรียนรู้แล้วสามารถตรวจจับรอยขีดข่วนได้ 37 รอย คิดเป็น 77.08% แต่เมื่อแบ่งภาพเป็น 4 ส่วน สามารถเพิ่มอัตราการตรวจจับเป็น 90.90% และเมื่อแบ่งภาพในจำนวนส่วนที่มากขึ้น เช่น 9, 16, 36 หรือ 64 ส่วน สามารถตรวจจับรอยขีดข่วนได้ครบถ้วน 100% (นัฒธี, 2566)

บทที่ 3
วิธีดำเนินการวิจัย

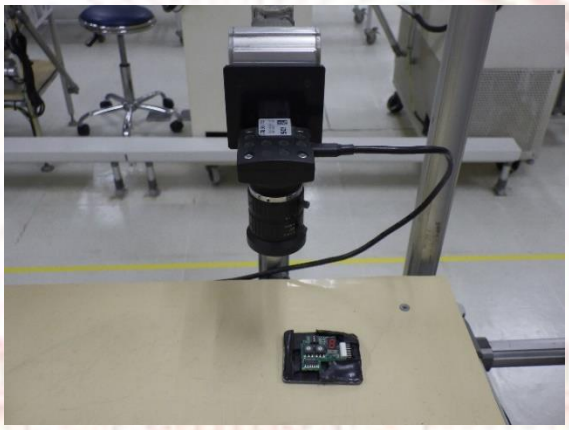
จากบทที่ 2 ซึ่งได้กล่าวถึงแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องที่ทางผู้วิจัยได้ทำการศึกษาเพื่อใช้สำหรับการดำเนินการวิจัยพัฒนาระบบปัญญาประดิษฐ์ในการตรวจจับข้อบกพร่องบนชิ้นส่วน ต่อไปนี้จะเรียกว่า “งานเสีย” โดยผู้วิจัยได้นำโมเดล YOLO V11 ที่แตกต่างกัน 2 รุ่นมาทำการพัฒนาเพื่อเปรียบเทียบกัน ได้แก่ Yolov11n และ Yolov11s ซึ่งทั้ง 2 โมเดลได้ถูกดำเนินการในการวิจัยตามขั้นตอนดังต่อไปนี้



ตารางที่ 3-1 แผนผังขั้นตอนการวิจัยพัฒนา

3.1 เก็บภาพตัวอย่างชิ้นงานดี

ผู้วิจัยได้เริ่มต้นจากการถ่ายรูปชิ้นงานดีที่อยู่ในกระบวนการผลิตโดยภายใต้แสงสว่างในพื้นที่กระบวนการผลิตตามปกติและถ่ายภาพด้วยกล้องอุตสาหกรรมเดิมที่มีใช้งานอยู่แล้วภายในโรงงานโดยทำการยึดกล้องกับโครงอลูมิเนียมไปป์ จากนั้นทำการเขียนคำสั่งเพื่อทำการถ่ายภาพ ซึ่งภาพที่ถ่ายได้มีความละเอียดของภาพที่ 640 x 640 pixels เป็นภาพชนิด JPG รวมได้ภาพชิ้นงานดีทั้งสิ้น 500 รูป



ภาพที่ 3-2 การติดตั้งกล้องอุตสาหกรรมเพื่อเก็บภาพตัวอย่างชิ้นงานดี

3.2 จำลองภาพชิ้นงานเสีย

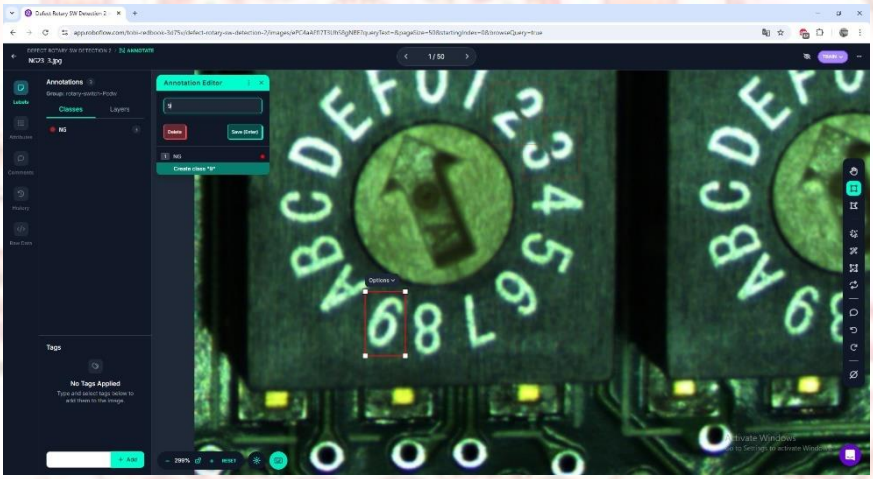
เนื่องจากในกระบวนการผลิตไม่มีงานเสีย ทางผู้วิจัยจึงได้ทำการนำภาพงานดีที่ได้ถ่ายภาพเก็บไว้ในขั้นตอนก่อนหน้านี้มาทำการตัดต่อด้วยโปรแกรม GIMP โดยใช้เครื่องมือ Clone Tool เพื่อเป็นการจำลองภาพงานเสีย โดยในภาพที่ 3-3 คือรูปที่ทำการตัดต่อบางส่วนของเลขศูนย์ให้ขาดไป

ภาพที่ 3-3 การใช้โปรแกรม GIMP ในการตัดต่อภาพเพื่อจำลองภาพชิ้นงานเสีย

3.3 สร้างป้ายกำกับ

การดำเนินการสร้างป้ายกำกับ (Label) หรือ คำอธิบาย (Annotation) ได้ใช้งานเว็บไซต์ Roboflow.com ซึ่งผู้วิจัยสามารถอัพโหลดรูปที่ต้องการสร้างกล่องขอบเขต (bounding box) ให้กับตัวอักขระที่ต้องการให้ตรวจจับได้แบบออนไลน์โดยไม่ต้องติดตั้งซอฟต์แวร์ และสามารถที่จะดำเนินการในขั้นตอนนี้จากที่ใดก็ได้เนื่องจากข้อมูลอยู่บนคลาวด์ของเว็บไซต์นี้ ซึ่งในการสร้างป้าย-กำกับในการวิจัยนี้ได้แบ่งเป็น 2 แนวทาง ได้แก่

1) การสร้าง class สำหรับงานเสียเพียงแค่ class เดียว คือ NG
2) การแบ่ง class ตามจำนวนอักขระ, ป้ายระบุงานเสียและขอบเขตของตัวสวิตช์ ซึ่งทำให้มี class ทั้งหมด 18 class ได้แก่ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D , E, F, NG และ Rotary Switch



ภาพที่ 3-4 การสร้างป้ายกำกับบนเว็บไซต์ roboflow.com

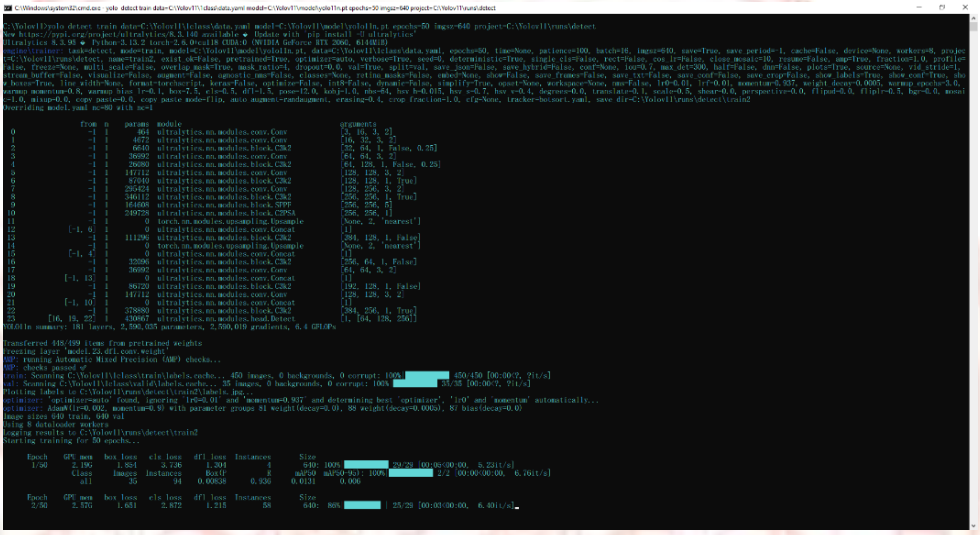
ในภาพที่ 3-4 เป็นตัวอย่างในการสร้างป้ายกำกับที่หมายเลข 9 นอกจากนี้จะเป็นการสร้าง class หมายเลข 9 ขึ้นมาอีกด้วย โดยในการวิจัยนี้มีการศึกษาด้วยว่าการแบ่ง class แบบใดที่ให้ความแม่นยำในการตรวจจับมากกว่ากัน ซึ่งจะมีการอธิบายผลการทดลองในส่วนนี้ในบทที่ 4 ต่อไป

3.4 ฝึกสอนปัญญาประดิษฐ์

ผู้วิจัยได้ใช้งานคอมพิวเตอร์ที่มีการติดตั้งการ์ดประมวลจอภาพรุ่น Nvidia Geforce RTX2060 เพื่อใช้งานซอฟต์แวร์ CUDA โดยในการวิจัยนี้นอกเหนือจากโมเดลที่ใช้ฝึกสอน และจำนวนป้ายกำกับที่แตกต่างกันแล้ว ยังมีอีกหนึ่งพารามิเตอร์ที่แตกต่างกันและทำการศึกษาว่าจะส่งผลต่อประสิทธิภาพ

ในการตรวจจับหรือไม่ นั่นคือ จำนวนครั้งในการฝึกสอนข้อมูลทั้งหมด (epochs) ที่แตกต่างกันอีกด้วย ซึ่งในการวิจัยนี้จะแบ่งจำนวนครั้งในการฝึกสอนเป็นขั้นๆ จำนวน 4 ขั้น ดังต่อไปนี้

1) epochs = 50
2) epochs = 100
3) epochs = 150
4) epochs = 200



ภาพที่ 3-5 หน้าต่างแสดงความคืบหน้าในการฝึกสอนโมเดล

3.5 การทดสอบ

3.5.1 การประเมินประสิทธิภาพของแบบจำลอง จากการศึกษางานวิจัยต่างๆ ที่มีผู้วิจัยท่านอื่นๆ ที่ได้ทำการศึกษามาก่อนหน้างานวิจัยนี้พบว่าสามารถประเมินประสิทธิภาพได้จาก Confusion Matrix ซึ่งเป็นเครื่องมือหนึ่งที่ใช้ประเมินผลลัพธ์ของการทำนาย (Predict) ของโมเดลปัญญาประดิษฐ์

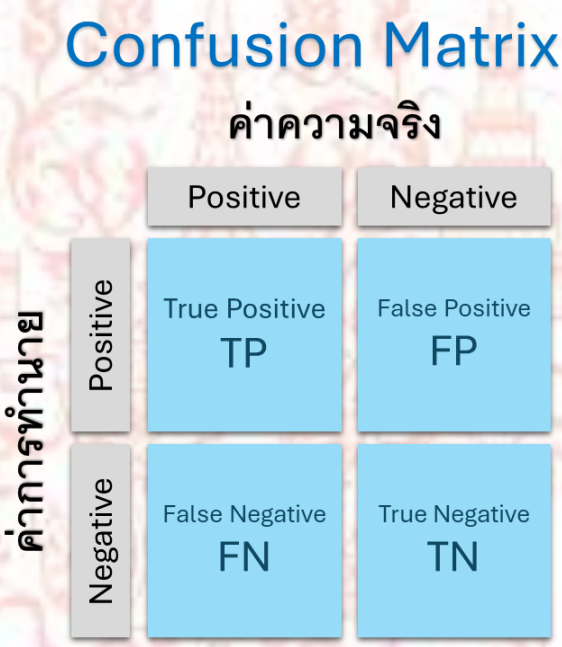
Confusion Matrix ถูกใช้เพื่อตรวจสอบการตรวจจับสิ่งใดสิ่งหนึ่ง โดยจะจำแนกสิ่งที่ตรวจจับได้ออกเป็น class ตามแต่ป้ายกำกับที่ถูกสร้างและได้ทำการฝึกสอนให้กับโมเดลนั้นๆ โดยทั่วไปจะมีลักษณะเป็นตาราง 2x2 โดยแกนตั้งจะเป็นค่าความจริง (Actual Value) และแกนนอนจะเป็นค่าการทำนาย (Predicted Value) ซึ่งแต่ละแกนจะมีค่า 2 ค่า คือ Positive และ Negative และในตารางนี้แต่ละช่องจะมีความหมายดังต่อไปนี้

1) True Positive (TP) ผลลัพธ์ของการทำนายค่าหรือ classถูกต้องตรงกับความจริง ซึ่งทั้งความจริงและคำทำนายมีค่าเป็น Positive

2) True Negative (TN) ผลลัพธ์ของการทำนายค่าหรือ classถูกต้องตรงกับความจริง แต่ความจริงของทั้งความจริงและคำทำนาย คือค่าเป็น Negative

3) False Positive (FP) ผลลัพธ์ของการทำนายค่าหรือ classไม่ตรงกับความจริง เนื่องจากความจริงมีค่าเป็น Negative แต่คำทำนายมีค่าเป็น Positive ซึ่งสามารถเรียกได้อีกอย่างว่า “ความผิดพลาดประเภทที่1” (Type I Error)

4) False Negative (FN) ผลลัพธ์ของการทำนายค่าหรือ classไม่ตรงกับความจริง เนื่องจากความจริงมีค่าเป็น Positive แต่คำทำนายมีค่าเป็น Negative “ความผิดพลาดประเภทที่ 2” (Type II Error)



ภาพที่ 3-6 ตาราง Confusion Matrix

สามารถยกตัวอย่างการประยุกต์ใช้ตาราง Confusion Matrix ซึ่งในการฝึกสอนโมเดลปัญญาประดิษฐ์ได้ดังตารางในภาพที่ 3-7 เป็นการฝึกสอนให้จำแนกว่าภาพใดคือแมวและภาพใดไม่ใช่แมว โดยกำหนดให้ค่า Positive คือ แมว และค่า Negative คือ ไม่ใช่แมว

จากตารางในภาพที่ 3-7 จะอธิบายได้ดังต่อไปนี้

1. ผลลัพธ์ช่องบนซ้ายคือ TP เนื่องจากคำทำนายคือ “แมว” และความจริงคือ “แมว”
2. ผลลัพธ์ช่องบนขวาคือ FP เนื่องจากคำทำนายคือ “แมว” แต่ความจริงคือ “ไม่ใช่แมว”
3. ผลลัพธ์ช่องล่างซ้ายคือ FN เนื่องจากคำทำนายคือ “ไม่ใช่แมว” แต่ความจริงคือ “แมว”

4. ผลลัพธ์ช่องล่างขวาคือ TN เนื่องจากคำทำนายคือ “ไม่ใช่แมว” และความจริงก็ “ไม่ใช่แมว”

ภาพที่ 3-7 การประยุกต์ใช้ Confusion Matrix กับการแยกแยะรูปแมวและที่ไม่ใช่แมว

และจาก Confusion Matrix นี้ เมื่อทำการนับจำนวนครั้งที่เป็น TP, FP, TN, FN โดยสมมติว่าทำการทำนายจำนวน 20 ครั้ง แล้วได้ผลการทำนายตามภาพที่ 3-8

		Actual	
		Positive	Negative
Predicted	Positive	TP = 8	FP = 4
	Negative	FN = 2	TN = 6

ภาพที่ 3-8 ตาราง Confusion Matrix เมื่อนับผลลัพธ์ของการทำนายในแต่ละประเภท

เราจะสามารถคำนวณหาค่าความถูกต้อง, ค่าความเที่ยงตรง, ค่า Recall ได้จากวิธีคำนวณต่อไปนี้

3.1.1.1 ค่าความแม่นยำ (Accuracy) เป็นการเปรียบเทียบจำนวนครั้งที่โมเดลสามารถทำนายผลได้ถูกต้องกับจำนวนการทำนายทั้งหมด คำนวณได้จากการสูตรดังต่อไปนี้

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3\text{-}1)$$

$$Accuracy = \frac{8+6}{8+6+4+2}$$

$$Accuracy = \frac{14}{20}$$

$$Accuracy = 0.7$$

โดยค่าความแม่นยำที่คาดหวังจากการทำนายของโมเดลที่ทำการฝึกสอนในงานวิจัยนี้คือ 0.9 ขึ้นไปซึ่งได้จากการทดสอบทำโดยการจำลองภาพงานเสียขึ้นมาอีกจำนวน 32 ภาพ ซึ่งมากจากจำนวนของอักขระบน 16-position rotary switch ซึ่งบนชิ้นงาน 1 ชิ้นจะประกอบไปด้วย rotary switch จำนวน 2 ตำแหน่ง โดยผู้วิจัยได้ทำการทดสอบโดยให้โมเดลที่ได้ถูกฝึกสอนให้จดจำลักษณะงานเสียมาแล้วนั้น ลองทำนายว่าภาพงานเสียที่ป้อนเข้าไปนั้น มีข้อบกพร่องที่ตัวอักขระใด ถูกต้องตามที่กำหนดไว้หรือไม่ แล้วคำนวณหาค่าความแม่นยำออกมา

3.1.1.2 ค่าความเที่ยงตรง (Precision) คือ ค่าความแม่นยำที่เมื่อเปรียบเทียบการทำนายว่าจริงและผลการทำนายคือจริง (True Positive) กับการทำนายว่าจริงและผลการทำนายก็คือจริง (True Positive) รวมกับการทำนายว่าจริงแต่ผลการทำนายคือไม่จริง (False Positive) คิดจากสูตรดังต่อไปนี้

$$Precision = \frac{TP}{TP+FP} \quad (3\text{-}2)$$

จากตัวอย่างผลลัพธ์ที่ได้จากภาพที่ 3-7 จะได้ค่า Precision เท่ากับ 0.67 ได้จากการคำนวณดังต่อไปนี้

$$Precision = \frac{8}{8+4}$$

$$Precision = 0.67$$

ค่าความเที่ยงตรงนี้ยิ่งมีค่าใกล้ 1 มากเท่าใดก็หมายความว่ามีความเที่ยงตรงมากขึ้นเท่านั้น จึงเป็นตัวสะท้อนประสิทธิภาพของโมเดลได้ค่าหนึ่ง ซึ่งค่าความเที่ยงตรงของโมเดลที่คาดหวังสำหรับการวิจัยนี้คือ 0.9 ขึ้นไป

3.1.1.3 ค่าการเรียกคืน (Recall) คือค่าการจดจำการตรวจจับวัตถุในภาพที่เราสร้างในแบบจำลองขึ้นมา และจะยิ่งความแม่นยำมากขึ้นเมื่อมีค่าเข้าใกล้ 1 โดยจะสามารถคำนวณได้จากสูตรดังต่อไปนี้

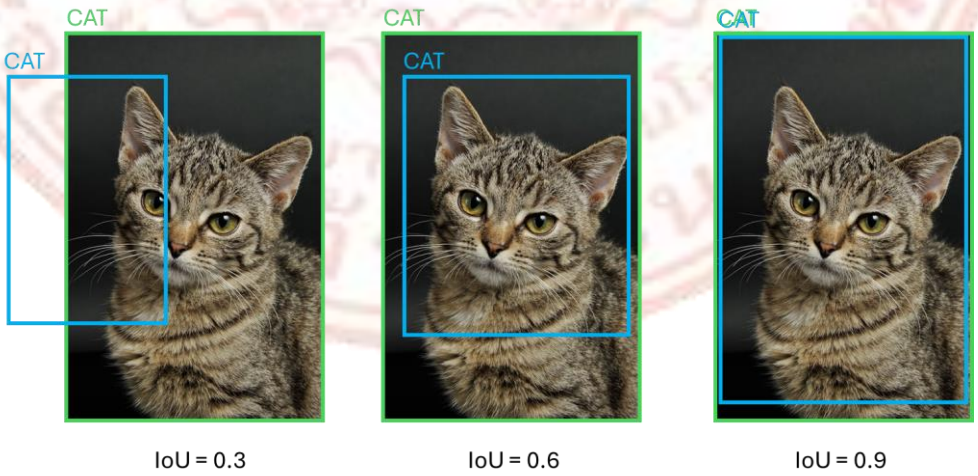
$$Recall = \frac{TP}{TP+FN} \tag{3-3}$$

จากตัวอย่างผลลัพธ์ที่ได้จากภาพที่ 3-7 จะได้ค่า Recall เท่ากับ 0.8 ได้จากการคำนวณดังต่อไปนี้

$$Recall = \frac{8}{8+2}$$

โดยคาดหวังค่าการเรียกคืนที่ 0.9 ขึ้นไปจึงจะถือว่ามีประสิทธิภาพเป็นที่น่าพอใจ

3.1.1.4 mAP50 คือค่าเฉลี่ยความเที่ยงตรงของแต่ละ class โดย mAP50 (Mean average precision at IOU over 0.5) เป็นผลลัพธ์จากการที่โมเดลทำนายทำการตีกรอบ (bounding box) แล้วเปรียบเทียบกับกรอบเฉลย (ground truth) ด้วยค่า Intersect over Union (IoU) = 0.5 ขึ้นไป



ภาพที่ 3-9 ตัวอย่างของ Intersect over Union (IoU)

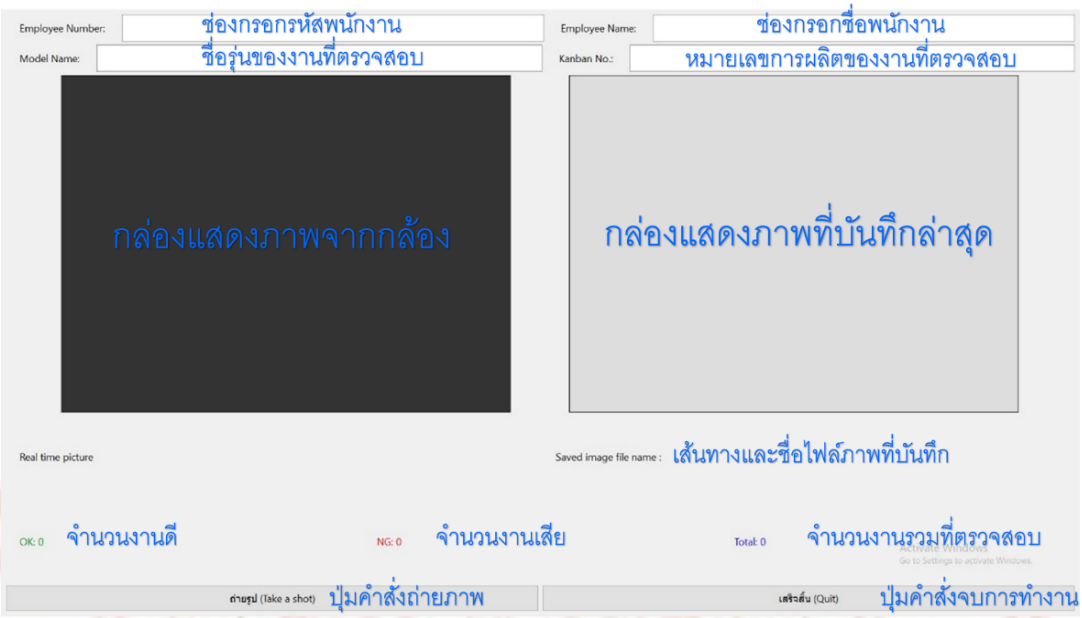
จากภาพที่ 3-9 กรอบที่เขียวคือ Ground Truth ซึ่งกำหนดพิกัดของรูปแมวเอาไว้ ส่วนกรอบสีฟ้าคือส่วนที่โมเดลทำการทำนายและตีกรอบขึ้นมาที่เรียกว่า Bounding Box ซึ่งเมื่อนำสองส่วนนี้มาเปรียบเทียบอัตราส่วนระหว่างกันจะได้ค่า IoU ออกมา สามารถอธิบายได้ว่ายิ่งกรอบสีฟ้าที่โมเดลตีขึ้นมาใกล้เคียงกับกรอบสีเขียวมากเท่าใด ค่า IoU ก็จะมีค่ามากขึ้นเท่านั้น โดยมีสูตรการคำนวณดังต่อไปนี้

$$IoU = \frac{Intersection\ Area}{Union\ Area} \tag{3-4}$$

เมื่อนำ IoU ของแต่ละภาพที่ได้มาหาค่าเฉลี่ยจะได้ค่า mAP50 ออกมา ซึ่งหากว่าค่า mAP50 มากกว่า 0.5 ก็ขึ้นไปจะถือว่าการทำนายของโมเดลมีความถูกต้อง และความถูกต้องจะมากขึ้นเมื่อค่าเข้าใกล้ 1 โดยการวิจัยนี้คาดหวังค่า mAP50 ให้มีค่าตั้งแต่ 0.9 ขึ้นไปจะถือว่าโมเดลนั้นมีประสิทธิภาพในการตรวจจับเป็นที่น่าพอใจ อย่างไรก็ตามในกรณีที่มีค่าใดค่าหนึ่งไม่ถึงเกณฑ์ที่กำหนดไว้ ผู้วิจัยจะต้องทำการเพิ่มจำนวนครั้งของการฝึกสอนขึ้นเป็นลำดับตามที่ระบุไว้ในข้อ 3.4 และทำการทดสอบใหม่อีกครั้ง แต่หากมีการเพิ่มจำนวนครั้งของการฝึกสอนจนครบทุกลำดับตามที่ระบุไว้ในข้อ 3.4 แล้ว ยังไม่ได้ค่าใดค่าหนึ่งตามเกณฑ์ข้างต้น ก็ให้ถือว่า ณ ระดับจำนวนครั้งการฝึกสอนใดที่ให้ค่า Precision, Recall และ mAP50 สูงที่สุดเป็นเงื่อนไขที่ดีที่สุดสำหรับการฝึกสอนโมเดล

3.6 การสร้างส่วนติดต่อผู้ใช้งาน (User Interface)

เป็นการออกแบบหน้าต่างโปรแกรมแล้วเขียนขึ้นด้วยภาษา Python โดยมีการเรียกใช้งานไลบรารีต่างๆ เช่น OpenCV, PyQt6 และ YOLO ซึ่งได้นำโมเดลที่ได้ฝึกสอนไว้แล้วในขั้นตอนที่ 3.4 มาใช้งาน ส่วนติดต่อผู้ใช้งานนี้สร้างขึ้นมาในรูปแบบของ Graphic User Interface (GUI) เพื่อให้พนักงานสามารถใช้งานได้ง่ายโดยไม่จำเป็นต้องมีความรู้ทางด้านการเขียนคำสั่งของอัลกอริทึม YOLO ก็สามารถที่จะใช้งานโมเดลตรวจจับข้อบกพร่องบนชิ้นงานได้ โดยได้ออกแบบให้มีหน้าต่างการทำงานเพียงหน่าต่างเดียว มีลักษณะหน้าตาโปรแกรมดังภาพที่ 3-10



ภาพที่ 3-10 หน้าต่างส่วนติดต่อผู้ใช้งานแบบ Graphic User Interface (GUI)

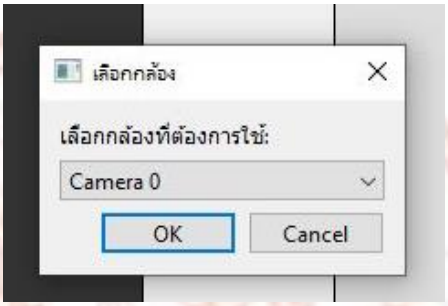
3.6.1 ส่วนประกอบของส่วนติดต่อผู้ใช้งานประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

1. ช่องกรอกรหัสพนักงาน
2. ช่องกรอกชื่อพนักงาน
3. ชื่อรุ่นของงานที่ตรวจสอบ
4. หมายเลขการผลิตของงานที่ตรวจสอบ
5. กล่องแสดงภาพจากกล้อง
6. กล่องแสดงภาพที่บันทึกล่าสุด
7. เส้นทางและชื่อไฟล์ภาพที่บันทึก
8. จำนวนงานดี
9. จำนวนงานเสีย
10. จำนวนงานรวมที่ตรวจสอบ
11. ปุ่มคำสั่งถ่ายภาพ
12. ปุ่มคำสั่งจบการทำงาน

3.6.2 การทำงานของส่วนติดต่อผู้ใช้งาน

3.6.2.1 เมื่อทำการเปิดโปรแกรมจะมีหน้าต่างเล็กดังภาพที่ 3-11 ขึ้นมาที่กลางหน้าจอ

เพื่อให้ผู้ใช้งานเลือกกล้องที่เชื่อมต่อ ส่วนนี้มีไว้ในกรณีมีกล้องที่เชื่อมต่ออยู่กับคอมพิวเตอร์มากกว่า 1 ชุดขึ้นไป เมื่อผู้ใช้งานคลิกเลือกกล้อง แล้วคลิกที่ปุ่ม OK ก็จะเปลี่ยนการเลือกใช้งานกล้องตัวนั้น แต่หากคลิกที่ปุ่ม Cancel ระบบจะเลือกกล้องตัวแรกที่พบให้โดยอัตโนมัติ



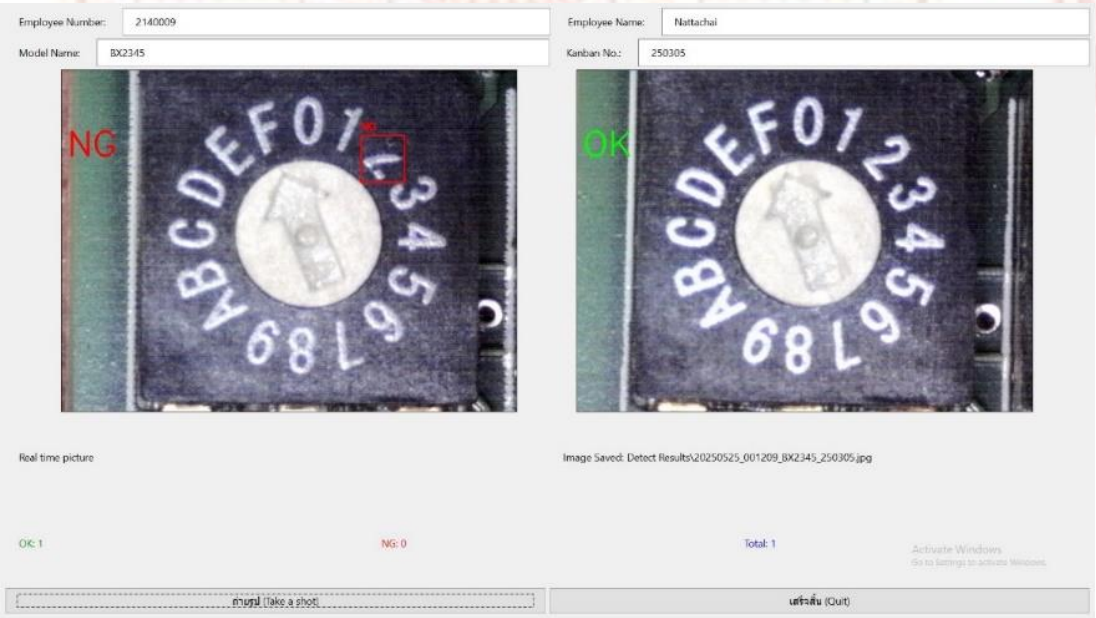
ภาพที่ 3-11 หน้าต่างเลือกใช้งานกล้อง

กรอกข้อมูลรหัสพนักงาน, ชื่อพนักงาน, ชื่อรุ่นของงานที่ตรวจสอบ, หมายเลขการผลิตของงานที่ตรวจสอบ จากนั้นนำชิ้นงานมาวางในตำแหน่งที่กล้องถูกตั้งไว้ ภาพจากกล้องจะถูกแสดงที่กล่องแสดงภาพจากกล้อง (กรอบทางด้านซ้าย) ซึ่งหากตรวจไม่พบตัวอักขระที่ไม่สมบูรณ์จะมีคำว่า “OK” เป็นตัวหนังสือสีเขียวขึ้นทับทางด้านซ้ายบนของภาพในกล่องแสดงภาพดังภาพที่ 3-12

ภาพที่ 3-12 ตัวอย่างเมื่อชิ้นงานที่นำมาตรวจสอบเป็นงานดี

3.6.2.2 คลิกที่ปุ่มคำสั่งถ่ายภาพเพื่อทำการบันทึกภาพการตรวจสอบนั้น โดยภาพที่ถูกบันทึกจะแสดงขึ้นในกล่องแสดงภาพที่บันทึก (กรอบทางด้านขวา) และที่เส้นทางและชื่อไฟล์ภาพที่บันทึกจะแสดงที่อยู่ของไฟล์ภาพที่ทำการบันทึก ซึ่งโดยปกติจะถูกบันทึกในโฟลเดอร์ที่ชื่อว่า Detect Results และไฟล์ภาพที่บันทึกจะถูกตั้งชื่อโดยอัตโนมัติตามรูปแบบที่ถูกกำหนดไว้ คือ yyyymmdd_hhmmss_Model Name_Lot Number.jpg ตัวอย่างเช่นไฟล์ที่ชื่อว่า 20250325_101209_BX2345_250305.jpg หมายถึงงานที่ตรวจสอบในวันที่ 25 เดือน มีนาคม 2025 เวลา 10 โมง 12 นาที 9 วินาที โดยเป็นงานรุ่น BX2345 หมายเลขการผลิต 250305 เป็นต้น

3.6.2.3 กรณีงานที่ตรวจสอบเป็นที่มีข้อบกพร่องตัวอักขระที่ไม่สมบูรณ์ จะมีกรอบสี่เหลี่ยมสีแดงล้อมตัวอักขระนั้น และจะมีคำว่า “NG” เป็นตัวหนังสือสีแดงขึ้นทับทางด้านซ้ายบนของภาพในกล่องแสดงภาพนั้นดังภาพที่ 3-13 การบันทึกภาพงานเสียทำเช่นเดียวกับขั้นตอนในข้อ 3.6.2.1



ภาพที่ 3-13 ตัวอย่างเมื่อชิ้นงานที่นำมาตรวจสอบเป็นงานเสีย

3.6.2.4 นอกจากไฟล์ภาพที่ทำการบันทึกแล้วภายในโฟลเดอร์ Detect Results ยังประกอบไปด้วยไฟล์ Text ซึ่งจะถูกเขียนขึ้นตามวันที่ทำการตรวจสอบงาน เช่น ไฟล์ชื่อ 20250325 จะหมายความว่าเป็นไฟล์ที่ถูกสร้างขึ้นในวันที่ 25 มีนาคม 2025 เป็นต้น

ภาพที่ 3-14 ตัวอย่างไฟล์ที่อยู่ภายในโฟล์เดอร์ Detect Results

3.6.2.5 ข้อมูลที่อยู่ในไฟล์ Text จะประกอบไปด้วย ชื่อไฟล์ภาพ, ชื่อพนักงาน และรหัส-พนักงาน ซึ่งไฟล์นี้ถูกเขียนขึ้นเพื่อใช้ในการสอบกลับ (Traceability) กรณีที่มีงานเสียหลุดรอดไปถึงลูกค้า ก็สามารถที่กลับมาค้นดูข้อมูลของการตรวจสอบในวันที่ทำการตรวจสอบได้ หรือสามารถค้นหาจากหมายเลขผลิตก็ได้

ภาพที่ 3-15 ตัวอย่างข้อมูลที่อยู่ในไฟล์ text

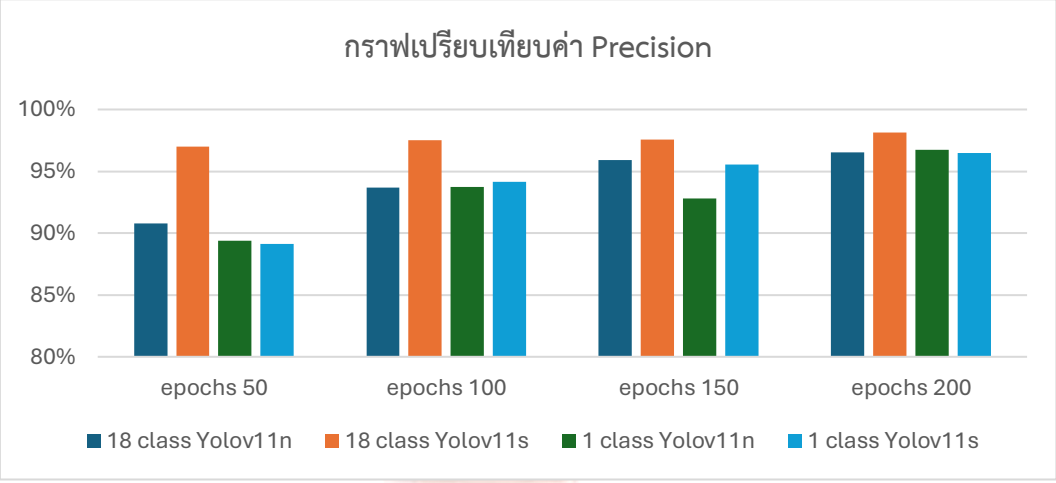
3.6.2.6 เมื่อทำการตรวจสอบเสร็จสิ้นพนักงานสามารถคลิกปุ่มคำสั่งจบการทำงานหน้าต่างโปรแกรมก็จะถูกปิดลงไป

บทที่ 4
ผลการวิจัย

จากการวางระเบียบขั้นตอนในการแบบจำลองพัฒนาปัญญาประดิษฐ์ที่ได้กล่าวไว้ในบทที่ 3 ซึ่งใช้โมเดลในการตรวจจับ 2 รุ่นได้แก่ Yolov11n และ Yolov11s โดยในแต่ละโมเดลจะมีการฝึกสอนปัญญาประดิษฐ์ด้วยจำนวนรอบที่ต่างกันเป็นลำดับๆ ได้แก่ 50, 100, 150 และ 200 รอบ และนอกจากนี้ยังใช้การแบ่ง Class ที่แตกต่างกันคือ 18 Class และ 1 Class ทำให้ได้ผลการในการวิจัยดังต่อไปนี้

4.1 เปรียบเทียบประสิทธิภาพด้วยค่าเปรียบเทียบด้วยค่า Precision

จากการฝึกสอนด้วยเงื่อนไขที่กำหนดไว้ทำให้ได้ค่า Precision ออกมาดังภาพที่ 4-1โดยจะเห็นได้ว่าการฝึกสอนด้วยโมเดล Yolov11s แบ่ง class เป็น 18 class และฝึกสอนจำนวน 200 รอบ จะได้ค่า Precision สูงที่สุดอยู่ที่ 98.1% รองลงมาคือ โมเดล Yolov11n แบ่ง class เป็น 1 class และฝึกสอนจำนวน 200 รอบ, โมเดล Yolov11n แบ่ง class เป็น 18 class และฝึกสอนจำนวน 200 รอบ และโมเดล Yolov11s แบ่ง class เป็น 1 class และฝึกสอนจำนวน 200 รอบ ตามลำดับ

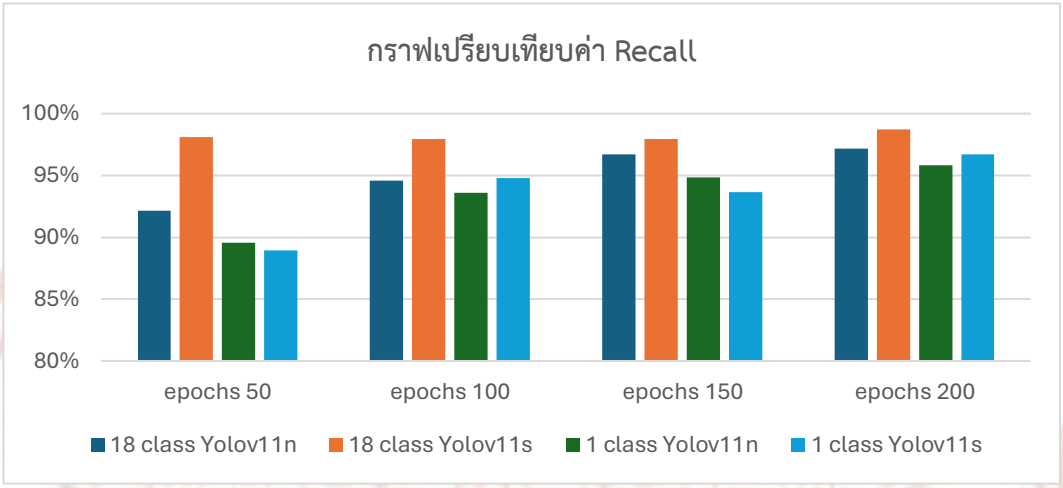


ภาพที่ 4-1 กราฟแท่งเปรียบเทียบค่า Precision

4.2 เปรียบเทียบประสิทธิภาพด้วยค่าเปรียบเทียบด้วยค่า Recall

การฝึกสอนโมเดลที่ได้ค่า Recall สูงที่สุด ได้แก่ การฝึกสอนด้วยโมเดล Yolov11s แบ่ง class เป็น 18 class และฝึกสอนจำนวน 200 รอบ โดยมีค่า Recall อยู่ที่ 98.7% รองลงมานั้น คือ โมเดล

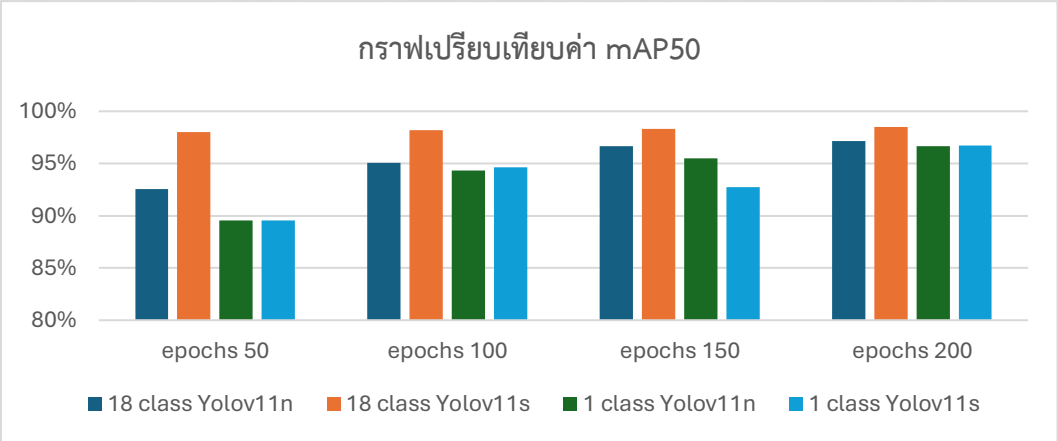
Yolov11n แบ่ง class เป็น 18 class และฝึกสอนจำนวน 200 รอบ, โมเดล Yolov11s แบ่ง class เป็น 1 class และฝึกสอนจำนวน 200 รอบ และโมเดล Yolov11n แบ่ง class เป็น 1 class และฝึกสอนจำนวน 200 รอบ ตามลำดับ



ภาพที่ 4-2 กราฟแท่งเปรียบเทียบค่า Recall

4.3 เปรียบเทียบประสิทธิภาพด้วยค่าเปรียบเทียบด้วยค่า mAP50

โมเดลที่มีค่า mAP50 ดีที่สุดคือ Yolov11s แบ่ง class เป็น 18 class และฝึกสอนจำนวน 200 รอบ ที่ 98.5% รองลงมาคือ โมเดล Yolov11n แบ่ง class เป็น 18 class ฝึกสอน 200 รอบ และทั้งโมเดล Yolov11s และ Yolo11n ที่แบ่ง class เป็น 1 class ฝึกสอน 200 รอบ ได้ค่า mAP50 เท่ากัน

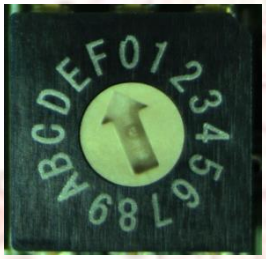
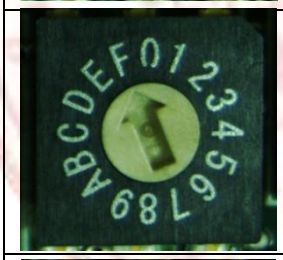
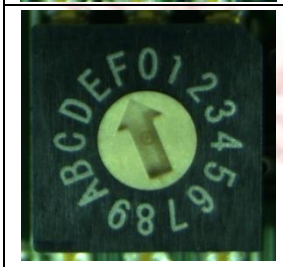
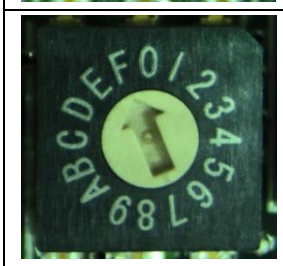


ภาพที่ 4-3 กราฟแท่งเปรียบเทียบค่า mAP50

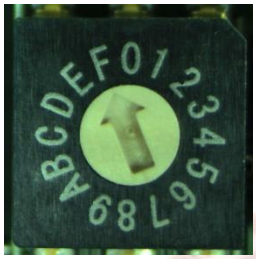
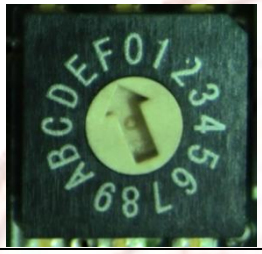
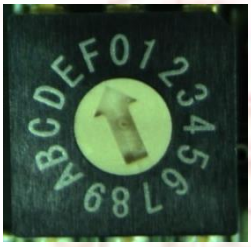
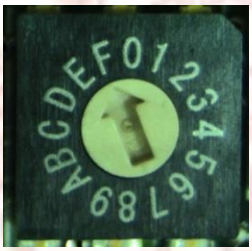
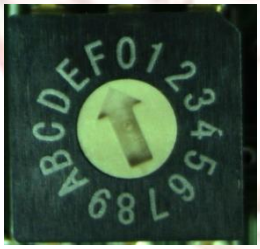
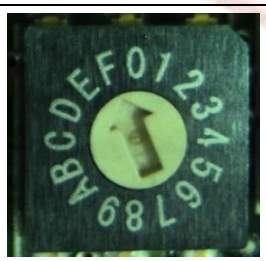
4.4 ประสิทธิภาพความแม่นยำจากการทดสอบ

จากผลการฝึกสอนข้างต้น ผู้วิจัยจึงทำการนำโมเดลที่ถูกฝึกสอนจำนวน 200 รอบของโมเดล Yolo ทั้ง 2 รุ่น (Yolov11n และYolov11s) และทั้ง 2 class มาทำการทดสอบว่าโมเดลเหล่านี้จะสามารถตรวจจับข้อบกพร่องในด้านอักขระที่ไม่สมบูรณ์ได้อย่างแม่นยำเพียงใด โดยทำการทดสอบการตรวจจับภาพงานเสียที่จำลองขึ้นจำนวน 32 รูป แบ่งได้เป็นภาพงานเสียที่ตำแหน่ง SW1 จำนวน 16 รูป และตำแหน่ง SW2 อีกจำนวน 16 รูป ได้ผลการทดสอบดังต่อไปนี้

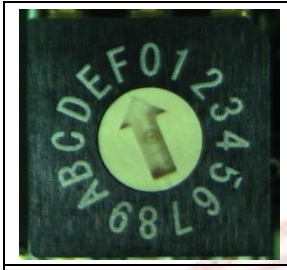
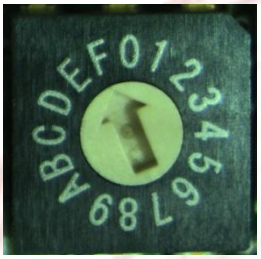
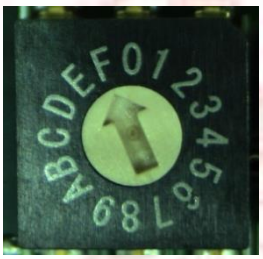
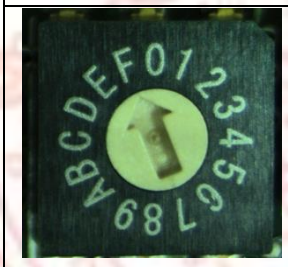
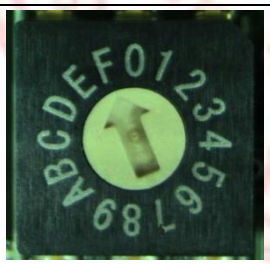
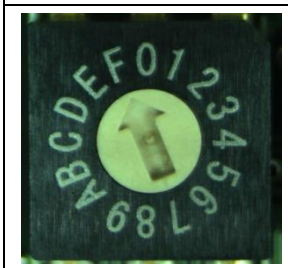
ตารางที่ 4-1 ตารางแสดงผลการตรวจจับด้วยโมเดลที่ฝึกสอนที่แตกต่างกัน

ภาพงานเสียจำลอง	ตำแหน่งและอักขระที่ไม่สมบูรณ์	ผลการตรวจจับ			
		yolov11n 18 class	yolov11s 18 class	yolov11n 1 class	yolov11s 1 class
	SW1 / 0	TP	FP	TP	TP
	SW2 / 0	FP	TP	TP	TP
	SW1 / 1	TP	FP	TP	TP
	SW2 / 1	TP	FN	TP	TP

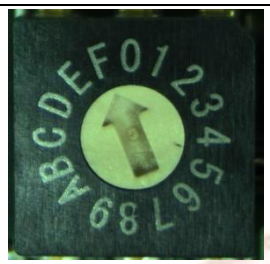
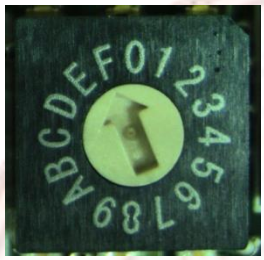
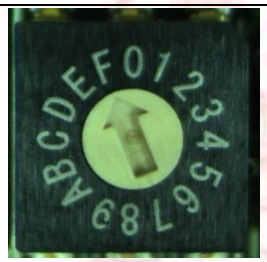
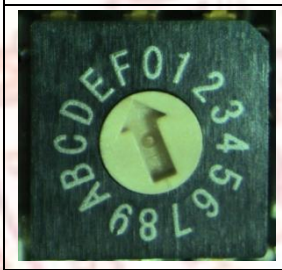
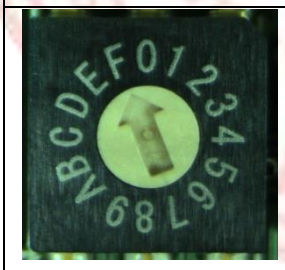
ตารางที่ 4-1 (ต่อ)

	SW1 / 2	TP	TP	TP	TP
	SW2 / 2	TP	FP	TP	FP
	SW1 / 3	FP	FP	TP	TP
	SW2 / 3	FP	FP	TP	TP
	SW1 / 4	TP	TP	TP	TP
	SW2 / 4	FN	FN	FN	TP

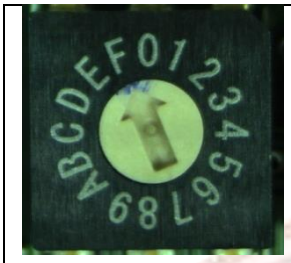
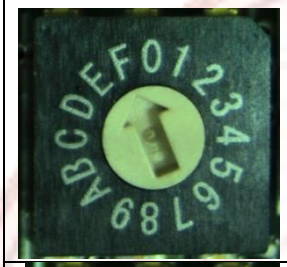
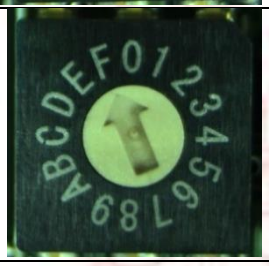
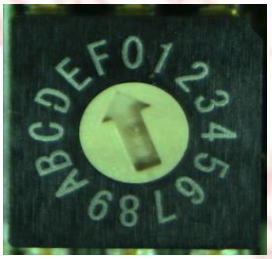
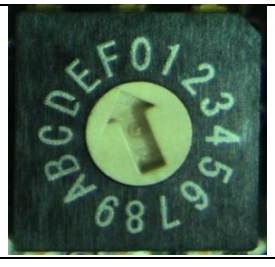
ตารางที่ 4-1 (ต่อ)

	SW1 / 5	TP	FP	TP	TP
	SW2 / 5	FP	FP	TP	TP
	SW1 / 6	FP	TP	TP	TP
	SW2 / 6	FP	FP	TP	TP
	SW1 /7	TP	TP	TP	TP
	SW2 / 7	TP	FP	TP	TP

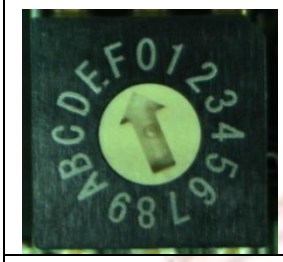
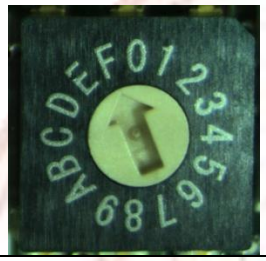
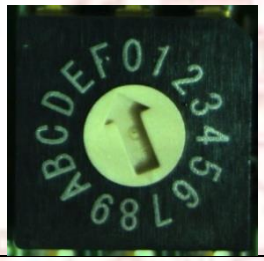
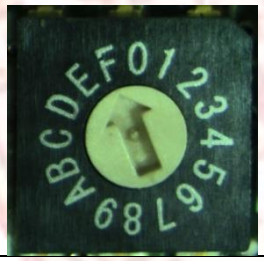
ตารางที่ 4-1 (ต่อ)

	SW1 / 8	FP	TP	TP	TP
	SW2 / 8	FP	FP	TP	TP
	SW1 / 9	FP	TP	TP	FP
	SW2 / 9	TP	TP	TP	TP
	SW1 / A	FN	TP	FP	TP
	SW2 / A	FN	FP	TP	TP

ตารางที่ 4-1 (ต่อ)

	SW1 / B	FN	FP	FP	TP
	SW2 / B	TP	FP	FP	TP
	SW1 / C	FP	TP	TP	TP
	SW2 / C	FP	FP	TP	TP
	SW1 / D	FN	TP	TP	TP
	SW2 / D	TP	TP	TP	TP

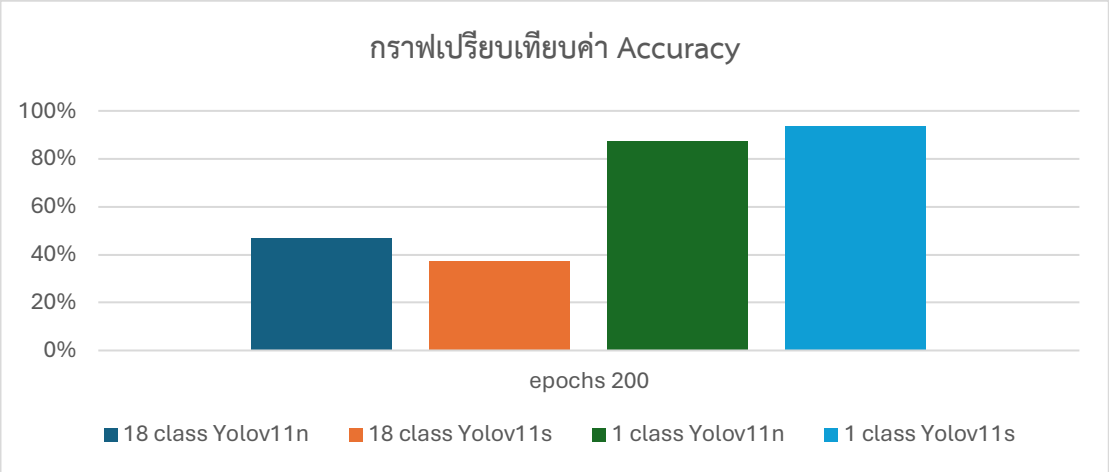
ตารางที่ 4-1 (ต่อ)

	SW1 / E	TP	FP	TP	TP
	SW2 / E	FP	FP	TP	TP
	SW1 / F	TP	FP	TP	TP
	SW2 / F	TP	FP	TP	TP

ตารางที่ 4-2 ตารางแสดงผลรวมค่า TP, TN, FP, FN ที่นับได้และค่า Accuracy ของแต่ละโมเดล

โมเดล	จำนวนclass	TP	TN	FP	FN	ACCURACY
yolov11n	18	15	0	12	5	46.9%
yolov11s	18	12	0	18	2	37.5%
yolov11n	1	28	0	3	1	87.5%
yolov11s	1	30	0	2	0	93.8%

เมื่อนำผลลัพธ์ของการตรวจจับมาสรุปเพื่อคำนวณหาค่า Accuracy จะได้ดังตารางที่ 4-2 ซึ่งจะพบว่าโมเดล Yolov11s ซึ่งมีการแบ่ง class เพียงแค่ 1 class มีประสิทธิภาพในการตรวจจับภาพจำลองงานเสียดีที่สุด มีความแม่นยำอยู่ที่ 93.8% รองลงมาคือ Yolov11n ที่แบ่ง class เป็น 1 class ที่ 87.5% ส่วนโมเดล Yolov11n และ Yolov11s ที่แบ่งเป็น 18 class กลับให้ค่าความแม่นยำต่ำกว่า 50% ทั้ง 2 โมเดล



ภาพที่ 4-4 กราฟแท่งเปรียบเทียบค่า Accuracy

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแบบจำลองปัญญาประดิษฐ์ในการตรวจจับข้อบกพร่องด้านความสมบูรณ์ของตัวอักขระบนชิ้นส่วนอิเล็กทรอนิกส์ประเภท 16-position rotary switch โดยใช้โมเดล YOLOv11 จำนวน 2 โมเดล ได้แก่ YOLOv11n และ YOLOv11s ภายใต้เงื่อนไขการแบ่งคลาสป้ายกำกับแบบ 1 class และ 18 class และทำการฝึกสอนด้วยจำนวนรอบการฝึกสอน (epochs) ที่แตกต่างกัน 4 ลำดับ ได้แก่ 50, 100, 150 และ 200 รอบ ซึ่งผลการวิจัยพบว่า

5.1.1 โมเดล YOLOv11s แบบ 1 class ฝึกสอน 200 รอบ ให้ผลลัพธ์ดีที่สุด โดยมีค่า Precision 97.5%, Recall 98.2% และ mAP50 98.3% เมื่อนำมาทดสอบกับภาพจำลองงานเสีย 32 ภาพ พบว่ามี ค่า Accuracy สูงถึง 93.8%

5.1.2 โมเดล YOLOv11n แบบ 1 class ฝึกสอน 200 รอบ ได้ค่า Accuracy รองลงมาที่ 87.5%, Precision 96.7%, Recall 95.8% และ mAP50 96.7%

5.1.3 ในขณะที่การแบ่ง class แบบ 18 class ให้ผลลัพธ์ต่ำกว่าการแบ่ง class แบบ 1 class เมื่อเปรียบเทียบด้วยค่า Accuracy โดย YOLOv11n และ YOLOv11s สามารถตรวจจับได้แม่นยำเพียง 46.9% และ 37.5% ตามลำดับ

จากผลการทดลองข้างต้นสรุปได้ว่า โมเดล YOLOv11s ซึ่งแบ่งป้ายกำกับ เป็น 1 class และฝึกสอนด้วยจำนวนรอบที่ 200 รอบ เหมาะสมที่สุดสำหรับการใช้งานจริงในการตรวจจับข้อบกพร่องด้านอักขระบนชิ้นส่วนอิเล็กทรอนิกส์ประเภท 16-Position Rotary Switch

ทั้งนี้หากมีการนำโมเดลปัญญาประดิษฐ์นี้ไปใช้ที่กระบวนการผลิตจะส่วนลดต้นทุนในการจัดซื้ออุปกรณ์ตรวจจับภาพของชิ้นงานที่มีข้อบกพร่องได้ราว 300,000 – 400,000 บาท ซึ่งในการวิจัยนี้ได้ใช้งานกล้องอุตสาหกรรมที่มีอยู่ภายในโรงงานอยู่แล้ว ทำให้ไม่เกิดค่าใช้จ่ายในการจัดซื้ออุปกรณ์ใหม่ นอกจากนี้ในด้านการฝึกสอนโมเดลปัญญาประดิษฐ์ก็สามารถใช้คอมพิวเตอร์สำหรับงานออกแบบ 3 มิติ ที่ทางโรงงานใช้งานอยู่ได้เช่นกัน เนื่องจากมีการ์ดประมวลผลจอภาพของ Nvidia ซึ่งสามารถใช้งานคุณสมบัติของซอฟต์แวร์ CUDA ได้ ซึ่งจะส่งผลให้การฝึกสอนทำได้รวดเร็วกว่าคอมพิวเตอร์สำนักงานทั่วไป การพัฒนาโมเดลปัญญาประดิษฐ์นี้จึงมาความคุ้มค่าอย่างยิ่ง

5.2 ปัญหาอุปสรรคและข้อจำกัดของงานวิจัย

5.2.1 จำนวนงานเสียจริงมีจำนวนน้อย ทำให้จำเป็นต้องสร้างภาพจำลองข้อบกพร่องขึ้นมา ซึ่งอาจไม่สามารถครอบคลุมลักษณะข้อบกพร่องที่เกิดขึ้นจริงในกระบวนการผลิตได้ทั้งหมด ส่งผลต่อความหลากหลายของข้อมูลฝึกสอนโมเดล

5.2.2 การใช้โมเดลที่มีประสิทธิภาพสูงขึ้น เช่น Yolov11s ต้องการทรัพยากรคอมพิวเตอร์ที่มีประสิทธิภาพสูง เช่น การ์ดจอ (GPU) ที่รองรับ CUDA และหน่วยความจำสูง และเมื่อเพิ่มจำนวนภาพหรือจำนวนครั้งในการฝึกอาจจะพบปัญหา เช่น การเกิดข้อผิดพลาดระหว่างการฝึกสอนได้ ทำให้ต้องมีการฝึกสอนซ้ำหลายรอบ

5.2.3 การทดสอบภาพจำกัดเฉพาะในสภาพแวดล้อมจำลอง ยังไม่ได้ทดสอบกับภาพจากสายการผลิตจริงที่มีแสง เงา หรือความไม่สม่ำเสมอของพื้นผิว ซึ่งอาจส่งผลต่อความแม่นยำของโมเดลเมื่อนำไปใช้งานจริงในอนาคต

5.3 ข้อเสนอแนะ

5.3.1 ควรขยายการศึกษาสู่การตรวจจับข้อบกพร่องประเภทอื่น ๆ เช่น การบัดกรีผิดพลาด หรือการประกอบไม่สมบูรณ์ เป็นต้น

5.3.2 ศึกษาเปรียบเทียบโมเดลอื่นเพิ่มเติม เช่น YOLOv11m, YOLOv11l, YOLOv11x เพื่อพัฒนาระบบให้มีประสิทธิภาพยิ่งขึ้น

5.3.3 ทดสอบกับภาพชิ้นงานจากสายการผลิตจริงในสภาพแสงและมุมมองหลากหลาย เพื่อเพิ่มความแม่นยำและความเที่ยงตรงของโมเดลในสถานการณ์จริง

5.3.4 การใช้ฟิลเตอร์ภายใน OpenCV แปลงภาพเป็นสีขาวดำ เพื่อลดการสะท้อนของแสงบนพื้นผิวของตัวอุปกรณ์น่าจะช่วยทำให้ตัวอักขระมีความชัดเจนมากขึ้นในการฝึกสอนโมเดล

5.3.5 สามารถต่อยอดการพัฒนาโดยให้ทำงานร่วมกับระบบออโตเมชั่นได้ เช่น เมื่อตรวจสอบงานเสียให้แขนกลโรบอททำการแยกชิ้นงานนั้นไว้ในพื้นที่งานต้องสงสัยเพื่อรอการยืนยัน

บรรณานุกรม

ภาษาไทย

สิริทัศน์ เลิศตระกูลถาวร. *การพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน*.สารนิพนธ์ วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีไทย-ญี่ปุ่นสาขาวิชาเทคโนโลยีสารสนเทศ, 2563.

ปัญจวรรษ แซ่เฉิน และศิริบวรรัตนกุล ฐิติรัตน์. (2566). “การจำแนกความสุกของทะลายปาล์มสดด้วยการเรียนรู้เชิงลึก.” *วารสารวิจัยและพัฒนา มจธ.*. ปีที่ 46 ฉบับที่ 1 (มกราคม - มีนาคม 2566) : 81-104.

กฤษฎา ไกรสุริยวงศ์ และสายรุ้ง บุบผาพันธ์. “การบริหารจัดการองค์กรด้วยปัญญาประดิษฐ์แห่งโลกเสมือนจริง.” *Journal of Interdisciplinary Innovation Review.* ปีที่ 7 ฉบับที่ 3 (พฤษภาคม – มิถุนายน 2567) : 397-411.

จุฑาทิพย์ จงวนิชย์. “ไทยได้ประโยชน์จากสงครามการค้าจริงหรือ?.” *THAIPUCBLICA*. [วารสารออนไลน์] 2566. [สืบค้นวันที่ 30 มิถุนายน 2567]. จากhttps://thaipublica.org/2023/06/does-thailand-benefit-from-trade-war/

กวิน ชินพงศ และคณะ. “การสกัดคุณลักษณะเอบีซีดีเพื่อประเมินความเสี่ยงโรคผิวหนังเมลาโนมาด้วยการประมวลผลภาพ.” *The Journal of Chulabhorn Royal Academy.* ปีที่ 3 ฉบับที่ 4 (ตุลาคม - ธันวาคม 2564) : 233 - 245.

ปฏิญญา ตันฑวิวัฒน์ และคณะ. “การวิเคราะห์ภาพคนและสัมภาระสำหรับการตรวจจับวัตถุที่ปราศจากเจ้าของ.” *Journal of Engineering and Digital Technology (JEDT).* ปีที่ 9 ฉบับที่ 2 (กรกฎาคม – ธันวาคม 2564) : 49-60.

เวณิกา บุญอาษา, พัชรี จันทร์เพ็ง และศุภโชค สอนศิลป์พงศ์. “ผลการใช้เทคนิคการวินิจฉัยระดับความสามารถทางคณิตศาสตร์ของผู้เรียนผ่านการเรียนรู้ของเครื่อง.” *Journal of Inclusive and Innovative Education.* ปีที่ 7 ฉบับที่ 2 (พฤษภาคม - สิงหาคม 2566) : 61-74.

วรวิทย์ ฝั้นคำอ้าย, ศุภชัย วงศ์นภาวิเศษ, สหวรรษ กิติยะ, และนงนุช เกตุ้ย. “การพัฒนาโปรแกรมจำแนกประเภทหน้ากากและตรวจจับการสวมใส่ ด้วยเทคโนโลยีประมวลผลภาพ.” *Journal of Engineering and Digital Technology (JEDT).* ปีที่ 11 ฉบับที่ 1 (มกราคม - มิถุนายน 2566) : 43-53.

นัฒธี มละสาร. การตรวจจับพื้นผิวรอยขีดข่วนของวัตถุโลหะโดยใช้วิธีการประมวลผล การแบ่งภาพและโครงข่ายประสาทแบบคอนโวลูชัน. การศึกษาค้นคว้าอิสระ วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยสุโขทัยธรรมาธิราช, 2566.

กรณิการ์ มูลโพธิ์. “หุ่นยนต์คูก้าคัดแยกฉลากโดยใช้อัลกอริทึม YOLO และบรรจุภัณฑ์.” *ENGINEERING TRANSACTIONS.* ปีที่ 25 ฉบับที่ 2 (กรกฎาคม – ธันวาคม 2565) : 89-99.

วุฒิชัย วัชรารัตน์ และเจษฎา ตัณฑนุช. “การจำแนกพันธุ์ข้าวจากภาพของเมล็ดข้าวสารด้วยวิธีการตรวจหาวัตถุ.” *Journal of Engineering and Digital Technology (JEDT).* ปีที่ 11 ฉบับที่ 2 (ธันวาคม 2566) : 14-24.

จตุพร ศรีสอาดรักษ์. ระบบตรวจจับการพกอาวุธปืนสั้นด้วย YOLOv8. สารนิพนธ์ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์, 2566.

สำนักงานส่งเสริมเศรษฐกิจดิจิทัล. “Machine Learning สิ่งใกล้ตัวแห่งโลกยุคใหม่.” [ออนไลน์] 2563. [สืบค้นวันที่ 9 กรกฎาคม 2567]. จาก https://www.depa.or.th/th/article-view/article11-2563

ยุวเรศมคฐ์ สิทธิชาญบัญชา. “ปัญญาประดิษฐ์ Artificial intelligence (AI) กับการใช้ประโยชน์ทางการแพทย์และเวชศาสตร์ฉุกเฉิน.” *วารสารการแพทย์ฉุกเฉินแห่งประเทศไทย.* ปีที่ 1 ฉบับที่ 1 (มกราคม-มิถุนายน 2564) : 91-104.

เอกรัตน์ สุขสุคนธ์. “การพัฒนาระบบวิเคราะห์และนับจำนวนผลมะนาวด้วยเทคโนโลยีคอมพิวเตอร์วิทัศน์.” *Journal of Information Science and Technology.* ปีที่ 13 ฉบับที่ 1 (มกราคม - มิถุนายน 2566) : 10-16.

สุวรรณกาญจน์ สุพมาตรา, ศุภกิตต์ สายสุนทร, และอนุสรณ์ เชื้อสามารถ. “การศึกษาการตรวจจับสิ่งเจือปนในผักโขมแช่แข็งด้วยการเรียนรู้เชิงลึก โมเดล YOLOv4.” *สหวิทยาการและความยั่งยืนบริทรรศน์ไทย.* ปีที่ 12 ฉบับที่ 2 (กรกฎาคม - ธันวาคม 2566) : 132-144.

อิสราภรณ์ อมรสวัสดิ์วัฒนา, ศุภฤกษ์ จันทร์ศุภเสน และอลงกรณ์ นมะหุต “เครื่องนับลูกไก่.” *วารสารวิศวกรรมศาสตรและนวัตกรรม.* ปีที่ 16 ฉบับที่ 3 (กรกฎาคม - กันยายน 2566) : 81-93

ภาษาอังกฤษ

Li, Y.-T., Kuo, P., & Guo, J.-I. “Automatic Industry PCB Board DIP Process Defect Detection with Deep Ensemble Method.” *IEEE Transactions on Components, Packaging and Manufacturing Technology*. 11 (February 2021) : 312-323.

OpenCV. “Introduction” [online] 2024. [cited June 30, 2024]. Available from : https://docs.opencv.org/4.11.0/d1/dfb/intro.html

TensorGym. “The Complete History and Evolution of PyTorch | Deep Learning Framework Timeline”, [online] (n.d.). [cited June 30, 2024]. Available from : https://tensorgym.com/blog/pytorch-history

Štaka, Z., Mišić, M., & Tomašević, M. . *CPU vs. GPU: Performance Evaluation of Classical*. 24th International Symposium INFOTEH-JAHORINA. East Sarajevo, Bosnia and Herzegovina, 19-21 March 2025 : 1-6.

ภาคผนวก ก

ประสิทธิภาพการฝึกสอนโมเดลปัญญาประดิษฐ์

ประสิทธิภาพการฝึกสอนโมเดลปัญญาประดิษฐ์

ตารางที่ ก-1 ตารางเปรียบเทียบค่า Precision ของการฝึกสอนที่มีเงื่อนไขแตกต่างกัน

	Yolov11n		Yolov11s	
จำนวนรอบ	**1 class**	**18 class**	**1 class**	**18 class**
50	0.894	0.908	0.891	0.970
100	0.938	0.937	0.942	0.975
150	0.928	0.959	0.955	0.975
200	0.976	0.966	0.965	0.981

ตารางที่ ก-2 ตารางเปรียบเทียบค่า Recall ของการฝึกสอนที่มีเงื่อนไขแตกต่างกัน

	Yolov11n		Yolov11s	
จำนวนรอบ	**1 class**	**18 class**	**1 class**	**18 class**
50	0.896	0.921	0.889	0.981
100	0.936	0.946	0.948	0.979
150	0.948	0.947	0.937	0.979
200	0.958	0.972	0.967	0.987

ตารางที่ ก-3 ตารางเปรียบเทียบค่า mAP50 ของการฝึกสอนที่มีเงื่อนไขแตกต่างกัน

	Yolov11n		Yolov11s	
จำนวนรอบ	**1 class**	**18 class**	**1 class**	**18 class**
50	0.896	0.925	0.895	0.980
100	0.943	0.951	0.946	0.982
150	0.955	0.967	0.927	0.983
200	0.967	0.971	0.967	0.985

ภาคผนวก ข

โค้ดโปรแกรมส่วนติดต่อผู้ใช้งาน (User Interface)

โค้ดโปรแกรมส่วนติดต่อผู้ใช้งาน (User Interface)

เขียนขึ้นด้วยภาษา Python ดังต่อไปนี้

```
import sys
import cv2
import os
from PyQt6.QtWidgets import (
    QApplication, QMainWindow, QPushButton, QLabel, QWidget, QVBoxLayout,
QHBoxLayout, QLineEdit
)
from PyQt6.QtCore import QThread, pyqtSignal, Qt
from PyQt6.QtGui import QImage, QPixmap
from ultralytics import YOLO
from datetime import datetime

# โหลดโมเดล
model = YOLO("best.pt")

# โฟลเดอร์บันทึกรูปผลลัพธ์
output_folder = "Detect Results"
os.makedirs(output_folder, exist_ok=True)

class YOLOThread(QThread):
    image_saved = pyqtSignal(str)
    image_to_show = pyqtSignal(QImage)
    finished = pyqtSignal()
    image_to_show_2 = pyqtSignal(QPixmap)  # สัญญาณใหม่สำหรับแสดงภาพที่สอง

    # Signals for counter updates
    update_ok_count = pyqtSignal(int)
    update_ng_count = pyqtSignal(int)
```

```
update_total_count = pyqtSignal(int)

def __init__(self):
    super().__init__()
    self.cap = self.initialize_camera()
    self.image_counter = 1
    self.running = True
    self.take_picture = False
    self.model_name = ""
    self.kanban_no = ""

    # Initialize counters
    self.ok_count = 0
    self.ng_count = 0

def initialize_camera(self):
    """Check available cameras and allow user to select one."""
    available_cameras = self.get_available_cameras()

    if not available_cameras:
        print(" ✗ ไม่พบกล้องที่ใช้งานได้")
        return None  # No cameras found

    selected_camera = self.select_camera(available_cameras)
    cap = cv2.VideoCapture(selected_camera, cv2.CAP_DSHOW)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 800)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 600)

    if not cap.isOpened():
        print(f" ✗ ไม่สามารถเปิดกล้อง {selected_camera} ได้")
```

```
            return None
        return cap
def get_available_cameras(self):
"""Detect available cameras."""
available_cameras = []
for i in range(10):  # Check first 10 camera indexes
temp_cap = cv2.VideoCapture(i)
if temp_cap.isOpened():
available_cameras.append(i)
temp_cap.release()
return available_cameras

def select_camera(self, available_cameras):
"""Show a selection dialog for the user to pick a camera."""
from PyQt6.QtWidgets import QInputDialog

camera_index, ok = QInputDialog.getItem(
None, "เลือกกล้อง", "เลือกกล้องที่ต้องการใช้:",
[f"Camera {i}" for i in available_cameras], 0, False
)

if ok:
return available_cameras[int(camera_index.split()[-1])]
return available_cameras[0]  # Default to first detected camera



def run(self):
while self.running:
ret, frame = self.cap.read()
if not ret:
```

```
print(" ✗ ไม่สามารถอ่านภาพจากกล้องได้")
frame = self.get_placeholder_image()

frame = cv2.flip(frame, -1)
#frame = cv2.resize(frame, (640, 500))
results = model(frame, stream=True, imgsz=640)

ng_found = False

for result in results:
for box in result.boxes:
x1, y1, x2, y2 = map(int, box.xyxy[0])
label = model.names[int(box.cls[0])]
if label == "NG":
ng_found = True
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 2)
cv2.putText(frame, label, (x1, y1 - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)

# Update counters
if ng_found:
cv2.putText(frame, "NG", (10, 150), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 4)
else:
cv2.putText(frame, "OK", (10, 150), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 255, 0), 4)


if self.take_picture:
if ng_found:
self.ng_count += 1
self.update_ng_count.emit(self.ng_count)
```

```
else:
self.ok_count += 1
self.update_ok_count.emit(self.ok_count)
self.save_image(frame)
self.take_picture = False
# Update total count
total_count = self.ok_count + self.ng_count
self.update_total_count.emit(total_count)

rgb_image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
h, w, ch = rgb_image.shape
bytes_per_line = ch * w
qt_image = QImage(rgb_image.data, w, h, bytes_per_line,
QImage.Format.Format_RGB888)

if qt_image.isNull():
print(" ✕ QImage สร้างไม่สำเร็จ")
else:
self.image_to_show.emit(qt_image)

self.cap.release()
self.finished.emit()

def save_image(self, frame):
# Get current date and time for the filename
current_time = datetime.now().strftime("%Y%m%d_%H%M%S")
filename = f"{current_time}_{self.model_name}_{self.kanban_no}.jpg"
output_path = os.path.join(output_folder, filename)

cv2.imwrite(output_path, frame)
```

```
self.image_saved.emit(f"Image Saved: {output_path}")

log_filename = f"{datetime.now().strftime('%Y%m%d')}.txt"
log_path = os.path.join(output_folder, log_filename)
log_content = f"{filename},{self.employee_name},{self.employee_number}\n"
with open(log_path, "a") as log_file:
log_file.write(log_content)

image_path = os.path.join("Detect Results", filename)
if os.path.exists(image_path):
print(f"✅ แสดงภาพไฟล์: {image_path}")
pixmap = QPixmap(image_path)
self.image_to_show_2.emit(pixmap)  # ส่งสัญญาณให้แสดงภาพใน label_image2
else:
print(f"❌ ไม่พบไฟล์: {image_path}")
self.image_counter += 1

def stop(self):
self.running = False
if self.cap.isOpened():
self.cap.release()  # Release the camera properly

def capture(self):
self.take_picture = True

def get_placeholder_image(self):
"""สร้างภาพสีเทาถ้าไม่สามารถเปิดกล้องได้"""
placeholder = cv2.imread("placeholder.jpg")
if placeholder is None:
```

```
placeholder = cv2.cvtColor(cv2.resize(cv2.imread("gray.jpg"), (720, 560)),
cv2.COLOR_BGR2RGB)
return placeholder

class MainWindow(QMainWindow):
def __init__(self):
super().__init__()
self.setWindowTitle("Rotary Switch Defect Inspection AI V.1.00 - Developed by
R.Nattachai")
self.setGeometry(0, 0, 1600, 780)
self.showFullScreen()  # Set the window to full-screen mode


# Text boxes for input
self.label_employee_number = QLabel("Employee Number:", self)
self.label_employee_number.setStyleSheet("font-size: 18px; padding: 10px 20px;")
self.text_employee_number = QLineEdit(self)
self.text_employee_number.setStyleSheet("font-size: 18px; padding: 10px 20px;")


self.label_employee_name = QLabel("Employee Name:", self)
self.label_employee_name.setStyleSheet("font-size: 18px; padding: 10px 20px;")
self.text_employee_name = QLineEdit(self)
self.text_employee_name.setStyleSheet("font-size: 18px; padding: 10px 20px;")

self.label_model_name = QLabel("Model Name:", self)
self.label_model_name.setStyleSheet("font-size: 18px; padding: 10px 20px;")
self.text_model_name = QLineEdit(self)
self.text_model_name.setStyleSheet("font-size: 18px; padding: 10px 20px;")

self.label_kanban_no = QLabel("Kanban No.:", self)
```

```
self.label_kanban_no.setStyleSheet("font-size: 18px; padding: 10px 20px;")
self.text_kanban_no = QLineEdit(self)
self.text_kanban_no.setStyleSheet("font-size: 18px; padding: 10px 20px;")

# ป้ายข้อความ
self.label_message = QLabel("Real time picture", self)
self.label_message.setStyleSheet("font-size: 18px; padding: 10px 20px;")
self.label_message2 = QLabel("Saved image file name :", self)
self.label_message2.setStyleSheet("font-size: 18px; padding: 10px 20px;")

# Create labels for counters
self.label_ok_count = QLabel("OK: 0", self)
self.label_ok_count.setStyleSheet("font-size: 18px; padding: 10px 20px; color: green;")

self.label_ng_count = QLabel("NG: 0", self)
self.label_ng_count.setStyleSheet("font-size: 18px; padding: 10px 20px; color: red;")

self.label_total_count = QLabel("Total: 0", self)
self.label_total_count.setStyleSheet("font-size: 18px; padding: 10px 20px; color:
blue;")


# ปุ่มกด
self.button_take = QPushButton("ถ่ายรูป (Take a shot)", self)
self.button_take.clicked.connect(self.take_picture)
self.button_take.setStyleSheet("font-size: 18px; padding: 10px 20px;")

self.button_quit = QPushButton("เสร็จสิ้น (Quit)", self)
self.button_quit.clicked.connect(self.close_app)
self.button_quit.setStyleSheet("font-size: 18px; padding: 10px 20px;")
```

```
# แสดงภาพ Real-time
self.label_image = QLabel(self)
#self.label_image.setFixedSize(1200, 600)
self.label_image.setFixedSize(800, 600)
self.label_image.setStyleSheet("border: 1px solid black; background-color: #333;")

# สร้าง QLabel เพื่อแสดงภาพที่สอง
self.label_image2 = QLabel(self)
#self.label_image2.setFixedSize(320, 240)
self.label_image2.setFixedSize(800, 600)
self.label_image2.setStyleSheet("border: 1px solid black; background-color: #ddd;")

# Layout
layout1 = QVBoxLayout()
picture_layout = QHBoxLayout()
message_layout = QHBoxLayout()
input_layout = QHBoxLayout()  # New layout for input fields
input_layout2 = QHBoxLayout()
button_layout = QHBoxLayout()
counter_layout = QHBoxLayout()


# Input fields layout
input_layout.addWidget(self.label_employee_number)
input_layout.addWidget(self.text_employee_number)
input_layout.addWidget(self.label_employee_name)
input_layout.addWidget(self.text_employee_name)
input_layout2.addWidget(self.label_model_name)
input_layout2.addWidget(self.text_model_name)
input_layout2.addWidget(self.label_kanban_no)
input_layout2.addWidget(self.text_kanban_no)
```

```
# Other layouts
picture_layout.addWidget(self.label_image)
picture_layout.addWidget(self.label_image2)

message_layout.addWidget(self.label_message)
message_layout.addWidget(self.label_message2)

counter_layout.addWidget(self.label_ok_count)
counter_layout.addWidget(self.label_ng_count)
counter_layout.addWidget(self.label_total_count)

button_layout.addWidget(self.button_take)
button_layout.addWidget(self.button_quit)

layout1.addLayout(input_layout)
layout1.addLayout(input_layout2)
layout1.addLayout(picture_layout)
layout1.addLayout(message_layout)
layout1.addLayout(counter_layout)
layout1.addLayout(button_layout)

container = QWidget()
container.setLayout(layout1)
self.setCentralWidget(container)

# เริ่มการรัน YOLO ใน Thread
self.yolo_thread = YOLOThread()
self.yolo_thread.image_saved.connect(self.show_message)
self.yolo_thread.image_to_show.connect(self.set_image)
self.yolo_thread.image_to_show_2.connect(self.set_image_2)  # เชื่อมต่อสัญญาณที่สอง
```

```
# Connect signals for counters
self.yolo_thread.update_ok_count.connect(self.update_ok_counter)
self.yolo_thread.update_ng_count.connect(self.update_ng_counter)
self.yolo_thread.update_total_count.connect(self.update_total_counter)

self.yolo_thread.finished.connect(self.close)
self.yolo_thread.start()

def take_picture(self):
# Get input data
self.yolo_thread.model_name = self.text_model_name.text()
self.yolo_thread.kanban_no = self.text_kanban_no.text()
self.yolo_thread.employee_name = self.text_employee_name.text()
self.yolo_thread.employee_number = self.text_employee_number.text()
self.yolo_thread.capture()

def close_app(self):
self.yolo_thread.stop()
self.yolo_thread.wait()  # Wait for the thread to finish
self.close()

def show_message(self, message):
self.label_message2.setText(message)

def set_image(self, qt_image: QImage):
if qt_image.isNull():
print(" ❌ รูปภาพที่ได้รับเป็น Null")
else:
print(" ✅ กำลังแสดงภาพใน QLabel")
```

```
self.label_image.setPixmap(QPixmap.fromImage(qt_image))


def set_image_2(self, pixmap: QPixmap):
if pixmap.isNull():
print("❌ รูปภาพที่สองเป็น Null")
else:
print("✅ กำลังแสดงภาพที่สองใน QLabel")
scaled_pixmap = pixmap.scaled(self.label_image2.size(),
Qt.AspectRatioMode.KeepAspectRatio,
Qt.TransformationMode.SmoothTransformation)
self.label_image2.setPixmap(scaled_pixmap)

def update_ok_counter(self, count: int):
self.label_ok_count.setText(f"OK: {count}")

def update_ng_counter(self, count: int):
self.label_ng_count.setText(f"NG: {count}")

def update_total_counter(self, count: int):
self.label_total_count.setText(f"Total: {count}")

def show_message(self, message: str):
self.label_message2.setText(message)


def closeEvent(self, event):
self.yolo_thread.stop()
self.yolo_thread.wait()
event.accept()
```

```
if __name__ == '__main__':
app = QApplication(sys.argv)
window = MainWindow()
window.show()
sys.exit(app.exec())
```

ประวัติผู้เขียน

ชื่อ	นาย ณัฐชัย รุ่งเชษฐ์
ชื่อการค้นคว้าอิสระ	การประยุกต์ใช้ระบบปัญญาประดิษฐ์ในการตรวจจับลักษณะข้อบกพร่องบนชิ้นส่วนอิเล็กทรอนิกส์
สาขาวิชา	วิศวกรรมการจัดการ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ประวัติ	ปีการศึกษา 2550 สำเร็จการศึกษาระดับปริญญาบริหารธุรกิจบัณฑิต สาขาภาษาอังกฤษธุรกิจ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา