



รายงานวิจัยฉบับสมบูรณ์

การประยุกต์ใช้งานความเป็นจริงเสริมในงานด้านอุตสาหกรรมอัตโนมัติ
(ระยะที่ 1)

Applied Augmented Reality for Industrial Automation (Phase 1)

นายประจักษ์ จิตเงินมะดัน หัวหน้าโครงการวิจัย

โครงการวิจัยประเภทเงินรายได้

คณะวิทยาการสารสนเทศ

ประจำปีงบประมาณ พ.ศ. 2566

มหาวิทยาลัยบูรพา

สัญญาเลขที่ 006/2566

รายงานวิจัยฉบับสมบูรณ์

การประยุกต์ใช้งานความเป็นจริงเสริมในงานด้านอุตสาหกรรมอัตโนมัติ
(ระยะที่ 1)

Applied Augmented Reality for Industrial Automation (Phase 1)

นายประจักษ์ จิตเงินมะดัน หัวหน้าโครงการวิจัย

คณะวิทยาการสารสนเทศ

มิถุนายน 2566

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับทุนสนับสนุนการวิจัยจากเงินรายได้ คณะวิทยาการสารสนเทศ ประจำปีงบประมาณ พ.ศ. 2566 เลขที่สัญญา 006/2566

คณะผู้วิจัยขอขอบคุณคณะวิทยาการสารสนเทศ และห้องปฏิบัติการวิจัยสื่อดิจิทัลและปฏิสัมพันธ์ (Digital Media and Interaction Laboratory) ที่เอื้อเฟื้องบประมาณในการดำเนินการ และสถานที่ในการวางแผน ปฏิบัติ และทดสอบชิ้นงานให้มีความสมบูรณ์ รวมถึงการให้บริการด้านสาธารณูปโภคต่าง ๆ ด้วย

และในท้ายที่สุด คณะผู้วิจัยขอขอบคุณผู้ร่วมออกแบบการทดลอง และทีมงานสนับสนุนทุกคนที่ทุ่มเท แรงกายแรงใจในการดำเนินการงานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดี

มิถุนายน 2567

บทคัดย่อ

ในอุตสาหกรรมและเทคนิคการผลิตสมัยใหม่ การบำรุงรักษาที่รวดเร็วและแม่นยำเป็นแนวคิดที่สำคัญ มีหลายแนวทางที่ตอบสนองข้อกำหนดเหล่านี้ รวมถึงเทคโนโลยีความจริงเสริม (Augmented Reality หรือ AR) เทคโนโลยี AR ร่วมกับเทคนิคการตรวจจับวัตถุแบบเรียลไทม์ นำเสนอวิธีแก้ปัญหาที่เป็นไปได้ในการปรับปรุงและเปลี่ยนแปลงเทคนิคการบำรุงรักษาในอนาคตอันใกล้ เพื่อหาโมเดล AI ที่เหมาะสมที่สุดสำหรับวัตถุประสงค์นี้ การศึกษาได้เปรียบเทียบโมเดล YOLOv5, Resnet50 และ SSD MobileNet ที่ผ่านการฝึกฝนมาแล้วในการตรวจจับองค์ประกอบของ PLCnext โดยใช้ภาพ 960 ภาพสำหรับการฝึกฝนและทดสอบ โมเดลต่าง ๆ ได้รับการฝึกฝนและทดสอบ โดยเน้นถึงความสามารถในการทำงานแบบเรียลไทม์ ผลการศึกษาแสดงให้เห็นว่าโมเดล YOLOv5 ที่ผ่านการฝึกฝนมาแล้วเป็นโมเดลที่ดีที่สุดในการเพิ่มประสิทธิภาพการตรวจสอบแบบอินเทอร์แอคทีฟ ลดเวลาการหยุดทำงาน และเพิ่มประสิทธิภาพในสภาพแวดล้อมอุตสาหกรรม อีกทั้งยังเหมาะสำหรับการพัฒนาแอปพลิเคชัน AR ในระบบอัตโนมัติ 4.0 หลังจากได้รับการฝึกฝนด้วยภาพที่จำเป็น

Abstract

In the modern industries and manufacturing techniques, quick and precise maintenance is a vital concept. Several approaches fulfill these requirements, including the Augmented Reality (AR) technology. The AR technology, combined with real-time object detection techniques, offers a possible solution to improve and disrupt the maintenance techniques in the near future. For finding the most suitable AI model for this purpose, this study compared the pre-trained YOLOv5, Resnet50, and SSD MobileNet models in detecting PLCnext components. Using 960 images for training and testing purposes, the models were trained and tested, highlighting their real-time capabilities. The results show that the pre-trained YOLOv5 model is the best to enhance interactive monitoring, minimize downtime, and boost efficiency in industrial settings and suitable for AR application development in automation 40 after being trained with required images.

สารบัญ

1	บทนำ	1
1.1	ที่มาและความสำคัญ.....	1
1.2	วัตถุประสงค์	2
1.3	ขอบเขตของการวิจัย.....	2
1.4	เครื่องมือที่ใช้การพัฒนา.....	2
1.4.1	อุปกรณ์ด้านฮาร์ดแวร์ (Hardware).....	2
1.4.2	ซอฟต์แวร์ที่ใช้ในการพัฒนา (Software)	2
1.5	วิธีการดำเนินการวิจัย (Research methodology).....	3
1.5.1	ศึกษาปัญหาและงานวิจัยที่เกี่ยวข้อง	3
1.5.2	ศึกษาและสร้างตัวต้นแบบโมเดลสำหรับการรู้จำความผิดปกติของเครื่องจักรด้วยวิธีการ Object Detection	3
1.5.3	เขียนโปรแกรมเพื่อทดสอบการทำงานของ AR และทำการเก็บข้อมูลการใช้งาน	3
1.5.4	วิเคราะห์ผลจากข้อมูลที่รวบรวมไว้ ทำการปรับปรุงกระบวนการตามความเหมาะสม	3
1.5.5	เขียนบทความวิจัยและตีพิมพ์บทความวิจัย.....	3
1.5.6	สรุปผลและทำรายงานฉบับสมบูรณ์.....	3
1.6	ประโยชน์ที่คาดว่าจะได้รับ	3
2	ทฤษฎีและโครงการที่เกี่ยวข้อง.....	4
2.1	ทฤษฎีที่เกี่ยวข้อง	4
2.1.1	ทฤษฎีการมองเห็นด้วยคอมพิวเตอร์	4
2.1.2	ปัญญาประดิษฐ์	4
2.1.3	การตรวจจับวัตถุ.....	4

2.1.4	ระบบเสริมความเป็นจริงหรือเทคโนโลยีความเป็นจริงเสริม (Augmented Reality)	5
2.1.5	HoloLens.....	6
2.1.6	โปรแกรมยูนิตี้ (Unity).....	7
2.1.7	โปรแกรมเลเบลอิมเมจ (LabelIMG).....	7
2.1.8	โปรแกรมอนาคอนด้า (Anaconda).....	8
2.1.9	Pretrained Model	8
2.1.10	YOLO	8
2.1.11	TensorFlow	9
2.1.12	ResNet50	9
2.1.13	SSD MobileNet	10
2.2	งานที่เกี่ยวข้อง (Related Works)	10
3	วิธีการดำเนินงานโครงการ	11
3.1	ภาพรวมของการดำเนินงาน	11
3.2	การเตรียมข้อมูล	12
3.2.1	การเก็บรวบรวมภาพ (Image Obtaining)	12
3.2.2	การปรับขนาดภาพ (Image Resizing)	14
3.2.3	การติดป้ายกำกับภาพ (Image Labeling).....	16
3.2.4	การเสริมเพิ่มคุณลักษณะของภาพ (Image Augmentation)	17
3.2.5	การแปลงรูปแบบข้อมูล (Data Format Conversion).....	21
3.3	การฝึกโมเดล	22
4	ผลการดำเนินการ	25
5	สรุปผล.....	28
5.1	สรุปผลการดำเนินงาน.....	28

5.2	ข้อค้นพบสำคัญ	28
5.3	ปัญหา อุปสรรค และข้อจำกัด	28
5.4	ข้อเสนอแนะ	28
5.5	แนวทางในการพัฒนาในอนาคต.....	29
5.6	ผลงานวิจัยที่ได้รับการตีพิมพ์และเผยแพร่แล้ว	29
6	บรรณานุกรม	35
	References	35
7	ภาคผนวก.....	36
7.1	อธิบาย Code Python เบื้องต้น.....	36
7.1.1	โค้ดปรับขนาดรูปภาพ.....	36
7.1.2	โค้ดหมุนรูปภาพ	38
7.2	ตัวอย่างภาพจากชุดข้อมูล.....	40
7.2.1	ชุดข้อมูลแบบทั้งเครื่อง PLCnext	40
7.2.2	ชุดข้อมูลแบบเฉพาะเจาะจงจากจุดสนใจ	46
8	ประวัตินักวิจัยและคณะ	52
8.1	หัวหน้าโครงการวิจัย	52

สารบัญตาราง

ตารางที่ 1 ค่าความสูญเสียของแต่ละโมเดล	25
ตารางที่ 2 ค่า mAP ของแต่ละโมเดล	26
ตารางที่ 3 ชุดข้อมูลแบบทั้งเครื่อง PLCnext	40
ตารางที่ 4 ชุดข้อมูลแบบเฉพาะเจาะจงจากจุดสนใจ	46

สารบัญภาพ

ภาพที่ 1 ระบบเสริมความเป็นจริง (Augmented Reality)	5
ภาพที่ 2 Hololens	6
ภาพที่ 3 Unity	7
ภาพที่ 4 LabelIMG	7
ภาพที่ 5 Anaconda	8
ภาพที่ 6 TensorFlow	9
ภาพที่ 7 ภาพรวมของระบบ	11
ภาพที่ 8 การเตรียมข้อมูล	12
ภาพที่ 9 การเก็บภาพจากมุมมองด้านบน	13
ภาพที่ 10 การเก็บภาพจากมุมมองระนาบแนวตั้ง	13
ภาพที่ 11 การวัดระยะห่างแบบเฉพาะส่วน	14
ภาพที่ 12 Flowchart แสดงการทำงานของโค้ดปรับขนาดรูปภาพ	15
ภาพที่ 13 การติดป้ายกำกับรูปภาพ	17
ภาพที่ 14 การสร้างภาพใหม่จากการหมุนรูปภาพต้นฉบับ	18
ภาพที่ 15 Flowchart แสดงการทำงานของโค้ดหมุนรูปภาพ	19
ภาพที่ 16 กระบวนการฝึกอบรมโมเดล	23
ภาพที่ 17 กราฟแสดงค่าความสูญเสีย	23
ภาพที่ 18 กราฟแสดงค่าความสูญเสียของแต่ละโมเดล	25
ภาพที่ 19 กราฟแสดงค่า mAP ของแต่ละโมเดล	27
ภาพที่ 20 โค้ดปรับขนาดรูปภาพ	36
ภาพที่ 21 นำเข้าไลบรารี os และ PIL	36
ภาพที่ 22 สร้างฟังก์ชัน resize_images_in_folder	37
ภาพที่ 23 วนลูปหาไฟล์รูปภาพ	37
ภาพที่ 24 เปิดรูปภาพและทำการปรับขนาด	37
ภาพที่ 25 บันทึกภาพที่ได้รับการปรับขนาดแล้ว	37
ภาพที่ 26 กำหนดเส้นทางของโฟลเดอร์ และ เรียกใช้ฟังก์ชันปรับขนาด	38

ภาพที่ 27 โค้ดหมุนรูปภาพ.....	38
ภาพที่ 28 นำเข้าไลบรารี os และ PIL	38
ภาพที่ 29 กำหนดเส้นทางของโพลเดอร์ดันทางและปลายทาง	39
ภาพที่ 30 ววนลูปผ่านแต่ละไฟล์ในโพลเดอร์ดันทาง	39
ภาพที่ 31 เปิดไฟล์ภาพและวนลูปหมุนรูปภาพ	39
ภาพที่ 32 สร้างชื่อไฟล์ใหม่.....	39
ภาพที่ 33 บันทึกภาพที่หมุนแล้ว	39

คำอธิบายสัญลักษณ์และคำย่อ

คำศัพท์	คำอธิบาย
Augmented Reality (AR)	Augmented Reality (AR) คือ การผสมผสานข้อมูลเสมือนจริงเข้ากับโลกแห่งความเป็นจริง ทำให้ผู้ใช้งานสามารถมองเห็นวัตถุหรือข้อมูลเสมือนจริงปรากฏบนสภาพแวดล้อมจริงที่พวกเขาสัมผัสอยู่ ผ่านอุปกรณ์เช่น แว่นตา AR, สมาร์ทโฟน หรือแท็บเล็ต ผู้ใช้งานสามารถรับประสบการณ์ที่เพิ่มเติมจากข้อมูลเสมือนจริงที่ซ้อนทับกับโลกจริง ทำให้สามารถทำงานหรือโต้ตอบกับสิ่งแวดล้อมได้อย่างมีประสิทธิภาพ AR ช่วยเพิ่มการรับรู้และการตอบสนองต่อสถานการณ์โดยการเสริมสิ่งที่เราเห็นด้วยข้อมูลกราฟิกหรือข้อความที่มีประโยชน์ เช่น การแสดงข้อมูลเทคนิคเกี่ยวกับเครื่องจักรที่ผู้ปฏิบัติงานกำลังมองเห็น หรือการแสดงทิศทางการนำทางในสภาพแวดล้อมจริง
Virtual Reality (VR)	Virtual Reality (VR) คือ การสร้างระบบหรือสิ่งแวดล้อมจำลองในรูปแบบของภาพสามมิติ โดยผู้ใช้งานจะต้องใช้แว่นพิเศษที่เรียกว่า แว่น VR ในการรับรู้ความรู้สึกเสมือนจริงเมื่อใช้งานระบบ โดยจะมีการจำลองของจริงและ ผู้ใช้งานสามารถปฏิสัมพันธ์กับวัตถุหรือสิ่งแวดล้อมจำลองนั้นได้
Hololens	HoloLens คือ อุปกรณ์ความเป็นจริงเสริม (Augmented Reality, AR) จาก Microsoft ที่สร้างประสบการณ์ที่ผสมผสานเอาโลกเสมือนจริงเข้ากับโลกจริงผ่านแว่นตาอัจฉริยะ ผู้ใช้ HoloLens สามารถมองเห็นวัตถุหรือแอปพลิเคชันที่มีรูปแบบสามมิติซึ่งซ้อนทับกับสิ่งแวดล้อมจริงที่ตนอยู่ โดยไม่ต้องใช้อุปกรณ์เพิ่มเติมใดๆ การออกแบบที่ผสมผสานนี้ช่วยให้ผู้ใช้สามารถโต้ตอบกับวัตถุเสมือนจริงโดยใช้การเคลื่อนไหวของมือและคำสั่งเสียงได้อย่างอิสระ ทำให้สามารถใช้งานได้หลากหลายตั้งแต่การออกแบบ, การศึกษา, การผลิตจนถึงการฝึกอบรม HoloLens มอบประสบการณ์ที่ไม่เหมือนใครในการเชื่อมต่อข้อมูลดิจิทัลกับโลกจริง ซึ่งช่วยเพิ่มประสิทธิภาพในการทำงานและปรับปรุงการเรียนรู้ผ่านการโต้ตอบแบบเสริมภาพ

คำศัพท์	คำอธิบาย
Unity	Unity เป็นเอ็นจินพัฒนาเกมที่ทำให้ความสามารถในการสร้างประสบการณ์ความเป็นจริงเสริม (AR) ที่โต้ตอบได้ด้วย Unity นักพัฒนาสามารถออกแบบวัตถุและสภาพแวดล้อมเสมือนที่ซ้อนทับอยู่บนโลกจริง อนุญาตให้ผู้ใช้งานมองเห็นและโต้ตอบกับเนื้อหาดิจิทัลผ่านอุปกรณ์ต่าง ๆ เช่น HoloLens และสมาร์ทโฟน Unity นำเสนอเครื่องมือที่มีประสิทธิภาพสำหรับการพัฒนา AR ทำให้เป็นเครื่องมือที่นิยมใช้ในหลายๆ อุตสาหกรรม
Artificial Intelligence (AI)	ปัญญาประดิษฐ์ (Artificial Intelligence, AI) เป็นสาขาของวิทยาการคอมพิวเตอร์ที่เน้นการพัฒนาเครื่องจักรและระบบที่สามารถจำลองความสามารถทางปัญญาของมนุษย์ได้ เช่น การเรียนรู้, การแก้ปัญหา, และการตัดสินใจ AI ทำงานโดยการวิเคราะห์ข้อมูลขนาดใหญ่และระบุรูปแบบที่ซับซ้อนเพื่อทำนายผลลัพธ์หรือตัดสินใจโดยอัตโนมัติ ปัญญาประดิษฐ์ถูกใช้งานอย่างแพร่หลายในหลายๆ อุตสาหกรรม รวมทั้งในการพัฒนาโซลูชันที่ช่วยให้ระบบสามารถปรับตัวและตอบสนองต่อเหตุการณ์แบบเรียลไทม์ได้ AI มีความสำคัญในการปรับปรุงกระบวนการ, ประหยัดเวลา, และเพิ่มประสิทธิภาพในหลากหลายสาขาวิชา
Machine Learning (ML)	Machine Learning (ML) เป็นสาขาย่อยของปัญญาประดิษฐ์ (AI) ที่ศึกษาการพัฒนาอัลกอริธึมที่ช่วยให้คอมพิวเตอร์มีความสามารถในการเรียนรู้จากข้อมูลและปรับปรุงการทำงานโดยอัตโนมัติโดยไม่ต้องมีการเขียนโปรแกรมแบบชัดเจน การเรียนรู้ของเครื่องจะประมวลผลข้อมูลขนาดใหญ่, ระบุรูปแบบและความสัมพันธ์ภายในข้อมูล และใช้ความรู้นั้นในการตัดสินใจหรือทำนายผลลัพธ์ในอนาคต Machine Learning ถูกนำไปใช้ในหลายๆ อุตสาหกรรมเพื่อเพิ่มความแม่นยำและความเร็วในการตัดสินใจ รวมทั้งในการพัฒนาแอปพลิเคชันที่ต้องการการโต้ตอบอย่างฉับไวและเฉพาะเจาะจงต่อสถานการณ์ที่เกิดขึ้นจริง

1 บทนำ

1.1 ที่มาและความสำคัญ

เทคโนโลยีความเป็นจริงเสริมนั้นได้เข้ามามีบทบาทในอุตสาหกรรมอัตโนมัติอย่างมากเมื่อไม่นานมานี้ หนึ่งในเหตุผลที่สำคัญคือ การที่เทคโนโลยีนี้สามารถเติมเต็มข้อมูลให้กับผู้ปฏิบัติงานได้อย่างเพียงพอที่จะสามารถทำงานได้อย่างไม่มีสะดุด การให้ข้อมูลเสริมเข้าไปในสิ่งแวดล้อมจริงนี้ เรียกว่า Augmented Reality (AR) หรือความเป็นจริงเสริม (Wikipedia, 2023) ซึ่งความเป็นจริงเสริมนี้จะช่วยให้มีข้อมูลเพื่อการตัดสินใจเพิ่มขึ้นสำหรับผู้ใช้งาน หรือสามารถประยุกต์ใช้เพื่อการทำงานหรือฝึกปฏิบัติในอุตสาหกรรมอัตโนมัติ เช่น การซ่อมบำรุง การตรวจสอบสถานะของเครื่องจักร การวิเคราะห์การทำงานของเครื่องจักร เป็นต้น

มีงานวิจัยมากมายที่ศึกษาเกี่ยวกับเทคโนโลยี AR กับอุตสาหกรรมอัตโนมัติ เช่น งานวิจัยของ Chiang et al (Chiang, Shang, & Qiao, 2022) แสดงให้เห็นถึงการประยุกต์ใช้งานเทคโนโลยี AR กับการฝึกปฏิบัติในวิทยาลัยเทคนิคและการอาชีพเกี่ยวกับการฝึกประสบการณ์การทำงานในอุตสาหกรรม ซึ่งได้ผลดีมากและสามารถนำไปต่อยอดในการฝึกปฏิบัติในด้านอื่น ๆ ได้อีกด้วย นอกจากนั้น ในแง่ของทางเศรษฐกิจ ข้อมูลจาก World Economic Forum ระบุว่า การใช้งาน AR นั้น จะเกิดขึ้นอย่างแพร่หลาย เนื่องจากเทคโนโลยีนี้ สามารถนำมาเพิ่มการรับรู้ด้วยการสร้างภาพกราฟิก การเพิ่มข้อมูลเฉพาะส่วน และการบอกเล่าเรื่องราว ซึ่งจะทำให้การเรียนรู้และการทำงานนั้นสะดวกสบายขึ้นมาก (World Economic Forum, 2023) นอกจากนั้น ยังมีการใช้เทคโนโลยี AR ในเชิงอุตสาหกรรมอัตโนมัติ ที่ช่วยให้สามารถทำงานได้อย่างรวดเร็ว เช่น การทำงานร่วมกับแขนกลหรือหุ่นยนต์ ที่สามารถระบุตำแหน่งได้อย่างแม่นยำ การทำงานในเชิงการซ่อมบำรุงที่สามารถแก้ไขความบกพร่องของเครื่องจักรเบื้องต้นได้อย่างทันที โดยอาจจะมีการใช้เทคโนโลยีอื่น ๆ เข้ามาช่วยในการทำงานได้ด้วย เช่น การโทรศัพท์แบบวิดีโอกับวิศวกรผู้เชี่ยวชาญ หรือการเชื่อมต่อกับเซ็นเซอร์หรืออุปกรณ์ต่าง ๆ เพื่อรับข้อมูลนำเข้าจากอุปกรณ์จำพวก Internet of Things (IoT) และสามารถส่งไปทำการวิเคราะห์และประมวลผลเพื่อทำการแก้ปัญหาต่อไป (Sureshkumar, Agash, Ramya, Kaviyaraj, & Elanchezhian, 2021) เป็นต้น

ในการวิจัยนี้ เป็นการศึกษาเกี่ยวกับการผสมผสานเทคโนโลยี AR เข้ากับการรู้จำภาพในรูปแบบ Object Detection และการใช้ปัญญาประดิษฐ์ (Artificial Intelligence: AI) ที่เหมาะสมในการวิเคราะห์เครื่องจักรในอุตสาหกรรมอัตโนมัติ เพื่อทำการตรวจสอบ ดูแล และซ่อมบำรุงได้ตามลักษณะทางกายภาพที่ปรากฏ เช่น มีการกระพริบของไฟสัญญาณที่เกี่ยวข้อง มีการแจ้งเตือนหรือ alarm ไฟสัญญาณ หรือมีการชำรุดเสียหายของชิ้นส่วน เป็นต้น โดยในการวิเคราะห์นี้ จะใช้พื้นฐานการควบคุมเครื่องจักรด้วยเครื่องควบคุม PLCnext (PLCnext Community, 2023) และการสร้างตัวต้นแบบโมเดลในการรู้จำความผิดปกติของเครื่องควบคุมนั้น บนพื้นฐานของการรู้จำภาพแบบ Object Detection เพื่อสร้างกระบวนการหรือขั้นตอนในการดูแล ปรับปรุง หรือซ่อมแซมเครื่องจักรนั้นตามความเหมาะสม

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาความเป็นไปได้ในการออกแบบและพัฒนาเทคโนโลยีความเป็นจริงเสริม (Augmented Reality) เพื่อการใช้งานในอุตสาหกรรมอัตโนมัติ
- 2) เพื่อศึกษาการประยุกต์ใช้งาน AR การรู้จำภาพแบบ Object Detection และตัวต้นแบบ AI สำหรับงานด้านอุตสาหกรรมอัตโนมัติ

1.3 ขอบเขตของการวิจัย

- 1) เทคโนโลยีควบคุมเครื่องจักรที่ใช้งานคือ PLCnext
- 2) เทคโนโลยี AR ที่ใช้งาน อยู่บนพื้นฐานเทคโนโลยีของ Microsoft
- 3) การสร้างแบบจำลองจะใช้งานได้กับ AR เท่านั้น ไม่รวม VR

1.4 เครื่องมือที่ใช้การพัฒนา

1.4.1 อุปกรณ์ด้านฮาร์ดแวร์ (Hardware)

ลักษณะและส่วนประกอบคอมพิวเตอร์ที่แนะนำ

- 1) กราฟฟิก (Graphic) : NVIDIA RTX 3070 Laptop
- 2) หน่วยประมวลผล (Processor) : Intel i7-11800H equivalent or greater หรือดีกว่า
- 3) หน่วยความจำ (Memory) : RAM 32 GB หรือมากกว่า
- 4) การแสดงผลวิดีโอ (Output) : HDMI 14, DisplayPort 12 หรือใหม่กว่า
- 5) พอร์ต USB (Port USB) : 1x USB 20 หรือดีกว่า
- 6) ระบบปฏิบัติการ (Operating System) : Windows 10

อุปกรณ์ด้านฮาร์ดแวร์อื่น ๆ

- 7) Hololens 2
- 8) Monitor

1.4.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา (Software)

- 1) Unity 2022
- 2) LabelIMG
- 3) Anaconda

4) Visual Studio Code

1.5 วิธีการดำเนินการวิจัย (Research methodology)

1.5.1 ศึกษาปัญหาและงานวิจัยที่เกี่ยวข้อง

- 1) วางแผนการดำเนินงานและศึกษาความเป็นไปได้
- 2) ศึกษาและรวบรวมข้อมูลเกี่ยวกับภาษาที่ใช้ในการเขียนโปรแกรมและเครื่องมือที่ใช้

1.5.2 ศึกษาและสร้างตัวต้นแบบโมเดลสำหรับการรู้จำความผิดปกติของเครื่องจักรด้วยวิธีการ Object Detection

- 1) ทดลองการทำงานและการแสดงข้อมูลของอุปกรณ์
- 2) นำโมเดล Object Detection ที่เหมาะสมมาใช้งานร่วมกับอุปกรณ์ AR

1.5.3 เขียนโปรแกรมเพื่อทดสอบการทำงานของ AR และทำการเก็บข้อมูลการใช้งาน

- 1) การออกแบบวิธีการนำข้อมูลจากโมเดลมาใช้งาน
- 2) การออกแบบการแสดงผลข้อมูลที่ได้จากโมเดล

1.5.4 วิเคราะห์ผลจากข้อมูลที่รวบรวมไว้ ทำการปรับปรุงกระบวนการตามความเหมาะสม

- 1) ทดลองและวิเคราะห์หาจุดบกพร่องของโมเดล
- 2) ปรับปรุงโมเดลด้วยการเรียนรู้จากจุดบกพร่อง

1.5.5 เขียนบทความวิจัยและตีพิมพ์บทความวิจัย

1.5.6 สรุปผลและทำรายงานฉบับสมบูรณ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้ทราบและบันทึกองค์ความรู้ในวิธีการนำแบบจำลองปัญญาประดิษฐ์รูปแบบ Object Detection มาใช้งานกับเทคโนโลยี AR
- 2) ได้ตัวต้นแบบของแบบจำลองปัญญาประดิษฐ์ (AI Model) สำหรับการตรวจจับความผิดปกติของอุปกรณ์ควบคุมเครื่องจักรในอุตสาหกรรมอัตโนมัติ
- 3) เป็นแนวทางของการพัฒนาระบบตรวจจับความผิดปกติของอุปกรณ์ควบคุมเครื่องจักรด้วยเทคโนโลยี AR

2 ทฤษฎีและโครงการที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ทฤษฎีการมองเห็นด้วยคอมพิวเตอร์

การมองเห็นด้วยคอมพิวเตอร์ (Computer Vision) เป็นหนึ่งในสาขาวิทยาการที่สำคัญในด้านปัญญาประดิษฐ์ ซึ่งมีหลักการทำงานที่เกี่ยวข้องกับการให้คอมพิวเตอร์มีความสามารถในการ "มองเห็น" และตีความสิ่งที่มองเห็นได้จากภาพหรือวิดีโอ สาขานี้มุ่งเน้นไปที่การพัฒนาอัลกอริทึมที่สามารถตรวจจับ ระบุ และจำแนกวัตถุหรือลักษณะต่าง ๆ ที่ปรากฏในภาพเพื่อให้สามารถทำการวิเคราะห์หรือตัดสินใจได้อย่างอัตโนมัติ การใช้งานของการมองเห็นด้วยคอมพิวเตอร์นั้นกว้างขวาง ตั้งแต่การตรวจสอบคุณภาพในสายการผลิต การนำทางของหุ่นยนต์ ไปจนถึงการวิเคราะห์ข้อมูลทางการแพทย์และการจดจำใบหน้า ด้วยการพัฒนาที่รวดเร็วในเทคโนโลยีเซ็นเซอร์และอัลกอริทึม การมองเห็นด้วยคอมพิวเตอร์ได้กลายเป็นเครื่องมือที่สำคัญในการเพิ่มประสิทธิภาพและการทำงานอัตโนมัติในหลายๆ อุตสาหกรรม

2.1.2 ปัญญาประดิษฐ์

ปัญญาประดิษฐ์ (Artificial Intelligence, AI) เป็นสาขาของวิทยาการคอมพิวเตอร์ที่เกี่ยวข้องกับการสร้างระบบหรือเครื่องจักรที่สามารถจำลองความสามารถทางปัญญาของมนุษย์ได้ รวมถึงการเรียนรู้ (learning) การอ้างอิงตามเหตุผล (reasoning) การรับรู้ (perception) การทำความเข้าใจภาษาธรรมชาติ (natural language understanding) และการแก้ไขปัญห ในปัจจุบัน AI ถูกนำมาใช้ในหลากหลายอุตสาหกรรมเพื่อเพิ่มความเร็วและความแม่นยำในการทำงาน โดยระบบ AI สามารถพัฒนาจากการเรียนรู้ข้อมูลขนาดใหญ่ผ่านเทคนิคต่าง ๆ เช่น เครื่องเรียนรู้ (machine learning) และการเรียนรู้เชิงลึก (deep learning) การเรียนรู้เหล่านี้ช่วยให้อาศัยเครือข่ายประสาทเทียมที่สามารถจำลองกระบวนการทางประสาทของสมองมนุษย์เพื่อรับรู้และประมวลผลข้อมูล ความก้าวหน้าของ AI มีบทบาทสำคัญในการเปลี่ยนแปลงวิธีการทำงานในหลายๆ ภาคส่วน ทำให้งานที่ซับซ้อนหรือต้องการความแม่นยำสูงสามารถทำได้ง่ายขึ้นและมีประสิทธิภาพมากขึ้น

2.1.3 การตรวจจับวัตถุ

การตรวจจับวัตถุ (Object Detection) เป็นหนึ่งในสาขาย่อยของการมองเห็นด้วยคอมพิวเตอร์ที่มีการศึกษาวิธีการในการระบุและระบุตำแหน่งของวัตถุหนึ่งหรือหลายวัตถุภายในภาพหรือวิดีโอ กระบวนการนี้ไม่เพียงแต่จำแนกประเภทวัตถุเท่านั้น แต่ยังระบุขอบเขตของวัตถุนั้นในภาพหรือวิดีโอ (bounding boxes) การตรวจจับวัตถุมีความสำคัญมากในหลายๆ แอปพลิเคชัน เช่น การวิเคราะห์วิดีโอจากกล้องวงจรปิดเพื่อการรักษาความปลอดภัย ระบบนำทางของยานพาหนะอัตโนมัติ การจดจำวัตถุเพื่อการควบคุมคุณภาพในการผลิต และอื่นๆ

เทคนิคในการตรวจจับวัตถุมักจะพึ่งพาการใช้การเรียนรู้ของเครื่องและการเรียนรู้เชิงลึก โดยใช้เครือข่ายประสาทเทียมที่ได้รับการฝึกฝนเพื่อรู้จำและทำความเข้าใจเกี่ยวกับลักษณะต่าง ๆ ของวัตถุจากฐานข้อมูลภาพขนาดใหญ่ ตัวอย่างของโมเดลที่ใช้ในการตรวจจับวัตถุ ได้แก่ YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), และ Faster R-CNN ซึ่งแต่ละโมเดลมีจุดเด่นและข้อจำกัดที่ต่างกันเหมาะสำหรับการประยุกต์ใช้งานที่แตกต่างกัน

2.1.4 ระบบเสริมความเป็นจริงหรือเทคโนโลยีความเป็นจริงเสริม (Augmented Reality)

เทคโนโลยีความเป็นจริงเสริม (Augmented Reality หรือ AR) เป็นเทคโนโลยีที่ผสมผสานโลกดิจิทัลกับโลกจริงโดยการเพิ่มข้อมูลเสมือนเข้ามาในสภาพแวดล้อมจริงผ่านอุปกรณ์ต่าง ๆ เช่น แว่นตา AR หรือสมาร์ทโฟน การใช้ AR ช่วยให้ผู้ใช้สามารถมองเห็นและโต้ตอบกับวัตถุเสมือนที่ปรากฏในโลกจริง ทำให้เกิดประสบการณ์ที่เสมือนจริงและมีการโต้ตอบอย่างมีประสิทธิภาพกับสภาพแวดล้อมในชีวิตประจำวัน โดยไม่ต้องตัดขาดจากโลกจริง เช่นเดียวกับเทคโนโลยี VR

เทคโนโลยี AR ถูกนำมาใช้ในหลากหลายด้าน ไม่ว่าจะเป็นการฝึกอบรม การบำบัดรักษาทางการแพทย์ การท่องเที่ยว และการเล่นเกม ตัวอย่างเช่น การใช้ AR ในการฝึกอบรมสามารถช่วยให้ผู้เรียนเข้าใจเนื้อหาได้ง่ายขึ้นด้วยการแสดงภาพ 3 มิติ หรือการใช้ AR ในทางการแพทย์เพื่อช่วยในการวินิจฉัยโรคและการรักษาที่แม่นยำขึ้น ด้วยความสามารถในการเพิ่มข้อมูลเสมือนเข้ามาในสภาพแวดล้อมจริง AR จึงเป็นเทคโนโลยีที่มีศักยภาพในการปรับปรุงประสิทธิภาพและเพิ่มความน่าสนใจในหลากหลายกิจกรรม



ภาพที่ 1 ระบบเสริมความเป็นจริง (Augmented Reality)*

* ที่มา <https://medium.com/@huguesjoublin/how-can-virtual-reality-revolutionize-employee-engagement-programs-892f1b3d8a46>

2.1.5 HoloLens

HoloLens เป็นอุปกรณ์สวมศีรษะที่พัฒนาโดย Microsoft ซึ่งรวมเทคโนโลยีความเป็นจริงเสริม (Augmented Reality - AR) และความเป็นจริงผสม (Mixed Reality - MR) ไว้ด้วยกัน HoloLens ช่วยให้ผู้ใช้สามารถมองเห็นและโต้ตอบกับวัตถุเสมือนในโลกจริงผ่านหน้าจอแว่นที่มีความโปร่งใส ซึ่งช่วยให้การทำงานและการเรียนรู้มีความสมจริงและมีปฏิสัมพันธ์มากขึ้น

HoloLens รองรับการพัฒนาแอปพลิเคชันผ่านแพลตฟอร์มการพัฒนาอย่าง Unity และ Unreal Engine ทำให้นักพัฒนาสามารถสร้างแอปพลิเคชัน 3D และ AR ได้อย่างมีประสิทธิภาพ ด้วยเครื่องมือที่ครบครันและการรองรับการเขียนโค้ดด้วยภาษา C# และ API ที่หลากหลาย นักพัฒนาสามารถเพิ่มวัตถุเสมือนเข้าไปในโลกจริงและสร้างประสบการณ์ที่น่าทึ่งให้กับผู้ใช้

หนึ่งในความสามารถที่โดดเด่นของ HoloLens คือการตรวจจับและติดตามตำแหน่งของวัตถุและผู้ใช้ในพื้นที่จริง ซึ่งช่วยให้การโต้ตอบกับวัตถุเสมือนมีความแม่นยำและสมจริง HoloLens ยังมีระบบเสียงแบบสามมิติที่ช่วยให้ผู้ใช้สามารถรับรู้ทิศทางของเสียงในพื้นที่เสมือนจริง ทำให้ประสบการณ์การใช้งานมีความสมบูรณ์ยิ่งขึ้น

ด้วยความสามารถในการสร้างแอปพลิเคชัน AR และ MR ที่มีประสิทธิภาพและใช้งานง่าย HoloLens จึงเป็นเครื่องมือที่มีความสำคัญในหลาย ๆ วงการ ไม่ว่าจะเป็นการแพทย์ การศึกษา การฝึกอบรม และการผลิต ทำให้นักพัฒนาและผู้ใช้สามารถสร้างสรรค์นวัตกรรมใหม่ ๆ และปรับปรุงกระบวนการทำงานให้มีประสิทธิภาพมากขึ้น



ภาพที่ 2 Hololens*

* ที่มา <https://www.monarch-innovation.com/microsoft-hololens>

2.1.6 โปรแกรมยูนิตี้ (Unity)

Unity เป็นแพลตฟอร์มการพัฒนาเกมและการสร้างภาพจำลองแบบสามมิติที่ได้รับความนิยมอย่างแพร่หลาย เนื่องจากมีเครื่องมือที่ครบครันและใช้งานง่าย Unity รองรับการเขียนโค้ดด้วยภาษา C# และมี API ที่หลากหลาย ช่วยให้ผู้พัฒนาสามารถสร้างเกมและแอปพลิเคชัน 3D ได้อย่างมีประสิทธิภาพ นอกจากนี้ Unity ยังมีความสามารถในการสร้างแอปพลิเคชัน AR (Augmented Reality) ที่ช่วยให้ผู้พัฒนาสามารถเพิ่มวัตถุเสมือนเข้าไปในโลกจริงผ่านอุปกรณ์ต่าง ๆ เช่น สมาร์ทโฟนหรือแว่นตา AR Unity มีเครื่องมือที่ช่วยในการสร้างและจัดการวัตถุ 3D รวมถึงระบบฟิสิกส์ที่สมจริง ทำให้การพัฒนาแอปพลิเคชัน AR เป็นไปได้อย่างรวดเร็วและง่ายดาย เหมาะสำหรับทั้งผู้เริ่มต้นและผู้พัฒนามืออาชีพ



ภาพที่ 3 Unity

2.1.7 โปรแกรมเลเบลอิมเมจ (LabelIMG)

LabelIMG เป็นเครื่องมือโอเพ่นซอร์สที่ใช้สำหรับการติดป้ายกำกับภาพ (Image Annotation) ซึ่งมีความสำคัญอย่างยิ่งในงานด้านปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง (Machine Learning) โดยเฉพาะในการพัฒนาโมเดลการตรวจจับวัตถุ (Object Detection) และการจำแนกภาพ (Image Classification) โปรแกรมนี้รองรับการทำงานบนหลายแพลตฟอร์มทั้ง Windows, macOS และ Linux การใช้งาน LabelIMG นั้นง่ายและสะดวก ผู้ใช้สามารถเปิดภาพที่ต้องการติดป้ายกำกับและใช้เครื่องมือในโปรแกรมเพื่อสร้างกรอบสี่เหลี่ยมรอบวัตถุที่สนใจและระบุประเภทของวัตถุเหล่านั้น นอกจากนี้ยังสามารถบันทึกข้อมูลป้ายกำกับในรูปแบบ XML หรือ YOLO ที่รองรับโดยหลายแพลตฟอร์มการเรียนรู้ของเครื่อง ซึ่ง LabelIMG นับเป็นเครื่องมือที่มีประสิทธิภาพและเป็นที่นิยมในหมู่นักวิจัยและผู้พัฒนาในวงการนี้



LabelImg

ภาพที่ 4 LabelIMG

2.1.8 โปรแกรมอนาคอนด้า (Anaconda)

Anaconda เป็นโปรแกรมโอเพ่นซอร์สที่ช่วยในการจัดการทรัพยากรและสภาพแวดล้อมการพัฒนาในงานวิทยาศาสตร์ข้อมูลและการเรียนรู้ของเครื่อง (Machine Learning) รองรับหลายแพลตฟอร์มทั้ง Windows, macOS และ Linux ผู้ใช้สามารถสร้างและจัดการสภาพแวดล้อมการพัฒนาหลายแบบได้อย่างง่ายดายผ่านคำสั่ง conda ทำให้ติดตั้งแพ็คเกจและไลบรารีที่จำเป็นได้ในแต่ละสภาพแวดล้อมโดยไม่เกิดปัญหาความเข้ากันไม่ได้ นอกจากนี้ยังมี Jupyter Notebook สำหรับการพัฒนาและทดสอบโค้ดแบบอินเทอร์แอคทีฟ ช่วยให้นักวิจัยและผู้พัฒนาทดลองและปรับแก้โค้ดได้สะดวก Anaconda เป็นเครื่องมือที่มีประสิทธิภาพและเป็นที่ยอมรับในวงการนี้



2.1.9 Pretrained Model

Pretrained models เป็นโมเดลที่ถูกฝึกฝนล่วงหน้าบนชุดข้อมูลขนาดใหญ่และมีประสิทธิภาพสูงในการประมวลผลและวิเคราะห์ข้อมูลในด้านต่าง ๆ เช่น การตรวจจับวัตถุ (Object Detection) การจำแนกภาพ (Image Classification) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing) การใช้โมเดลที่ถูกฝึกฝนล่วงหน้าเป็นที่ยอมรับในวงการปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง (Machine Learning) เนื่องจากช่วยลดเวลาและทรัพยากรในการฝึกฝนโมเดลใหม่จากศูนย์ โมเดลเหล่านี้สามารถนำไปใช้ในโปรเจกต์ต่าง ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพ นอกจากนี้ pretrained models ยังสามารถปรับแต่งให้เหมาะสมกับงานเฉพาะได้ง่าย ทำให้นักพัฒนาและนักวิจัยสามารถสร้างสรรค์นวัตกรรมใหม่ ๆ ได้โดยไม่ต้องเริ่มต้นจากศูนย์ ซึ่งเป็นการเพิ่มประสิทธิภาพในการพัฒนาและทดสอบโมเดลในสภาพแวดล้อมที่หลากหลาย

2.1.10 YOLO

YOLO (You Only Look Once) เป็นหนึ่งในโมเดลการตรวจจับวัตถุ (Object Detection) ที่ได้รับความนิยมและมีประสิทธิภาพสูง YOLO มีความโดดเด่นในการตรวจจับวัตถุได้อย่างรวดเร็วและแม่นยำในภาพเดียว โดยไม่ต้องแบ่งภาพออกเป็นส่วน ๆ การใช้งาน YOLO สามารถตอบสนองความต้องการของหลาย ๆ อุตสาหกรรม ไม่ว่าจะเป็นการรักษาความปลอดภัย การตรวจสอบคุณภาพสินค้า หรือการประยุกต์ใช้ในยานยนต์ไร้คนขับ หนึ่งในเวอร์ชันที่ได้รับความนิยมคือ YOLOv5 ซึ่งได้รับการพัฒนาให้มีความแม่นยำสูงขึ้นและมีประสิทธิภาพในการประมวลผลที่ดีกว่า YOLO เวอร์ชันก่อนหน้า YOLOv5 ถูกออกแบบมาให้มีขนาดโมเดลที่เล็กลงแต่ยังคง

ความสามารถในการตรวจจับวัตถุที่รวดเร็วและแม่นยำ โดยรองรับการใช้งานที่หลากหลายและง่ายต่อการนำไปประยุกต์ใช้ในโปรเจกต์ต่าง ๆ ของนักพัฒนาและนักวิจัย

2.1.11 TensorFlow

TensorFlow เป็นแพลตฟอร์มโอเพ่นซอร์สสำหรับการเรียนรู้ของเครื่อง (Machine Learning) ที่ถูกพัฒนาโดย Google Brain ทีม TensorFlow ได้รับการออกแบบมาเพื่อให้ นักพัฒนาและนักวิจัยสามารถสร้างและฝึกฝนโมเดลการเรียนรู้ของเครื่องได้อย่างง่ายดายและมีประสิทธิภาพ ด้วยการรองรับการประมวลผลแบบกระจาย (Distributed Computing) ทำให้ TensorFlow สามารถจัดการกับข้อมูลขนาดใหญ่และทำการฝึกฝนโมเดลที่ซับซ้อนได้อย่างรวดเร็ว TensorFlow มีเครื่องมือและไลบรารีที่ครบครันสำหรับการพัฒนาโมเดลต่าง ๆ เช่น การจำแนกภาพ (Image Classification) การตรวจจับวัตถุ (Object Detection) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing) นอกจากนี้ TensorFlow ยังรองรับการใช้งานบนหลายแพลตฟอร์ม ไม่ว่าจะเป็นคอมพิวเตอร์ส่วนบุคคล สมาร์ทโฟน หรือแม้แต่ระบบคลาวด์ ทำให้เป็นเครื่องมือที่มีความยืดหยุ่นและเหมาะสำหรับการพัฒนาแอปพลิเคชันในวงกว้าง



ภาพที่ 6 TensorFlow

2.1.12 ResNet50

ResNet50 เป็นโมเดลเครือข่ายประสาทเทียมที่ลึกซึ่งถูกออกแบบมาเพื่อการจำแนกภาพ (Image Classification) และการตรวจจับวัตถุ (Object Detection) ResNet50 ย่อมาจาก Residual Network และมีความลึกทั้งหมด 50 ชั้น โครงสร้างของ ResNet50 ถูกออกแบบมาเพื่อลดปัญหาการเสื่อมถอยของโมเดล (Degradation Problem) ที่มักเกิดขึ้นในโมเดลที่มีความลึกมาก โดยใช้แนวคิด Residual Learning ซึ่งช่วยให้การฝึกฝนโมเดลลึก ๆ ทำได้ง่ายขึ้นและมีประสิทธิภาพสูง โมเดลนี้ได้รับความนิยมอย่างแพร่หลายในงานด้านปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง (Machine Learning) เนื่องจากมีความแม่นยำสูงและสามารถนำไปประยุกต์ใช้ในงานต่าง ๆ ได้หลากหลาย นอกจากนี้ ResNet50 ยังถูกใช้เป็นโมเดลเบื้องต้น (Pretrained Model) สำหรับการฝึกฝนเพิ่มเติมในงานเฉพาะทาง ซึ่งช่วยลดเวลาและทรัพยากรในการพัฒนาโมเดลใหม่และเพิ่มความแม่นยำในการทำงานของระบบ

2.1.13 SSD MobileNet

SSD MobileNet เป็นโมเดลการตรวจจับวัตถุ (Object Detection) ที่ได้รับความนิยม เนื่องจากมีความสามารถในการตรวจจับวัตถุได้อย่างรวดเร็วและมีประสิทธิภาพสูง โมเดลนี้เป็นการรวมกันระหว่าง SSD (Single Shot MultiBox Detector) ซึ่งเป็นสถาปัตยกรรมที่มีประสิทธิภาพในการตรวจจับวัตถุหลายประเภทในภาพเดียว กับ MobileNet ซึ่งเป็นโมเดลที่มีขนาดเล็กและประหยัดพลังงาน เหมาะสำหรับการใช้งานบนอุปกรณ์ที่มีทรัพยากรจำกัด เช่น สมาร์ทโฟนและอุปกรณ์ IoT SSD MobileNet ถูกออกแบบมาเพื่อลดความซับซ้อนในการประมวลผล โดยยังคงความแม่นยำในการตรวจจับวัตถุในภาพ ทำให้นักพัฒนาและนักวิจัยสามารถนำไปใช้ในแอปพลิเคชันที่ต้องการการประมวลผลแบบเรียลไทม์ได้อย่างมีประสิทธิภาพ นอกจากนี้ SSD MobileNet ยังสามารถปรับแต่งให้เหมาะสมกับงานเฉพาะทางได้ง่าย ช่วยเพิ่มความยืดหยุ่นในการนำไปใช้ในหลากหลายอุตสาหกรรม

2.2 งานที่เกี่ยวข้อง (Related Works)

การตรวจจับแบบเรียลไทม์มีความสำคัญอย่างยิ่งในหลายภาคส่วน โดยเฉพาะอย่างยิ่งในด้านความปลอดภัยและการตรวจจับวัตถุหรือบุคคลในภาพหรือวิดีโอ หนึ่งในเทคนิคที่ได้รับความนิยมสำหรับการตรวจจับวัตถุแบบเรียลไทม์คือ YOLO (You Only Look Once) ซึ่งสามารถตรวจจับวัตถุในภาพหรือวิดีโอได้อย่างรวดเร็วและแม่นยำ (Redmon, Divvala, Girshick, & Farhadi, 2016) ในบริบทของการตรวจจับส่วนต่าง ๆ ของเครื่องจักร เช่น PLCnext, YOLO สามารถนำมาใช้เพื่อระบุและจัดหมวดหมู่ชิ้นส่วนต่าง ๆ ของเครื่องจักรได้แบบเรียลไทม์ ทำให้สามารถเฝ้าติดตามและตรวจสอบสถานะของเครื่องจักรได้อย่างรวดเร็วและแม่นยำ

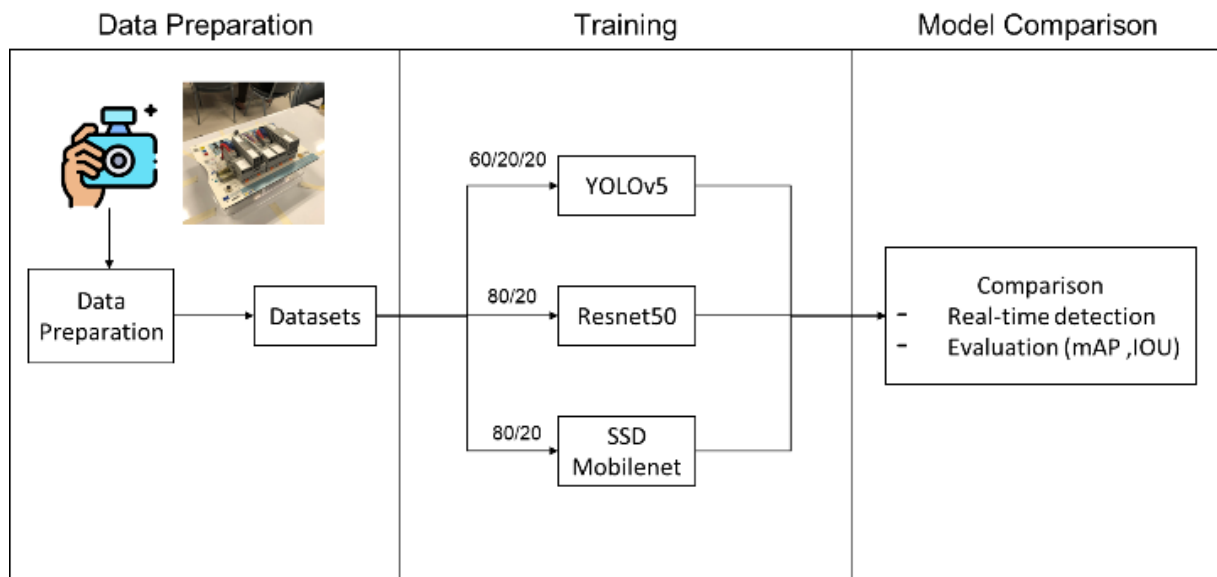
YOLO ได้รับการพัฒนาต่อเนื่องมาตั้งแต่เวอร์ชันแรก ผ่าน YOLOv2, YOLOv3 และ YOLOv4 โดยแต่ละเวอร์ชันได้มีการเพิ่มความเร็วและความแม่นยำในการตรวจจับวัตถุ (Bochkovskiy, Wang, & Liao, 2020) สำหรับ YOLOv5 มีการปรับปรุงในด้านความเร็วและความแม่นยำของการตรวจจับวัตถุ ทำให้เหมาะสมมากยิ่งขึ้นสำหรับการใช้ตรวจจับส่วนประกอบต่าง ๆ ของเครื่องจักรเช่น PLCnext (Yao et al, 2021)

การผสมรวมโมเดลที่ผ่านการฝึกอบรมเข้ากับ Unity 3D engine นั้นท้าทาย เว็บไซต์ของ GitHub มีทรัพยากรที่ยอดเยี่ยมสำหรับเรื่องนี้ (EnoxSoftware, 2023) เว็บไซต์นี้มีบทเรียนที่มีประโยชน์เกี่ยวกับการส่งออกโมเดล YOLO รวมถึงโค้ดที่พร้อมใช้งาน นอกจากนี้ยังมีเวอร์ชันอ้างอิงของ Unity ที่ทำงานร่วมกับโมเดลเหล่านี้ได้อย่างราบรื่น ทรัพยากรนี้พิสูจน์แล้วว่ามีความสำคัญอย่างยิ่ง

3 วิธีการดำเนินงานโครงการ

3.1 ภาพรวมของการดำเนินงาน

ผู้พัฒนาออกแบบและพัฒนาโมเดลปัญญาประดิษฐ์ โดยโครงสร้างในการพัฒนามีลักษณะดังนี้



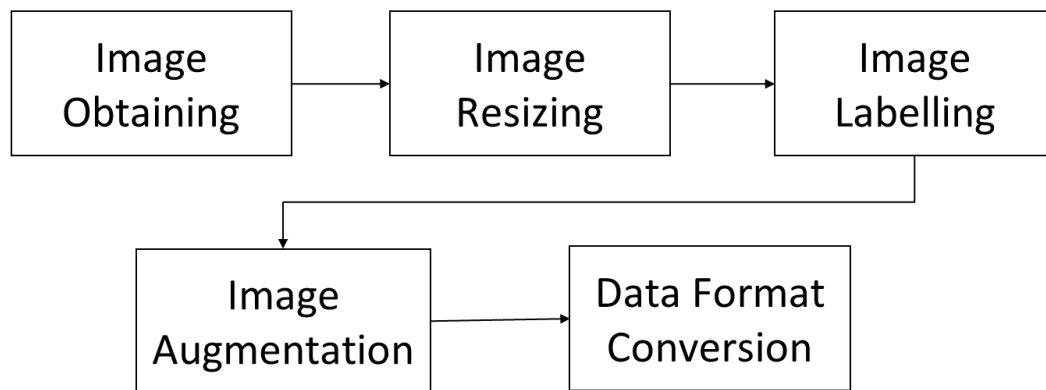
ภาพที่ 7 ภาพรวมของระบบ

ตามที่แสดงในภาพที่ 7 ภาพรวมของระบบทั้งกระบวนการถูกแบ่งออกเป็นสามส่วน ส่วนแรกเกี่ยวข้องกับการรวบรวมและเตรียมข้อมูลให้พร้อมสำหรับการฝึกและการทดสอบ ส่วนที่สองคือกระบวนการฝึกโมเดล ในกระบวนการนี้ ข้อมูลที่รวบรวมมาได้จะถูกนำไปฝึกกับโมเดลที่ถูกฝึกล่วงหน้าแล้วสามโมเดล เพื่อดูผลลัพธ์ในหลายๆ แ่งมุม โมเดลที่ถูกฝึกล่วงหน้าแล้วได้แก่ YOLOv5, Resnet50, และ SSD MobileNet ผลการทดสอบช่วยในการตัดสินใจว่าโมเดลใดเหมาะสมที่สุดสำหรับงานของเรา ส่วนสุดท้ายของภาพรวมระบบคือการเปรียบเทียบผลลัพธ์ของโมเดลที่ผ่านการฝึกทั้งสามโมเดล เกณฑ์การเปรียบเทียบมาจากลักษณะของโมเดลที่ต้องการ เช่น ความเร็วและความแม่นยำในการตรวจจับแบบเรียลไทม์ โมเดลที่ผ่านการฝึกและได้ผลลัพธ์ที่ดีที่สุดจะถูกใช้สำหรับงานในอนาคตต่อไป

นอกจากนี้ภาพที่ 7 แสดงภาพรวมของระบบที่มีการรวบรวม เตรียม และแบ่งข้อมูลออกเป็นชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบ ตามข้อกำหนดของแต่ละโมเดล ข้อมูลถูกแบ่งเป็น 60/20/20 (ฝึก/ตรวจสอบ/ทดสอบ) สำหรับ YOLOv5, 80/20 (ฝึก/ทดสอบ) สำหรับ Resnet50 และ 80/20 (ฝึก/ทดสอบ) สำหรับ SSD MobileNet ตามลำดับ จากนั้นผลลัพธ์ถูกนำมาเปรียบเทียบเพื่อหาว่าโมเดล AI ใดที่เหมาะสมที่สุดสำหรับการตรวจจับแบบเรียลไทม์ของ PLCnext

3.2 การเตรียมข้อมูล

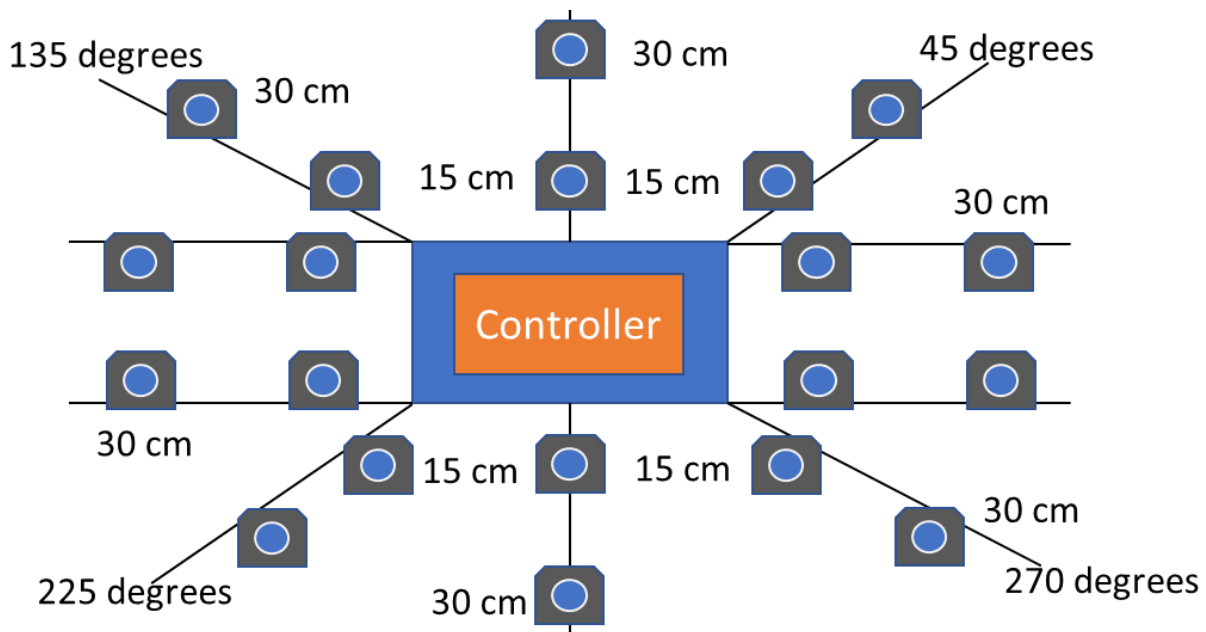
กระบวนการเตรียมข้อมูลสามารถดูได้จากภาพที่ 8 ด้านล่างนี้ โดยเริ่มต้นจากการรับภาพ (Image Obtaining) การปรับขนาดภาพ (Image Resizing) การติดป้ายกำกับภาพ (Image Labeling) การเสริมเพิ่มคุณลักษณะของภาพ (Image Augmentation) และการแปลงรูปแบบข้อมูล (Data Format Conversion)



ภาพที่ 8 การเตรียมข้อมูล

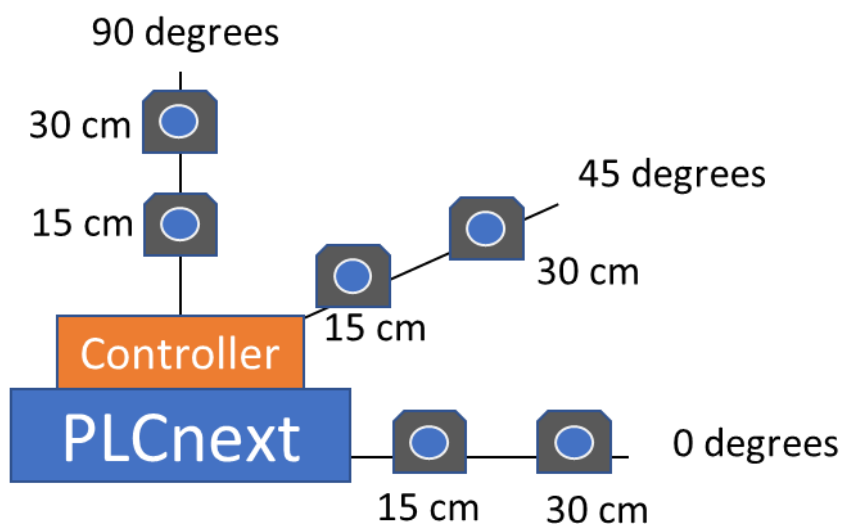
3.2.1 การเก็บรวบรวมภาพ (Image Obtaining)

ในส่วนของการเก็บรวบรวมภาพ เราได้รวบรวมภาพของ PLCnext จากมุมต่าง ๆ เพื่อให้ได้ชุดข้อมูลที่ครอบคลุมสำหรับการฝึก โมเดล ภาพที่ถ่ายมีความละเอียด 640x480 เนื่องจากขนาดนี้เหมาะสมกับการใช้งานกับโมเดลที่ถูกฝึกล่วงหน้า นอกจากนี้ เรายังทดลองใช้ความละเอียดที่สูงขึ้นที่ 3024x3024 เพื่อสังเกตความแตกต่างที่สำคัญ อย่างไรก็ตาม ผลที่ได้คือการฝึกโมเดลใช้เวลานานมากขึ้นเนื่องจากกระบวนการปรับขนาดภาพที่จำเป็นก่อนการรับรู้ นอกจากนี้ เรายังแบ่งการเก็บรวบรวมข้อมูลออกเป็นสองสถานการณ์ที่แตกต่างกันตามแสงและสีพื้นหลัง ในชุดแรก เราวาง PLCnext บนโต๊ะที่มีพื้นผิวสีขาวและใช้แสงจากเพดานตามปกติ ส่วนในชุดที่สอง เราวาง PLCnext บนโต๊ะที่มีพื้นหลังสีน้ำตาลและใช้ อุปกรณ์ให้แสงเพิ่มเติมเพื่อสร้างสภาพแวดล้อมที่แตกต่างจากชุดข้อมูลแรก รูปแบบและตำแหน่งของการถ่ายภาพสามารถดูได้ในภาพที่ 9



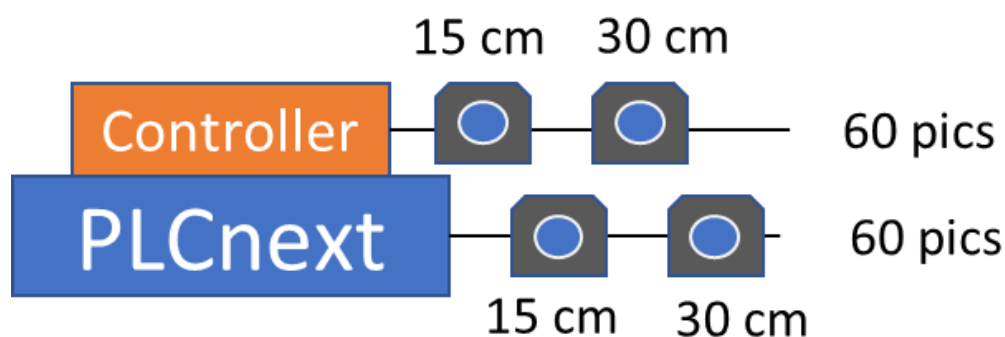
ภาพที่ 9 การเก็บภาพจากมุมมองด้านบน

ภาพที่ 9 แสดงมุมมองจากด้านบนของ PLCnext และตำแหน่งและมุมในการถ่ายภาพ จากภาพที่ 8 เราได้ตรวจสอบแกนแนวนอน ในระนาบนี้ เราได้กำหนดตำแหน่งสำหรับการถ่ายภาพทั้งหมด 10 จุดเพื่อครอบคลุมทั้งอุปกรณ์ PLCnext จะสังเกตได้ว่าในแต่ละจุดมีการถ่ายภาพที่ระยะห่าง 2 ระยะ คือ 15 เซนติเมตร และ 30 เซนติเมตร เพื่อให้โมเดลสามารถจดจำลักษณะของอุปกรณ์ PLCnext ได้ที่ระยะห่างต่าง ๆ



ภาพที่ 10 การเก็บภาพจากมุมมองระนาบแนวดิ่ง

ภาพที่ 10 แสดงมุมมองในการถ่ายภาพในระนาบแนวตั้ง ซึ่งมีทั้งหมดสามมุมมอง เมื่อเปรียบเทียบกับภาพที่ 9 จะพบว่าการถ่ายภาพหกภาพต่อแถวแนวตั้ง และเมื่อหมุนรอบ PLCnext เพื่อถ่ายภาพครบทุกมุมมอง ทำให้ได้ภาพทั้งหมด 60 ภาพ จากนั้น ในชุดเดียวกัน เราได้เปลี่ยนโพกัสจาก PLCnext ทั้งหมดไปยังส่วนที่เล็กกว่า เช่น คอนโทรลเลอร์ Digital I/O และ Analog I/O โดยยังคงใช้มุมมองในการถ่ายภาพเหมือนเดิม สิ่งที่เปลี่ยนแปลงคือระยะทางในการถ่ายภาพ ซึ่งในตอนแรกจะวัดระยะทางจาก PLCnext ทั้งหมด แต่ตอนนี้จะวัดระยะทางจากส่วนที่สนใจแบบเฉพาะเจาะจง ดังที่แสดงในภาพที่ 11



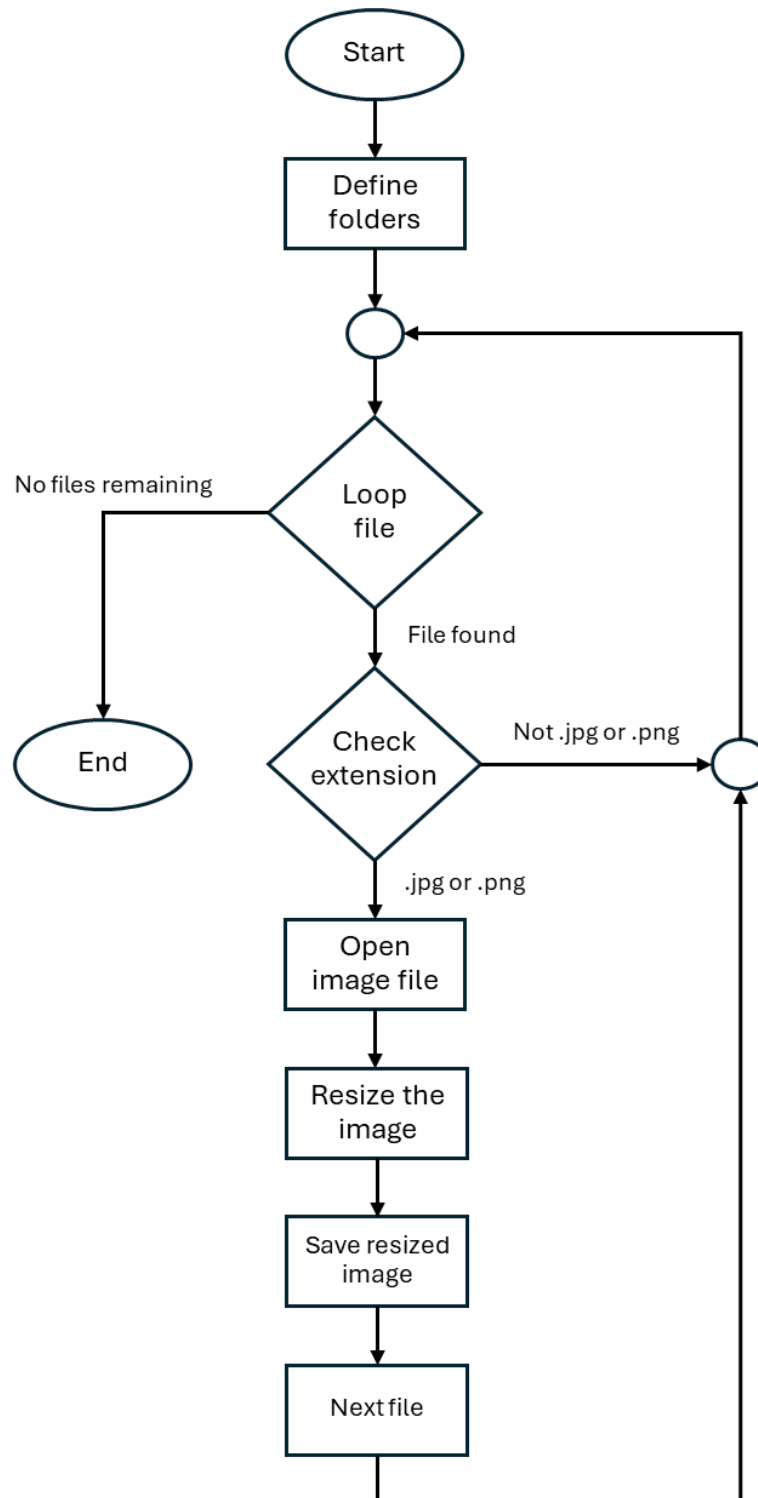
ภาพที่ 11 การวัดระยะห่างแบบเฉพาะส่วน

ดังนั้น การเก็บรวบรวมข้อมูลนี้ให้ผลลัพธ์เป็นภาพต้นฉบับทั้งหมด 120 ภาพจากแต่ละชุด รวมเป็น 240 ภาพ ภาพเหล่านี้จะนำไปสู่ขั้นตอนถัดไปที่มุ่งเน้นการเพิ่มขนาดชุดข้อมูล และเพิ่มประสิทธิภาพในการนำไปใช้งาน

3.2.2 การปรับขนาดภาพ (Image Resizing)

ภาพที่เราามีมีขนาดแตกต่างกัน โดยบางภาพมีความละเอียดสูงถึง 3024x3024 พิกเซล อย่างไรก็ตาม ภาพที่มีความละเอียดสูงเช่นนี้ไม่เหมาะสมสำหรับการฝึกเนื่องจากขนาดของมัน เพื่อให้กระบวนการฝึกเป็นไปอย่างมีประสิทธิภาพ เราจำเป็นต้องปรับขนาดของภาพเหล่านี้ ในกรณีนี้ โมเดลส่วนใหญ่ใช้ภาพที่มีขนาด 640x480 พิกเซลสำหรับการฝึก ดังนั้น เราจึงปรับขนาดภาพให้ใกล้เคียงกับขนาดนี้มากที่สุดเพื่อให้เหมาะสมกับการฝึกโมเดล

โดยกระบวนการนี้จะใช้วิธีการเขียนโค้ด Python เพื่อทำการปรับขนาดภาพทั้งหมดที่เตรียมไว้ในครั้งเดียว เพราะต้องการภาพในขนาดเดียวกัน Flowchart แสดงตัวอย่างการทำงานของโค้ดมีดังนี้



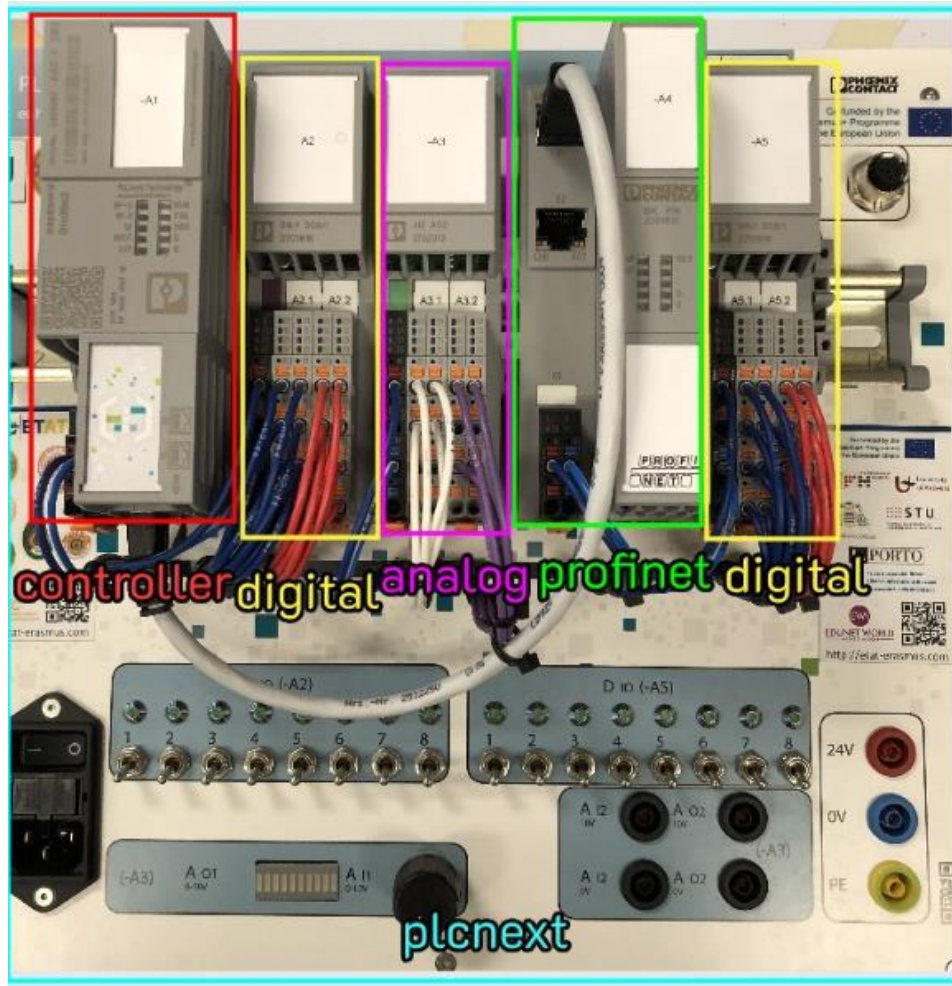
ภาพที่ 12 Flowchart แสดงการทำงานของโค้ดปรับขนาดรูปภาพ

Flowchart ของโค้ดปรับขนาดภาพ จากภาพที่ 12 มีกระบวนการดังนี้

- 1 Start (เริ่มต้น): เริ่มต้นกระบวนการ
- 2 Define folders (กำหนดโฟลเดอร์): กำหนดโฟลเดอร์สำหรับเก็บภาพต้นฉบับและโฟลเดอร์สำหรับเก็บภาพที่ถูกปรับขนาด
- 3 Loop file (วนลูปไฟล์): วนลูปผ่านไฟล์ทั้งหมดในโฟลเดอร์ที่กำหนด
- 4 Check extension (ตรวจสอบนามสกุลไฟล์): ตรวจสอบนามสกุลของไฟล์ว่าเป็นไฟล์ภาพหรือไม่ (เช่น jpg, png)
 - ถ้าไม่ใช่ไฟล์ภาพ (No): ข้ามไฟล์นั้นไปและทำงานกับไฟล์ถัดไป
 - ถ้าเป็นไฟล์ภาพ (Yes): ทำตามขั้นตอนต่อไป
- 5 Open image file (เปิดไฟล์ภาพ): เปิดไฟล์ภาพที่ต้องการปรับขนาด
- 6 Resize the image (ปรับขนาดภาพ): ปรับขนาดของภาพให้เป็นขนาดที่ต้องการ
- 7 Save resized image (บันทึกภาพที่ถูกปรับขนาด): บันทึกภาพที่ถูกปรับขนาดลงในโฟลเดอร์ที่กำหนด
- 8 Next file (ไฟล์ถัดไป): ทำงานกับไฟล์ถัดไปในลูป
- 9 End (สิ้นสุด): สิ้นสุดกระบวนการเมื่อไฟล์ทั้งหมดถูกประมวลผลแล้ว

3.2.3 การติดป้ายกำกับภาพ (Image Labeling)

หลังจากการเก็บรวบรวมและปรับขนาดภาพ เราได้ทำการติดป้ายกำกับวัตถุในภาพ ขั้นตอนนี้ช่วยให้โมเดลเรียนรู้ลักษณะและตำแหน่งของวัตถุ ในโครงการนี้ โมเดลควรได้รับการฝึกฝนเกี่ยวกับส่วนประกอบต่าง ๆ ที่ควบคุมการทำงานของ PLCnext ที่มีขนาดไม่เล็กเกินไป ส่วนประกอบเหล่านี้ถูกอธิบายในภาพที่ 13 ด้านล่าง ป้ายกำกับถูกกำหนดให้กับทั้งอุปกรณ์ PLCnext และส่วนประกอบย่อยต่าง ๆ ซึ่งรวมถึง คอนโทรลเลอร์ PLCnext, I/O แบบดิจิทัล, I/O แบบแอนะล็อก, และคอนโทรลเลอร์ Profinet นันนมหาความว่า การฝึกโมเดลจะมีการจำแนกคลาสทั้งหมดจำนวน 6 คลาส ได้แก่ controller (PLCnext Controller), digital (Digital In/Output), analog (Analog In/Output), profinet (Profinet Controller), และ plcnext (EDU PLCnext Kit)

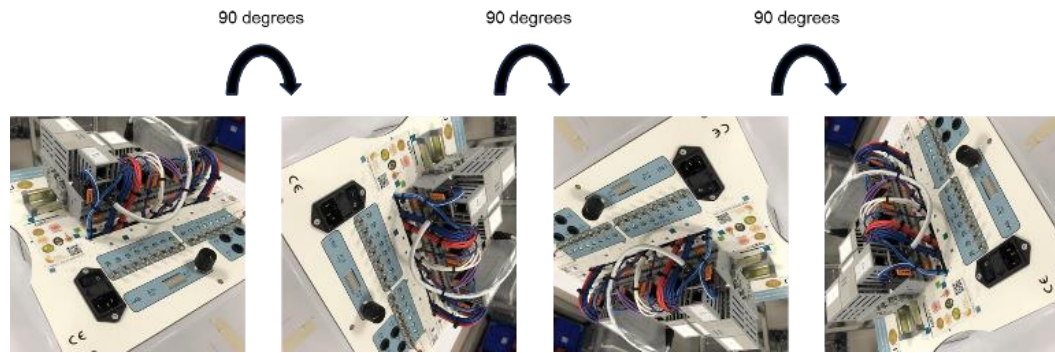


ภาพที่ 13 การติดป้ายกำกับรูปภาพ

สำหรับการติดป้ายกำกับ เราใช้โปรแกรมที่เรียกว่า LabelImg ซึ่งช่วยให้สามารถติดป้ายกำกับภาพสำหรับทั้งรูปแบบโมเดล YOLOv5 และ PascalVOC ที่ใช้ในการฝึกโมเดล Resnet50 และ SSD MobileNet ตามลำดับ

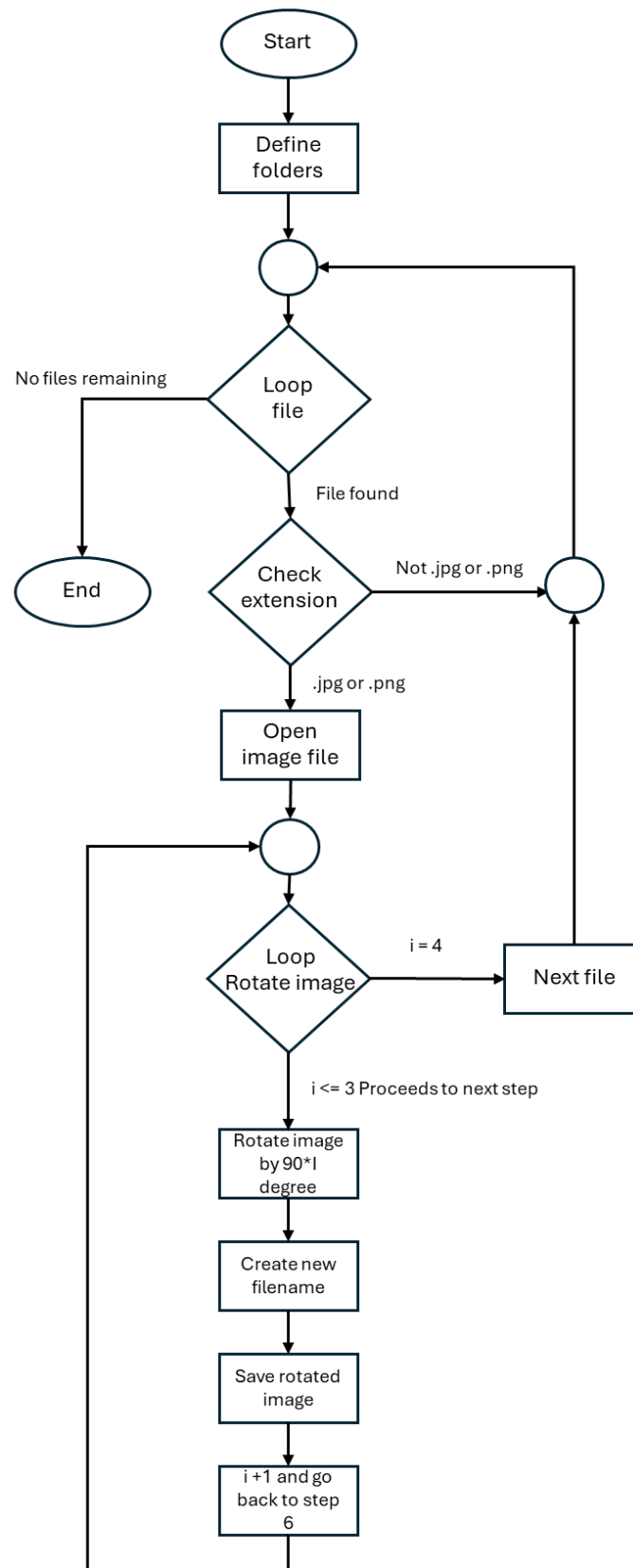
3.2.4 การเสริมเพิ่มคุณลักษณะของภาพ (Image Augmentation)

ขั้นตอนนี้ช่วยประหยัดเวลาในการเก็บรวบรวมข้อมูลและเพิ่มความหลากหลายให้กับชุดข้อมูลที่มีอยู่ เนื่องจากการเก็บข้อมูลเบื้องต้นมีเพียง 240 ภาพ เราจึงใช้กระบวนการนี้เพื่อเพิ่มจำนวนภาพ หลักการคือการหมุนภาพต้นฉบับ 90 องศาในแต่ละครั้ง ซึ่งทำให้ได้ภาพที่มีส่วนประกอบเดียวกันกับภาพต้นฉบับแต่มีตำแหน่งของวัตถุที่ต่างออกไป ดังนั้นภาพต้นฉบับแต่ละภาพสามารถหมุนได้สามครั้งเพื่อให้หมุนครบวงจร ซึ่งจะได้ภาพเพิ่มเติมสามภาพต่อภาพต้นฉบับ ดังที่แสดงในภาพที่ 14 ด้านล่าง



ภาพที่ 14 การสร้างภาพใหม่จากการหมุนรูปภาพต้นฉบับ

การหมุนรูปภาพนี้ใช้การทำงานของโค้ด Python ที่จะทำการหมุนรูปภาพทั้งหมดในโฟลเดอร์ที่เตรียมไว้ในลักษณะเดียวกันโดยมี Flowchart แสดงขั้นตอนการทำงานดังในภาพที่ 15 โดยในการทำงานในอนาคตนั้นสามารถใช้การเสริมเพิ่มคุณสมบัติของภาพอื่น ๆ เข้ามาเพิ่มเติมความหลากหลายของชุดข้อมูล เช่น การย่อ/ขยายขนาดของภาพ การตัดบางส่วน การทำภาพเอียง เป็นต้น



ภาพที่ 15 Flowchart แสดงการทำงานของโค้ดหมุนรูปภาพ

Flowchart ของโค้ดหมุนรูปภาพ จากภาพที่ 15 มีกระบวนการดังนี้

1. Start (เริ่มต้น): เริ่มต้นกระบวนการ
2. Define folders (กำหนดโฟลเดอร์): กำหนดโฟลเดอร์สำหรับเก็บภาพต้นฉบับและโฟลเดอร์สำหรับเก็บภาพที่หมุนแล้ว
3. Loop file (วนลูปไฟล์): วนลูปผ่านไฟล์ทั้งหมดในโฟลเดอร์ที่กำหนด
4. Check extension (ตรวจสอบนามสกุลไฟล์): ตรวจสอบนามสกุลของไฟล์ว่าเป็นไฟล์ภาพหรือไม่ (เช่น jpg, png)
 - ถ้าไม่ใช่ไฟล์ภาพ (No): ข้ามไฟล์นั้นไปและทำงานกับไฟล์ถัดไป
 - ถ้าเป็นไฟล์ภาพ (Yes): ทำตามขั้นตอนต่อไป
5. Open image file (เปิดไฟล์ภาพ): เปิดไฟล์ภาพที่ต้องการหมุน
6. Loop Rotate image (วนลูปหมุนภาพ): วนลูปผ่านการหมุนภาพ
7. Rotate image by $90^\circ * i$ degree (หมุนภาพตามองศาที่กำหนด)
8. Create new filename (สร้างชื่อไฟล์ใหม่): สร้างชื่อไฟล์ใหม่สำหรับภาพที่หมุนแล้ว
9. Save rotated image (บันทึกภาพที่หมุนแล้ว): บันทึกภาพที่หมุนแล้วลงในโฟลเดอร์ที่กำหนด
10. $i + 1$ and go back to step 6 (เพิ่มค่า i และกลับไปยังขั้นตอนที่ 6): เพิ่มค่า i แล้วกลับไปยังขั้นตอนที่ 6 เพื่อทำการหมุนภาพต่อไป
11. Next file (ไฟล์ถัดไป): ทำงานกับไฟล์ถัดไปในลูป
12. End (สิ้นสุด): สิ้นสุดกระบวนการเมื่อไฟล์ทั้งหมดถูกประมวลผลแล้ว

ความท้าทายที่พบในส่วนนี้คือเมื่อหมุนภาพที่มีอัตราส่วนไม่เท่ากัน (1:1) จะทำให้ขนาดของภาพเปลี่ยนแปลงไป ทำให้ต้องมีการคำนวณที่ซับซ้อนมากขึ้น ตัวอย่างเช่น ภาพขนาด 640x480 ซึ่งเป็นขนาดที่พบบ่อยในชุดข้อมูลนี้ เมื่อหมุนภาพครึ่งหนึ่งจะทำให้ได้ภาพใหม่ที่มีขนาด 480x640 ในส่วนนี้ เราใช้โค้ด Python เพื่อคำนวณตำแหน่งใหม่และบันทึกไว้ในไฟล์ txt ที่ระบุป้ายกำกับของภาพ การคำนวณทำโดยใช้สมการดังต่อไปนี้

$$x' = x \cos(90^\circ) - y \sin(90^\circ) \quad (1)$$

$$y' = x \sin(90^\circ) + y \cos(90^\circ) \quad (2)$$

สมการเหล่านี้ (1) และ (2) ถูกใช้ในการสลับพิกัด x และ y พร้อมกับเปลี่ยนเครื่องหมายของพิกัด x ใหม่ ซึ่งทำให้สามารถหมุนจุด 90 องศาตามเข็มนาฬิกาการรอบจุดกำเนิดได้อย่างมีประสิทธิภาพ สมการเหล่านี้ถูกนำมาใช้ในโค้ด Python สำหรับกระบวนการเตรียมข้อมูล

3.2.5 การแปลงรูปแบบข้อมูล (Data Format Conversion)

กระบวนการเตรียมข้อมูลที่กำลังกล่าวถึงข้างต้น เมื่อเสร็จสมบูรณ์แล้ว จะทำให้เราพร้อมที่จะฝึกโมเดลด้วย YOLOv5 อย่างไรก็ตาม สำหรับการฝึกโมเดลอีกสองโมเดล เราต้องมีไฟล์ชุดข้อมูลการฝึกที่ระบุป้ายกำกับของชุดข้อมูลในรูปแบบ PascalVOC ในกรณีนี้ เราสามารถปรับเปลี่ยนชุดข้อมูลที่มีอยู่ในโปรแกรม LabelImg โดยเปลี่ยนรูปแบบเป็น PascalVOC ซึ่งจะให้ไฟล์ XML ที่ระบุป้ายกำกับที่สามารถใช้ในการฝึกกับ Resnet50 และ SSD MobileNet ได้โดยตรง

3.3 การฝึกโมเดล

ในส่วนของการฝึกโมเดล เราใช้โน้ตบุ๊กที่มีสเปคดังนี้: CPU: 11th Gen Intel(R) Core(TM) i7-11800H @ 230GHz, GPU: NVIDIA GeForce RTX 3070 Laptop 8GB GPU, และ 32 GB RAM เราตั้งค่าการฝึกโมเดลให้มีค่าเหมือนกัน

เราแบ่งข้อมูลสำหรับการฝึกและการทดสอบ โดยมีความแตกต่างเล็กน้อยในการแบ่งข้อมูลระหว่าง YOLOv5 และโมเดลอื่น ๆ YOLOv5 ต้องการอัตราส่วนการแบ่งข้อมูล 60/20/20 สำหรับการฝึกทั่วไป ซึ่งใช้อัตราส่วนดังนี้: 60% ของข้อมูลใช้สำหรับฝึกโมเดล เพื่อให้โมเดลเรียนรู้ลักษณะและความสัมพันธ์ในข้อมูลเพื่อการทำนาย 20% ถัดไปใช้สำหรับการตรวจสอบความถูกต้อง โดยทดสอบโมเดลกับข้อมูลที่ไม่เคยเห็นมาก่อนเพื่อปรับแต่งและหาพารามิเตอร์ที่ดีที่สุด และสุดท้าย 20% สุดท้ายใช้สำหรับทดสอบโมเดลเพื่อประเมินประสิทธิภาพในสภาพแวดล้อมจริง การแบ่งข้อมูลออกเป็นสามส่วนนี้ทำให้มั่นใจได้ว่าโมเดลที่พัฒนาขึ้นมีประสิทธิภาพและสามารถนำไปใช้งานในสภาพแวดล้อมจริงได้อย่างมั่นใจ

แต่จะมีความแตกต่างสำหรับโมเดล Resnet50 และ SSD MobileNet ที่จะมีอัตราส่วนการแบ่งข้อมูลเป็น 80% สำหรับการฝึก และ 20% ที่เหลือจะใช้เพื่อประเมินประสิทธิภาพของโมเดลหลังการฝึก

ภาพที่ 16 แสดงกระบวนการฝึกโมเดล AI ภายในสภาพแวดล้อม TensorFlow 2 วิธีการฝึกและการกำหนดค่าถูกจัดเตรียมไว้เป็นค่าพื้นฐาน (T Gilbert, 2023) ซึ่งจะถูกดำเนินการฝึกโมเดลผ่าน Anaconda Prompt ที่ได้กำหนดสภาพแวดล้อมที่จำเป็นเอาไว้แล้ว

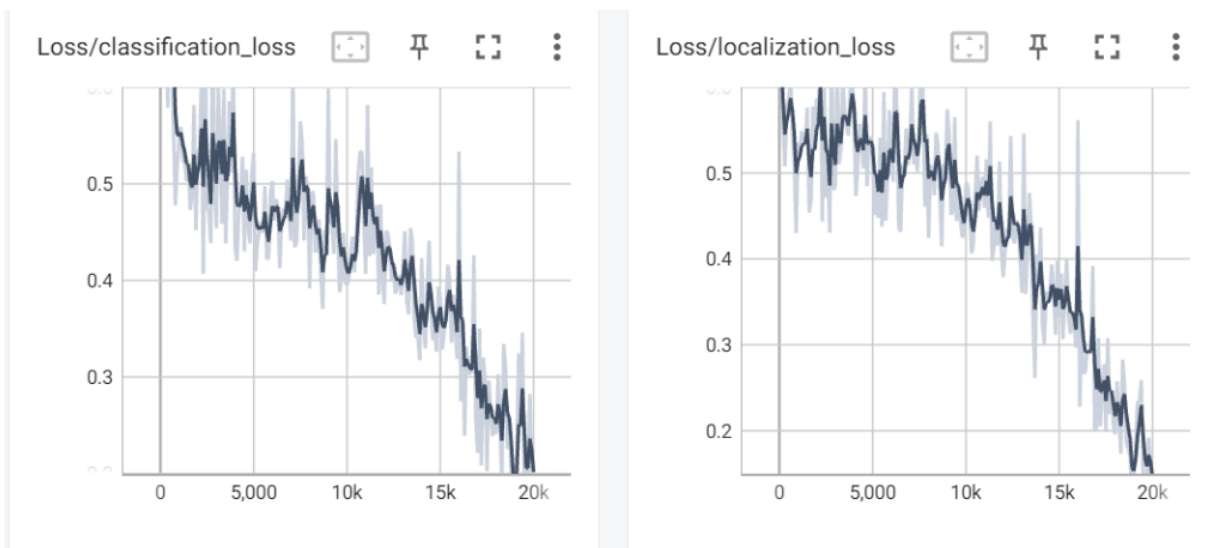
```

Annotated Prompt (Annotated)
{'learning_rate': 3.426075e-06}
I0927 08:23:23.837749 20916 model_lib_v2.py:708] {'Loss/classification_loss': 0.020388208,
'Loss/localization_loss': 0.013557735,
'Loss/regularization_loss': 0.060911715,
'Loss/total_loss': 0.094857655,
'learning_rate': 3.426075e-06}
INFO:tensorflow:Step 49900 per-step time 0.431s
I0927 08:24:06.938503 20916 model_lib_v2.py:705] Step 49900 per-step time 0.431s
INFO:tensorflow: {'Loss/classification_loss': 0.027434085,
'Loss/localization_loss': 0.0084671145,
'Loss/regularization_loss': 0.06091161,
'Loss/total_loss': 0.09681281,
'learning_rate': 8.559227e-07}
I0927 08:24:06.939499 20916 model_lib_v2.py:708] {'Loss/classification_loss': 0.027434085,
'Loss/localization_loss': 0.0084671145,
'Loss/regularization_loss': 0.06091161,
'Loss/total_loss': 0.09681281,
'learning_rate': 8.559227e-07}
INFO:tensorflow:Step 50000 per-step time 0.431s
I0927 08:24:50.075352 20916 model_lib_v2.py:705] Step 50000 per-step time 0.431s
INFO:tensorflow: {'Loss/classification_loss': 0.0262269,
'Loss/localization_loss': 0.011065709,
'Loss/regularization_loss': 0.060911596,
'Loss/total_loss': 0.0982042,
'learning_rate': 0.0}
I0927 08:24:50.076348 20916 model_lib_v2.py:708] {'Loss/classification_loss': 0.0262269,
'Loss/localization_loss': 0.011065709,
'Loss/regularization_loss': 0.060911596,
'Loss/total_loss': 0.0982042,
'learning_rate': 0.0}

```

ภาพที่ 16 กระบวนการฝึกอบรมโมเดล

หลังจากกระบวนการฝึกเริ่มขึ้น การแสดงภาพความก้าวหน้าในการเรียนรู้และการสูญเสียต่าง ๆ ของโมเดลสามารถดูได้ผ่านกราฟที่แสดงใน Tensorboard ภาพที่ 17 แสดง Tensorboard ของโมเดลที่กำลังถูกฝึก



ภาพที่ 17 กราฟแสดงค่าความสูญเสีย

ความหมายของค่าความสูญเสียแต่ละค่าสามารถอธิบายได้ดังนี้

1. การสูญเสียจากการจำแนกประเภท (Classification loss) หมายถึงส่วนหนึ่งของการคำนวณที่เกี่ยวข้องกับความแม่นยำของโมเดลในการทำนายประเภทที่ถูกต้องของวัตถุที่กำลังวิเคราะห์ ซึ่งแสดงถึงข้อผิดพลาดของโมเดลในการจำแนกประเภทของวัตถุ
2. การสูญเสียจากการระบุตำแหน่ง (Localization loss) หมายถึงข้อผิดพลาดที่เกี่ยวข้องกับความสามารถของโมเดลในการระบุตำแหน่งที่แม่นยำของวัตถุภายในภาพ ซึ่งเกี่ยวข้องกับความแม่นยำของพิกัดของกล่องที่ล้อมรอบวัตถุที่ระบุว่าวัตถุอยู่ที่ใดในพื้นที่

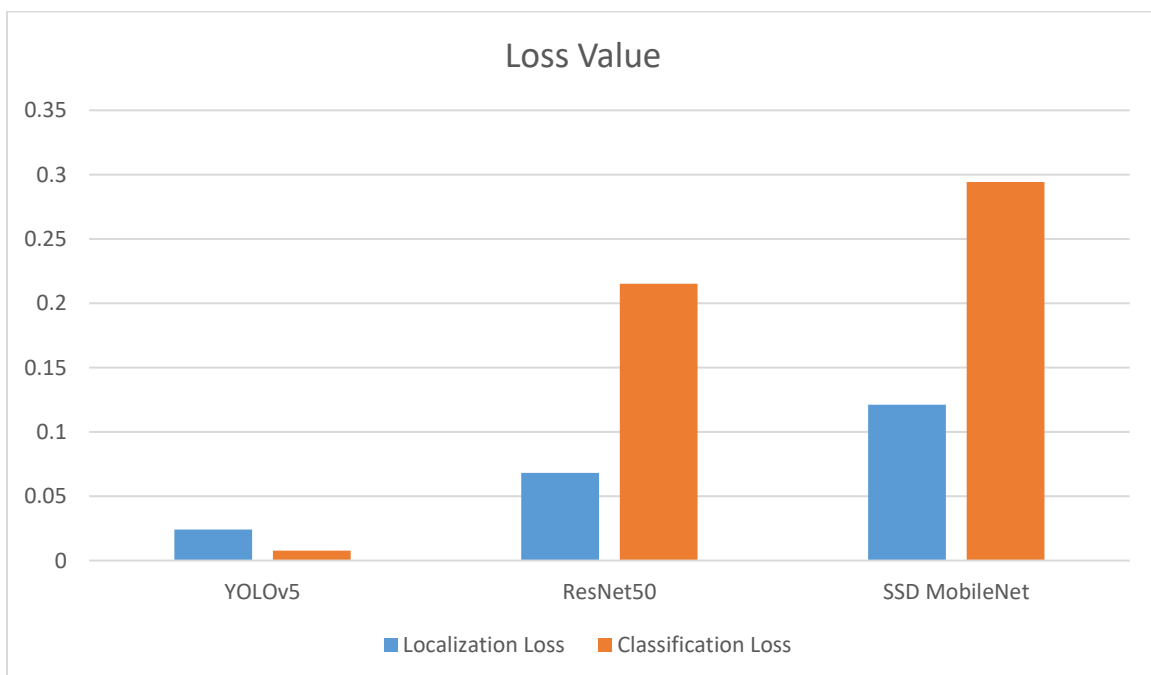
4 ผลการดำเนินการ

หลังจากฝึกโมเดล AI จนสำเร็จแล้ว การประเมินประสิทธิภาพของโมเดลเป็นสิ่งสำคัญเพื่อระบุว่าโมเดลใดเหมาะสมที่สุดสำหรับงานเฉพาะ การวัดผลที่ได้จากการประเมินจะแสดงถึงจุดแข็งและจุดอ่อนของแต่ละประเภทโมเดล และให้ข้อมูลเชิงลึกเกี่ยวกับอัตราการเรียนรู้และค่าการสูญเสีย (Loss) ซึ่งบ่งบอกว่า dataset ปัจจุบันยังสามารถเรียนรู้เพิ่มเติมได้หรือไม่ หรือว่าโมเดลได้บรรลุประสิทธิภาพสูงสุดสำหรับข้อมูลที่กำหนดแล้ว

เริ่มต้นด้วยการพิจารณาค่าการสูญเสีย (Loss) เมื่อโมเดลเรียนรู้มากขึ้น ค่าการสูญเสียควรลดลงและเข้าใกล้ค่า 0 ให้มากที่สุด แต่ละค่าการสูญเสียจะแสดงถึงแง่มุมต่าง ๆ ของกระบวนการเรียนรู้

ตารางที่ 1 ค่าความสูญเสียของแต่ละโมเดล

Loss Value	Models		
	YOLOv5	ResNet50	SSD MobileNet
Localization Loss	0.0242	0.0681	0.1211
Classification Loss	0.0078	0.2151	0.2944



ภาพที่ 18 กราฟแสดงค่าความสูญเสียของแต่ละโมเดล

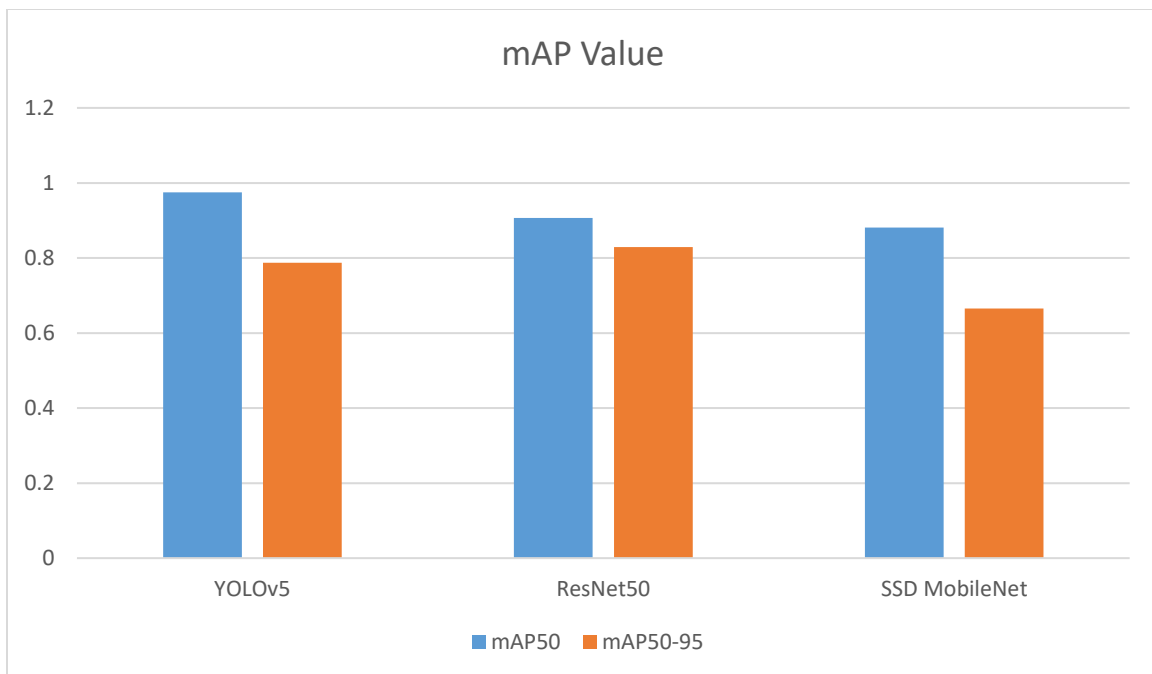
ในตารางที่ 1 สามารถสังเกตได้ว่าค่าการสูญเสียจากการฝึกแตกต่างกันอย่างมากในแต่ละสถาปัตยกรรม โดย YOLOv5 มีค่าการสูญเสียต่ำที่สุดในบรรดาโมเดลทั้งหมด แสดงถึงความแม่นยำที่เหนือกว่าในการทำนายคลาสและตำแหน่งของวัตถุโดยใช้ dataset เดียวกัน

mAP (Mean Average Precision) เป็นตัวชี้วัดสำคัญที่ใช้ในการประเมินประสิทธิภาพของโมเดลการตรวจจับและแบ่งส่วนวัตถุ ซึ่งรวมเอาความแม่นยำและการเรียกคืนของโมเดลมาไว้ในค่าเดียว ให้ภาพรวมที่กระชับเกี่ยวกับประสิทธิภาพของโมเดลในคลาสต่าง ๆ mAP ที่ค่า Intersection over Union (IoU) 0.50 (mAP50) คือกรณีที่ค่า IoU ถูกตั้งไว้ที่ 0.50 หมายความว่าในการตรวจจับที่ถูกต้อง ความซ้อนทับระหว่างกรอบที่ทำนายและกรอบความจริงบนพื้นต้องมียังน้อย 50% ในทางกลับกัน mAP ที่ค่า IoU ตั้งแต่ 0.50 ถึง 0.95 (mAP50-95) เป็นการประเมินที่เข้มงวดกว่า โดยเฉลี่ยค่า mAP จากหลายค่า IoU ตั้งแต่ 0.50 ถึง 0.95 ที่มีขั้นละ 0.05 ซึ่งให้ความเข้าใจที่ครอบคลุมมากขึ้นเกี่ยวกับประสิทธิภาพของโมเดลในระดับความซ้อนทับและขนาดของวัตถุที่แตกต่างกัน ทำให้ mAP50-95 เป็นตัวชี้วัดที่ทนทานมากขึ้นสำหรับการประเมินโมเดลในงานวิชาการ

การคำนวณรวมถึงการคำนวณค่า Precision และ Recall ที่เกณฑ์ความมั่นใจต่าง ๆ สำหรับแต่ละคลาส วัตถุ จากนั้นคำนวณค่า Average Precision (AP) สำหรับแต่ละคลาส และสุดท้ายเฉลี่ยค่า AP ในทุกคลาสเพื่อให้ได้ค่า mAP เครื่องมือต่าง ๆ และไลบรารี เช่น COCO Evaluation Toolkit สามารถช่วยในการคำนวณค่าชี้วัดเหล่านี้เพื่อช่วยในการประเมินโมเดลการตรวจจับและแบ่งส่วนวัตถุอย่างเข้มงวด

ตารางที่ 2 ค่า mAP ของแต่ละโมเดล

mAP Value	Models		
	YOLOv5	ResNet50	SSD MobileNet
mAP50	0.975	0.907	0.881
mAP50-95	0.787	0.829	0.665



ภาพที่ 19 กราฟแสดงค่า mAP ของแต่ละโมเดล

ค่าต่าง ๆ ในตารางที่ 2 เผยให้เห็นว่าแม้ ResNet50 จะมีค่าการสูญเสียที่สูงกว่า YOLOv5 แต่ความแตกต่างในพารามิเตอร์ทำให้ ResNet50 มีค่า Mean Average Precision (mAP) ที่สูงกว่าอย่างมีนัยสำคัญ นอกจากนี้ SSD MobileNet ซึ่งต้องการเวลาฝึกน้อยที่สุดและมีขนาดโมเดลเล็กที่สุดในงานนี้ ยังสามารถบรรลุผลลัพธ์ที่น่าพึงพอใจ ผลลัพธ์นี้ชี้ให้เห็นถึงข้อพิจารณาที่สำคัญในการตัดสินใจเลือกโมเดลสำหรับการใช้งานในอนาคต

5 สรุปผล

5.1 สรุปผลการดำเนินงาน

การศึกษานี้ได้วิจัยในโลกของการบำรุงรักษาอุตสาหกรรม โดยเน้นบทบาทสำคัญของกระบวนการตรวจสอบและซ่อมแซมอย่างรวดเร็วและแม่นยำโดยใช้เทคนิคการตรวจจับวัตถุ ในการทำเช่นนี้ได้มีการตรวจสอบโมเดลที่ถูกฝึกฝนล่วงหน้าที่แตกต่างกันสามโมเดล ผ่านการตรวจสอบอย่างเข้มงวดของโมเดล YOLOv5, Resnet50 และ SSD MobileNet พบว่าการใช้โมเดลที่ถูกฝึกฝนล่วงหน้าอย่างเหมาะสมนั้นเหมาะสมกับอุตสาหกรรมระบบอัตโนมัติ โมเดลเหล่านี้ไม่เพียงแต่ช่วยในกระบวนการตรวจจับ แต่ยังเพิ่มความแม่นยำ ทำให้เป็นทรัพย์สินสำคัญในสภาพแวดล้อมอุตสาหกรรม

5.2 ข้อค้นพบสำคัญ

1. โมเดลที่ถูกฝึกฝนล่วงหน้าที่มีประสิทธิภาพ: การศึกษาของเราเปิดเผยถึงประสิทธิภาพของโมเดลที่ถูกฝึกฝนล่วงหน้า โดยเฉพาะ YOLOv5, Resnet50 และ SSD MobileNet ในการตรวจจับวัตถุแบบเรียลไทม์ในบริบทของอุตสาหกรรม โมเดลเหล่านี้แสดงให้เห็นถึงความแม่นยำและประสิทธิภาพสูง ซึ่งชี้ให้เห็นถึงความเหมาะสมในการใช้งานในอุตสาหกรรมระบบอัตโนมัติ โดยเฉพาะอย่างยิ่งโมเดล YOLOv5

2. ข้อจำกัดของการใช้งานในปัจจุบัน: สิ่งสำคัญคือต้องทราบว่าข้อค้นพบในงานวิจัยนี้เนื่องจากการฝึกและทดสอบในสถานการณ์จำลองและยังไม่ได้รับการนำไปใช้งานในเครื่องจักรจริงหรืออุปกรณ์ PLCnext โมเดลเหล่านี้แสดงความสามารถในสภาพแวดล้อมที่ควบคุม ซึ่งเน้นถึงความจำเป็นในการทดสอบและปรับใช้งานเพิ่มเติมในแอปพลิเคชันอุตสาหกรรมจริง

5.3 ปัญหา อุปสรรค และข้อจำกัด

1. ขาดข้อมูลตัวอย่างกรณีที่ต้องซ่อมบำรุงที่ไม่ได้มาจากการใช้งานจริง
2. ข้อมูลที่ได้นำมาฝึกอบรมโมเดลเป็นเพียงการรู้จำลักษณะกายภาพภายนอกของเครื่อง PLCnext
3. จำนวนโมเดลที่นำมาเปรียบเทียบอาจน้อยเกินไป
4. มีความซับซ้อนในการพัฒนาโปรแกรมจริงเนื่องจากเครื่องมือที่ใช้หลาย ๆ ส่วนยังไม่มีให้นำมาใช้งานร่วมกันอย่างแพร่หลายมากนัก

5.4 ข้อเสนอแนะ

1. ควรทดลองใช้โมเดลที่หลากหลายมากกว่านี้
2. ควรเก็บชุดข้อมูลที่มีความแตกต่างทางกายภาพมากกว่านี้

3. ควรเพิ่มการทดลองการใช้งานโมเดลกับ Unity และ Hololens
4. บันทึก Screen capture ของแต่ละขั้นตอนที่ละเอียดมากกว่านี้

5.5 แนวทางในการพัฒนาในอนาคต

กลยุทธ์การประเมินที่ครอบคลุมเป็นสิ่งสำคัญเพื่อยืนยันประสิทธิภาพและความทนทานของระบบที่รวมกันในสภาพแวดล้อมอุตสาหกรรม แนวทางอย่างเป็นระบบจะรวมถึงการปรับใช้เทคโนโลยี AR, AI และ Automation (AAA) ที่ปรับแต่งในสถานการณ์อุตสาหกรรมที่หลากหลาย ครอบคลุมระดับการดำเนินงานที่แตกต่างกัน ความซับซ้อน และภาคส่วนต่าง ๆ ขอบเขตที่ขยายนี้จะทำให้แน่ใจได้ว่าโซลูชันมีความสามารถในการปรับตัวและทนทานต่อกรณีการใช้งานที่หลากหลาย

การประเมินประสิทธิภาพอย่างเข้มงวดจะเป็นสิ่งสำคัญ ซึ่งไม่เพียงแต่จะวัดความแม่นยำและความเร็วของอัลกอริทึมการตรวจจับวัตถุเท่านั้น แต่ยังรวมถึงการประเมินผลกระทบของระบบต่อประสิทธิภาพการดำเนินงาน การลดเวลาหยุดทำงาน และการเพิ่มผลผลิตโดยรวมอีกด้วย เมตริกความสำเร็จจำเป็นต้องเป็นแบบหลายมิติ ครอบคลุมด้านเชิงคุณภาพ เช่น ประสบการณ์ของผู้ใช้และความเข้าใจง่ายของอินเทอร์เฟซ โดยเฉพาะเมื่อใช้กับอุปกรณ์อย่าง HoloLens 2 การวัดเชิงปริมาณจะรวมถึงความแม่นยำของการซ้อนทับ AR ความน่าเชื่อถือของการคาดการณ์หรือคำแนะนำที่ขับเคลื่อนด้วย AI และความสามารถในการตอบสนองของระบบภายใต้สภาวะโหลดต่าง ๆ หรือในสถานการณ์วิกฤต

นอกจากนี้ การประเมินที่ครอบคลุมยังต้องคำนึงถึงความสามารถในการปรับขยายของเทคโนโลยีที่เกี่ยวข้อง โดยประเมินว่าพวกมันสามารถผสานรวมกับระบบอุตสาหกรรมที่มีอยู่ได้ดีเพียงใดและปรับตัวเข้ากับโครงสร้างพื้นฐานใหม่หรือที่ได้รับการอัปเดต นอกจากนี้ยังจำเป็นต้องทดสอบความยืดหยุ่นด้านความปลอดภัยทางไซเบอร์ของระบบ เพื่อให้แน่ใจว่ามีโปรโตคอลความปลอดภัยที่แข็งแกร่งเพื่อปกป้องข้อมูลการดำเนินงานที่สำคัญ การทดสอบและการประเมินแบบองค์รวมนี้มีความสำคัญต่อการระบุพื้นที่ที่อาจต้องปรับปรุง ขับเคลื่อนการปรับปรุงอย่างต่อเนื่อง และรับประกันว่าโซลูชันพร้อมสำหรับการนำไปใช้ในวงกว้างในเส้นทางสู่ระบบอัตโนมัติในอุตสาหกรรม 4.0

5.6 ผลงานวิจัยที่ได้รับการตีพิมพ์และเผยแพร่แล้ว

ผลงานนี้ได้รับการตีพิมพ์และเผยแพร่ที่ The 7th International Conference on Information Technology (InCIT2023) (<https://ieeexploreieeeorg/document/10413004/>)

Optimal AI Model for Industrial AR Application based on Object Detection

Kanuenjij Kubola
Faculty of Informatics
Burapha University
Chon Buri, Thailand
kubola@buu.ac.th

Bhusana Kongon
Faculty of Informatics
Burapha University
Chon Buri, Thailand
66910080@go.buu.ac.th

Prawit Boonmee
Faculty of Informatics
Burapha University
Chon Buri, Thailand
prawit@buu.ac.th

Prajaks Jitgermmanan
Faculty of Informatics
Burapha University
Chon Buri, Thailand
prajaks@buu.ac.th

Abstract— In the modern industries and manufacturing techniques, quick and precise maintenance is a vital concept. Several approaches fulfill these requirements, including the Augmented Reality (AR) technology. The AR technology, combined with real-time object detection techniques, offers a possible solution to improve and disrupt the maintenance techniques in the near future. For finding the most suitable AI model for this purpose, this study compared the pre-trained YOLOv5, Resnet50, and SSD MobileNet models in detecting PLCnext components. Using 960 images for training and testing purpose, the models were trained and tested, highlighting their real-time capabilities. The results show that the pre-trained YOLOv5 model is the best to enhance interactive monitoring, minimize downtime, and boost efficiency in industrial settings and suitable for AR application development in automation 4.0 after being trained with required images.

Keywords— Industrial Maintenance, Real-time Monitoring, Object Detection, YOLOv5, Augmented Reality

I. INTRODUCTION

In the modern industrial landscape, real-time monitoring and maintenance have become crucial for ensuring seamless operations. The advent of Augmented Reality (AR) technology has opened new avenues for interactive and efficient monitoring systems. Particularly, the integration of AR with real-time object detection can significantly enhance the maintenance and troubleshooting processes in industrial setups. In this endeavor, the focus of this paper is to explore and compare the performance of three prominent object detection models, namely YOLOv5, Resnet50, and SSD MobileNet, in identifying various components of a PLCnext controller in real-time. The motivation behind this study comes from the potential of leveraging accurate real-time object detection to facilitate AR-based interactive monitoring, which could, in turn, lead to reduced downtime and improved operational efficiency. By evaluating the effectiveness of the pre-trained models, namely YOLOv5, Resnet50, and SSD MobileNet, in detecting PLCnext components, this paper aims to provide insights into the feasibility and advantages of integrating such technologies with AR glasses (HoloLens 2) for a more intuitive and responsive industrial maintenance framework. Through this comparative analysis, we aspire to pave the way for more sophisticated AR-integrated monitoring systems that can

significantly contribute to the modernization of industrial maintenance protocols.

These three pre-trained models were chosen because they are quite popular and highly effective in identifying and classifying objects. However, each of these models has slight differences in their application, so we've compared them to highlight these distinct variations.

II. STATE OF THE ART

A. Real-time detection with YOLO

Real-time detection is significantly important across various sectors, especially in the realms of security and object or person detection within images or videos. One popular technique for real-time object detection is YOLO (You Only Look Once), which is capable of detecting objects in images or videos swiftly and accurately [1]. In the context of detecting different parts of a machine like PLCnext, YOLO can be employed to identify and categorize various machine components in real-time, enabling quick and precise monitoring and tracking of the machine's status.

YOLO has seen continuous development since its initial version, through YOLOv2, YOLOv3, and YOLOv4, with each version enhancing the speed and accuracy of object detection [2]. For YOLOv5, improvements in terms of object detection speed and accuracy have been made, making it increasingly suitable for use in detecting various components of a machine like PLCnext [3].

B. YOLO in Unity

The integration of the mentioned pre-trained models into Unity 3D engine is challenging. The GitHub website and repository host excellent resources for this matter of topic [4]. This repository provides useful tutorials on exporting YOLO models, including ready-to-use code. Moreover, there was a Unity reference version that seamlessly worked with these models. This resource proved to be incredibly valuable.

III. METHODOLOGY

A. System Overview

As can be seen from Figure 1, the system overview of the entire process is divided into three parts. The first part involves collecting and preparing data to be ready for training and testing purposes. The second part is the model

training process. In this process, the collected data was taken and trained with the three different pre-trained models to see the results in various aspects. The pre-trained models are YOLOv5, Resnet50, and SSD MobileNet. The tested results enable the decision which model is the most suitable for our task. The last part of the system overview is the result comparing of these three trained models. The comparison criteria come from the desired model characteristics, such as speed and accuracy in real-time detection. The trained model with the best results will be used for the further tasks in the future.

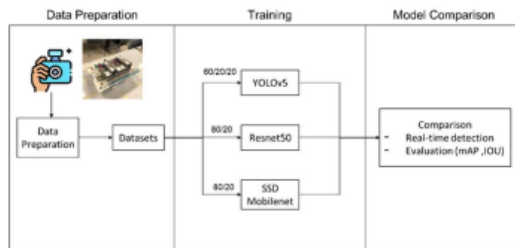


Fig. 1. System overview of the model training.

Fig. 1 shows the system overview where data was collected, prepared, and divided into training dataset and testing data set. According to the requirements of the each model, the data was divided into 60/20/20 (Train/Validate/Test) for YOLOv5, 80/20 (Train/Test) for Resnet50, and 80/20 (Train/Test) for SSD MobileNet respectively. Then, the results were compared to find the most suitable AI model for PLCnext real-time detection.

B. Data Preparation

The process of data preparation can be seen in the Figure 2 below. It begins with Image Obtaining, Image Resizing, Image Labeling, Image Augmentation, and Data Format Conversion.

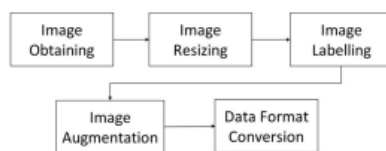


Fig. 2. Process of Data Preparation.

1) *Image Obtaining*: In the image collection part, we gathered images of PLCnext from various angles to ensure a comprehensive dataset for training. The images were captured at a resolution of 640x480 as this size is suitable for use with many pre-trained models. Additionally, we experimented with a higher resolution of 3024x3024 to observe any significant differences. However, the result was that training took much longer due to the necessary image resizing process before recognition. Furthermore, we divided the data collection into two different scenarios based on lighting and background color. In the first set, we placed the PLCnext on a table with a white surface and used regular ceiling lights. In the other set, we placed the PLCnext on a table with a brown background and used additional lighting equipment to create a different

environment from the first dataset. The patterns and positions of capturing images can be seen in the Figure 3.

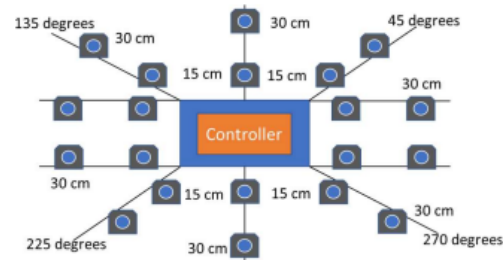


Fig. 3. The positions used to capture images of PLCnext from all sides.

Figure 3 shows the top view of the PLCnext and image taking positions and angles. From Figure 3, we examined the horizontal axis. In this plane, we have defined positions for capturing images at a total of 10 points to cover the entire PLCnext device. It can be observed that at each point, images are captured at 2 distances, which are 15 centimeters and 30 centimeters, in order to enable the model to recognize the characteristics of the PLCnext device at different distances.

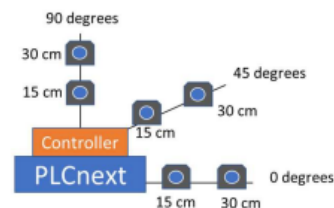


Fig. 4. The angles used to capture images of PLCnext in a vertical orientation.

Figure 4 illustrates the angles of image capture in the vertical plane, where the three angles are displayed in total. When compared to Figure 1, it equates to six images per vertical row, and upon rotating around the PLCnext to complete the circumference. This enabled 60 images to be obtained. Following this, within the same set, we shifted the focus from the entire PLCnext to zoom in on smaller sections such as the Controller, Digital I/O, and Analog I/O, while maintaining the same angles for capturing images. The only change was in the measurement of distance for capturing images. This means, initially, the distance was measured from the entire PLCnext, but now it's measured from the specific section of interest, as illustrated in Figure 5.

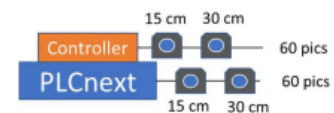


Fig. 5. The positions for capturing images when different focal points are in focus.

Therefore, this data collection yields a total of 120 original images from each set, summing up to 240 images.

These images will lead to the next steps aimed at increasing the dataset size and diversifying the data further.

2) *Image Resizing*: The images we have are of different sizes, with some being larger at 3024x3024 pixels. However, these high-resolution images are not suitable for training due to their size. To optimize our training process, we needed to resize these images. In this case, most models use images that are 640x640 pixels for training. Therefore, we resized the images to be as close as possible to this size for optimal training.

3) *Image Labeling*: Following the image collection and resizing, we labeled the objects in the images. This step allows the model to learn the features and positions of the objects. In this project, the model should be trained about the various components controlling the operations of the PLCnext that are not too small in size. These components are detailed in the Figure 6 below. The labels were assigned to both the PLCnext device and its various subcomponents, which includes PLCnext controller, digital I/Os, analog I/Os, and Profinet controller.

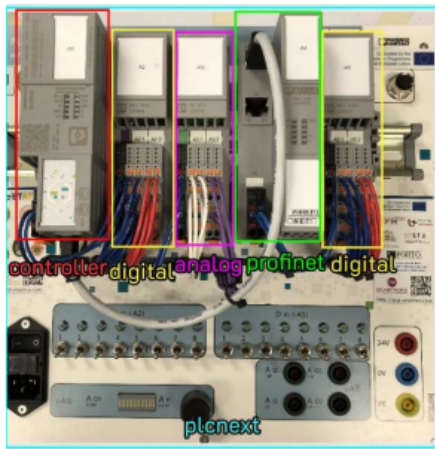


Fig. 6. The segmentation of the components of a PLCnext device.

For labeling purpose, we used the program called Labellmg [6], which allows labeling images for both YOLO and PascalVOC model formats, used for Resnet50 and SSD MobileNet training respectively.

4) *Image Augmentation*: This step provides time saving in data collection and adds variety to the existing dataset. Since the original collection only comprises 240 images, we used this process to increase the number of images. The principle here is to rotate the original images by 90 degrees each time, which results in images with the same components as the original but with different object positions. Therefore, each original image can be rotated three times to complete a full rotation, yielding three additional images per original image. This can be seen in Figure 7 below.

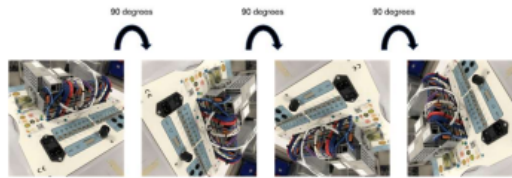


Fig. 7. Image Augmentation by rotating the image by 90 degrees.

The challenge encountered in this section was when rotating images that do not have a 1:1 ratio, it results in altered image dimensions, requiring more complex calculations. For instance, a 640x480 image, which is a common size in this dataset, when rotated once, results in a new image size of 480x640. In this section, we use Python code to calculate the new positions and save them in a txt file that specifies the labels of the images. The calculations were done using the following equations.

$$x' = x \cos(90^\circ) - y \sin(90^\circ) \quad (1)$$

$$y' = x \sin(90^\circ) + y \cos(90^\circ) \quad (2)$$

These equations (1) and (2) were used to swap the x and y coordinates while changing the sign of the new x-coordinate. This enabled the effectively rotating the point 90 degrees clockwise around the origin. These equations were used in the Python programming for data preparation process.

5) *Data Format Conversion*: The data preparation process mentioned above, once completed, brought us closer to train the model with YOLOv5. However, for training the other two models, the training dataset files specifying the labels of the dataset in PascalVOC format are needed. In this case, the existing dataset could be modified in Labellmg with the change of the format to PascalVOC. This yielded the XML files specifying labels that can be used for training with Resnet50 and SSD MobileNet directly.

C. Training

In the training section, we utilized a laptop with the following specifications: CPU: 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, GPU: NVIDIA GeForce RTX 3070 Laptop 8GB GPU, and 32 GB RAM. We configured the settings for each model to have similar values.

We divided the data for training and testing purposes. However, there was a slight difference in data splitting for YOLOv5 and other models. YOLOv5 requires a data split ratio of 60/20/20 for general purposes. This data ratio was used as follows: 60% of the data is used for training the model, where the model learns the features and relationships in the data to make predictions. The next 20% is used for validation, where the model is tested with unseen data to fine-tune and find the best parameters. Lastly, the final 20% is used for testing the model to evaluate its performance in real-world conditions. Splitting the data into these three parts ensures that the developed model performs well and generalizes to unseen data, giving us confidence in deploying the model in real environments. The Resnet50 and SSD MobileNet model will have a data split ratio of 80% for training, and the remaining 20% will be used to evaluate the model's performance post-training.

Figure 8 illustrates the training process of an AI model within the TensorFlow 2 environment. The training method and configuration was provided by T. Gilbert [5].

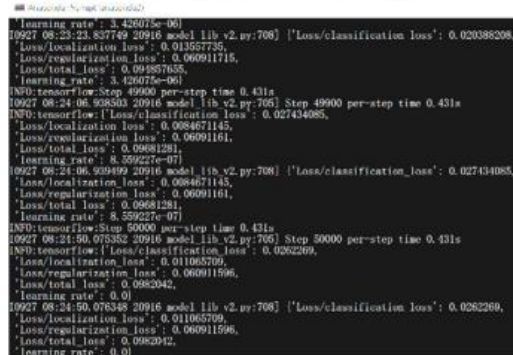


Fig. 8. The model is being trained.

After the training process began, the visualization of the learning progress and various losses of a model can be seen through graphs displayed in the Tensorboard. Figure 9 shows the Tensorboard of a model being trained.

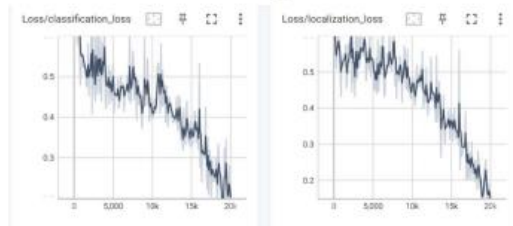


Fig. 9. Tensorboard displays various metrics obtained during the training process.

From Figure 9, the loss values obtained from the three trained models were compared in Table I.

1. Classification loss refers to the part of the calculation that deals with the model's accuracy in predicting the correct category of the objects it's analyzing. It represents the model's error in classifying the objects.

2. Localization loss refers to the error associated with the model's ability to precisely pinpoint the location of objects within an image. This involves the accuracy of the bounding box coordinates that identify where an object is in the space.

IV. RESULTS

After training the AI model successfully, it is essential to evaluate its performance to determine which variant is the most suitable for the propose specific task. The metrics obtained from the evaluation reveal the strengths and weaknesses of each model type. They provide insights into the learning rate and the losses, indicating whether the current dataset allows further learning or if the model has reached its optimal performance for the given data.

Let's take a look at the loss values first. As the model learns more, these values should ideally decrease and approach as close to 1 as possible. Each loss value indicates different aspects of the learning process.

TABLE I. LOSS VALUES OF DIFFERENT MODELS.

Loss Value	Models		
	YOLOv5	ResNet50	SSD MobileNet
Localization Loss	0.0242	0.0681	0.1211
Classification Loss	0.0078	0.2151	0.2944

In Table I, it can be observed that the losses incurred from training vary significantly across different architectures. YOLOv5 exhibits the lowest loss among all models, indicating superior accuracy in predicting object classes and positions using the same dataset.

The mAP (Mean Average Precision) is a crucial metric used to evaluate the performance of object detection and segmentation models. It encapsulates both the precision and recall of a model into a single value, providing a concise overview of the model's performance across different object classes. The mAP at Intersection over Union (IoU) of 0.50 (mAP50) is a specific instance where the IoU threshold is set at 0.50. This means that for a positive detection, the overlap between the predicted bounding box and the ground truth bounding box needs to be at least 50%. On the other hand, mAP at IoU ranging from 0.50 to 0.95 (mAP50-95) is a more rigorous evaluation where the mAP is averaged over multiple IoU thresholds from 0.50 to 0.95 with a step of 0.05. This provides a more comprehensive understanding of a model's performance across varying levels of overlap and object sizes, making mAP50-95 a more robust metric for evaluating models in academic papers. The computation involves calculating the precision and recall at various confidence thresholds for each object class, then computing the Average Precision (AP) for each class, and finally averaging the APs across all classes to obtain the mAP. Various tools and libraries, such as the COCO Evaluation Toolkit, can assist in computing these metrics to aid in the rigorous evaluation of object detection and segmentation models.

TABLE II. MAP50 AND MAP50-95 VALUE OF DIFFERENT MODELS.

Loss Value	Models		
	YOLOv5	ResNet50	SSD MobileNet
mAP50	0.975	0.907	0.881
mAP50-95	0.787	0.829	0.665

The values in Table II reveals that despite the higher loss values of ResNet50 compared to YOLOv5, the difference in parameters leads to significantly higher Mean Average Precision (mAP) values. Additionally, SSD MobileNet, which requires the least training time and has the smallest model size in this task, also achieves respectable results. This finding highlights a crucial consideration for decision-making in choosing a model for future implementations.

V. CONCLUSION AND FUTURE WORK

In conclusion, this study researched into the world of industrial maintenance, emphasizing the pivotal role of swift and precise monitoring and repair processes using Object Detection techniques. To enable this, three different pre-trained models were examined. Through our rigorous examination of YOLOv5, Resnet50, and SSD MobileNet models, we found a key insight: the reasonable use of pre-

trained models is greatly suitable for the automation industry. These models not only facilitate the detection process but also enhance accuracy, making them important assets in industrial settings.

A. Key Findings

1) *Effective Pre-trained Models:* Our study revealed the effectiveness of pre-trained models, particularly YOLOv5, Resnet50, and SSD MobileNet, in real-time object detection within the industrial context. These models showcased high accuracy and efficiency, suggesting their suitability for industrial automation applications, especially the YOLOv5 model.

2) *Limitation of Current Implementation:* It is crucial to note that our paper's findings are based on training and testing scenarios and have not been implemented in practical machinery or PLCnext devices. The models showcased their ability in controlled environments, highlighting the need for further testing and adaptation in real-world industrial applications.

B. Future Work

In the future work, a comprehensive evaluation strategy is essential to validate the efficacy and robustness of these integrated systems within industrial environments. A systematic approach will involve deploying the tailored AR, AI, and Automation (AAA) technologies in diverse industrial settings, encompassing varying operational scales, complexities, and sectors. This expanded scope ensures the adaptability and resilience of the solutions across a broad spectrum of use cases.

Conducting rigorous performance assessments will be pivotal. This entails not only measuring the accuracy and speed of object detection algorithms but also evaluating the system's impact on operational efficiency, downtime reduction, and overall productivity gains. Metrics for success need to be multi-dimensional, capturing qualitative aspects like user experience and interface intuitiveness, especially when used with devices such as HoloLens 2. Quantitative measures would include the precision of AR overlays, the reliability of AI-driven predictions or recommendations, and the responsiveness of the system under different load conditions or in crisis scenarios.

Furthermore, a comprehensive evaluation must also account for the scalability of the technologies involved, assessing how well they integrate with existing industrial systems and adapt to new or upgraded infrastructures. It is also imperative to test the system's cyber resilience, ensuring robust security protocols are in place to safeguard sensitive operational data. This holistic testing and evaluation phase is crucial for identifying potential areas for refinement, driving continuous improvement, and ensuring the solutions are ready for wider adoption in the journey towards full-scale Industrial Automation 4.0.

ACKNOWLEDGMENT

This research was conducted at the Digital Media and Interaction Research Laboratory (DMI), Faculty of

Informatics, Burapha University, and was co-funded by the Erasmus+ programme of the European Union (project ETAT Education & Training for Automation 4.0 in Thailand – n° 610154-EPP-1-2019-1-DE-EPPKA2-CBHE-JP). We express our sincere gratitude for the grant and the support provided.

REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.
- [2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.
- [3] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5," Electronics, vol. 10, no. 14, p. 1711, 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/14/1711>.
- [4] EnoxSoftware. "YOLOv5 With OpenCVForUnity Example." <https://github.com/EnoxSoftware/YOLOv5WithOpenCVForUnityExample> (accessed 1 september 2023, 2023).
- [5] G. Tanner. "Tensorflow Object Detection with Tensorflow 2: Creating a custom model." <https://gilberttanner.com/blog/tensorflow-object-detection-with-tensorflow-2-creating-a-custom-model/> (accessed 7 september 2023, 2023).
- [6] HumanSignal. "LabelImg." <https://github.com/qaprossoft/labelImg#labelimg> (accessed 03 July 2023, 2023).

6 บรรณานุกรม

References

- Redmon, J, Divvala, S, Girshick, R, & Farhadi, A (2016) You only look once: Unified, real-time object detection In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp 779-788)
- Bochkovskiy, A, Wang, C Y, & Liao, H Y M (2020) YOLOv4: Optimal speed and accuracy of object detection *arXiv preprint arXiv:200410934*
- Yao, J, Qi, J, Zhang, J, Shao, H, Yang, J, & Li, X (2021) A real-time detection algorithm for Kiwifruit defects based on YOLOv5 *Electronics*, 10(14), 1711
- EnoxSoftware (2023) *YOLOv5 With OpenCVForUnity Example* Retrieved September 1, 2023, from <https://github.com/EnoxSoftware/YOLOv5WithOpenCVForUnityExample>
- Tanner, G (2023) *Tensorflow Object Detection with Tensorflow 2: Creating a custom model* Retrieved September 7, 2023, from <https://gilberttanner.com/blog/tensorflow-object-detection-with-tensorflow-2-creating-a-custom-model/>
- HumanSignal (2023) *LabelImg* Retrieved July 3, 2023, from <https://github.com/qaprosoft/labelImg>
- Chiang, F-K, Shang, X, & Qiao, L (2022) Augmented reality in vocational training: A systematic review of research and applications *Computers in Human Behavior*, 129, 107125
- PLCnext Community (2023, June 9) PLCnext Community Website Retrieved from <https://www.plcnext-community.net/>
- Sureshkumar, S, Agash, C P, Ramya, S, Kaviyaraj, R, & Elanchezhian, S (2021) Augmented Reality with Internet of Things In IEEE <https://doi.org/10.1109/ica5093020219395941>
- Wikipedia (2023) Augmented reality Retrieved from https://en.wikipedia.org/w/index.php?title=Augmented_reality&oldid=1161819338
- World Economic Forum (2023, June 25) 3 ways Augmented Reality can help us with COVID-19 and beyond Retrieved from <https://www.weforum.org/agenda/2020/04/augmented-reality-covid-19-positive-use/>

7 ภาคผนวก

7.1 อธิบาย Code Python เบื้องต้น

7.1.1 โค้ดปรับขนาดรูปภาพ

```
import os
from PIL import Image

def resize_images_in_folder(folder_path, size=(640, 640)):
    for filename in os.listdir(folder_path):
        if filename.endswith(".jpg") or filename.endswith(".png"):
            img_path = os.path.join(folder_path, filename)
            with Image.open(img_path) as img:
                img = img.resize(size, Image.ANTIALIAS)
                img.save(img_path)
                print(f"Resized {filename}")

folder_path = 'path_to_your_folder' # เปลี่ยนเส้นทางไปยังโฟลเดอร์ของคุณ
resize_images_in_folder(folder_path)
```

ภาพที่ 20 โค้ดปรับขนาดรูปภาพ

1. การนำเข้าไลบรารีที่จำเป็นซึ่งในที่นี้จะมี 2 ตัวหลัก ได้แก่
 - 1.1 os เป็นไลบรารีที่ใช้จัดการไฟล์
 - 1.2 PIL (Python Imaging Library) เป็นไลบรารีที่ใช้ทำงานกับรูปภาพ

```
import os
from PIL import Image
```

ภาพที่ 21 นำเข้าไลบรารี os และ PIL

2. สร้างฟังก์ชัน “resize_images_in_folder” ที่รับพารามิเตอร์สองตัว คือ “folder_path” (เส้นทางของโฟลเดอร์ที่มีรูปภาพ) และ size (ขนาดใหม่ของรูปภาพ, ค่าเริ่มต้นคือ 640x480 พิกเซล)

```
def resize_images_in_folder(folder_path, size=(640, 640)):
```

ภาพที่ 22 สร้างฟังก์ชัน resize_images_in_folder

3. วนลูปเพื่อตรวจสอบไฟล์ทั้งหมดในโฟลเดอร์เพื่อตรวจหาไฟล์รูปภาพที่มีนามสกุล “jpg” หรือ “png”

```
for filename in os.listdir(folder_path):  
    if filename.endswith(".jpg") or filename.endswith(".png"):
```

ภาพที่ 23 วนลูปหาไฟล์รูปภาพ

4. สร้างเส้นทางไฟล์เต็มของรูปภาพโดยใช้ “ospathjoin” แล้วเปิดรูปภาพด้วย “Imageopen” จากนั้นทำการ Resize รูปภาพด้วย “imgresize” โดยมีพารามิเตอร์ “size” และ “ImageANTIALIAS” เพื่อรักษาคุณภาพของรูปภาพ

```
img = img.resize(size, Image.ANTIALIAS)  
img.save(img_path)  
print(f"Resized {filename}")
```

ภาพที่ 24 เปิดรูปภาพและทำการปรับขนาด

5. บันทึกรูปภาพที่ได้รับการปรับขนาดกลับไปยังที่เดิมโดยใช้ “imgsave” แล้วทำการแสดงผลว่าไฟล์ไหนถูกปรับขนาดแล้วบ้าง

```
img.save(img_path)  
print(f"Resized {filename}")
```

ภาพที่ 25 บันทึกภาพที่ได้รับการปรับขนาดแล้ว

6. กำหนดเส้นทางของโฟลเดอร์ที่ต้องการปรับขนาดรูปภาพภายในนั้น และ ทำการเรียกใช้ฟังก์ชันปรับขนาดรูปภาพ

```
folder_path = 'path_to_your_folder'
resize_images_in_folder(folder_path)
```

ภาพที่ 26 กำหนดเส้นทางของโฟลเดอร์ และ เรียกใช้ฟังก์ชันปรับขนาด

7.1.2 โค้ดหมุนรูปภาพ

```
import os
from PIL import Image

# กำหนดโฟลเดอร์ต้นทางและปลายทาง
source_folder = "path/to/source/folder"
destination_folder = "path/to/destination/folder"

# Loop ผ่านแต่ละไฟล์ในโฟลเดอร์ต้นทาง
for filename in os.listdir(source_folder):
    if filename.endswith(".jpg") or filename.endswith(".png"): # ตรวจสอบนามสกุลไฟล์
        image_path = os.path.join(source_folder, filename)
        image = Image.open(image_path)

        # Loop หมุนภาพ 3 ครั้ง
        for i in range(1, 4):
            rotated_image = image.rotate(90 * i) # หมุนภาพ 90, 180, 270 องศา

            # สร้างชื่อไฟล์ใหม่โดยต่อท้ายด้วยองศาที่หมุน
            new_filename = f"{os.path.splitext(filename)[0]}_{90 * i}.{os.path.splitext(filename)[1][1:]}"

            # บันทึกภาพที่หมุนแล้วลงในโฟลเดอร์ปลายทาง
            rotated_image.save(os.path.join(destination_folder, new_filename))
```

ภาพที่ 27 โค้ดหมุนรูปภาพ

1. การนำเข้าไลบรารีที่จำเป็นซึ่งในที่นี้จะมี 2 ตัวหลัก ได้แก่

1.1 os เป็นไลบรารีที่ใช้จัดการไฟล์

1.2 PIL (Python Imaging Library) เป็นไลบรารีที่ใช้ทำงานกับรูปภาพ

```
import os
from PIL import Image
```

ภาพที่ 28 นำเข้าไลบรารี os และ PIL

2. กำหนดเส้นทางของโฟลเดอร์ต้นทางและปลายทาง

```
# กำหนดโฟลเดอร์ต้นทางและปลายทาง
source_folder = "path/to/source/folder"
destination_folder = "path/to/destination/folder"
```

ภาพที่ 29 กำหนดเส้นทางของโฟลเดอร์ต้นทางและปลายทาง

3. วนลูปผ่านแต่ละไฟล์ในโฟลเดอร์ต้นทาง และ ตรวจสอบว่าเป็นไฟล์รูปภาพหรือไม่

```
for filename in os.listdir(source_folder):
    if filename.endswith(".jpg") or filename.endswith(".png"):
```

ภาพที่ 30 วนลูปผ่านแต่ละไฟล์ในโฟลเดอร์ต้นทาง

4. เปิดไฟล์ภาพและวนลูปหมุนรูปภาพ 3 ครั้ง

```
image_path = os.path.join(source_folder, filename)
image = Image.open(image_path)

# Loop หมุนภาพ 3 ครั้ง
for i in range(1, 4):
    rotated_image = image.rotate(90 * i) # หมุนภาพ 90, 180, 270 องศา
```

ภาพที่ 31 เปิดไฟล์ภาพและวนลูปหมุนรูปภาพ

5. สร้างชื่อไฟล์ใหม่ต่อท้ายด้วยองศาที่หมุน

```
# สร้างชื่อไฟล์ใหม่โดยต่อท้ายด้วยองศาที่หมุน
new_filename = f"{os.path.splitext(filename)[0]}_{90 * i}.{os.path.splitext(filename)[1][1:]}"
```

ภาพที่ 32 สร้างชื่อไฟล์ใหม่

6. บันทึกภาพที่หมุนแล้วลงในโฟลเดอร์ปลายทาง

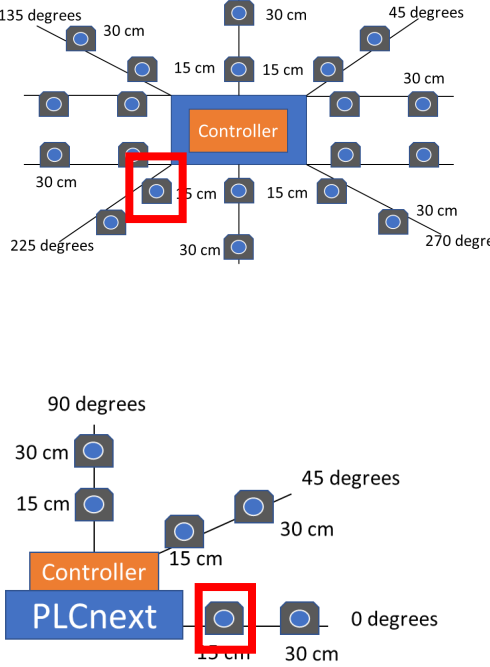
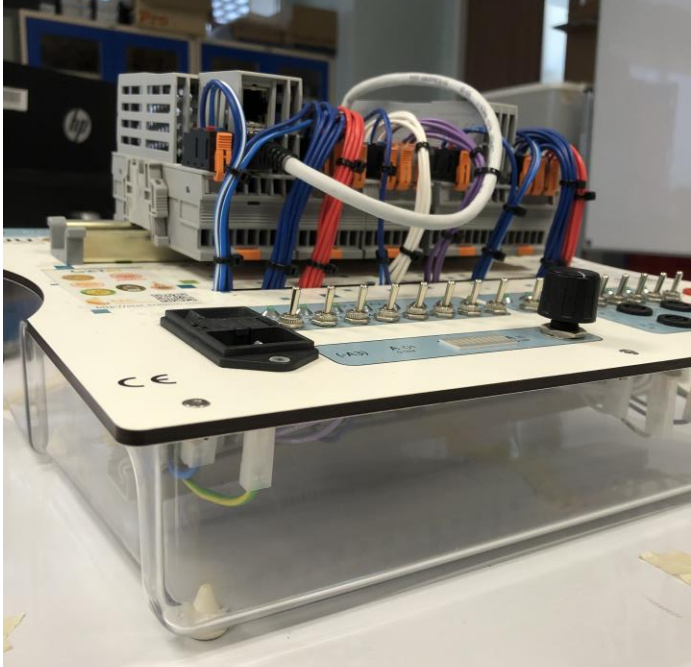
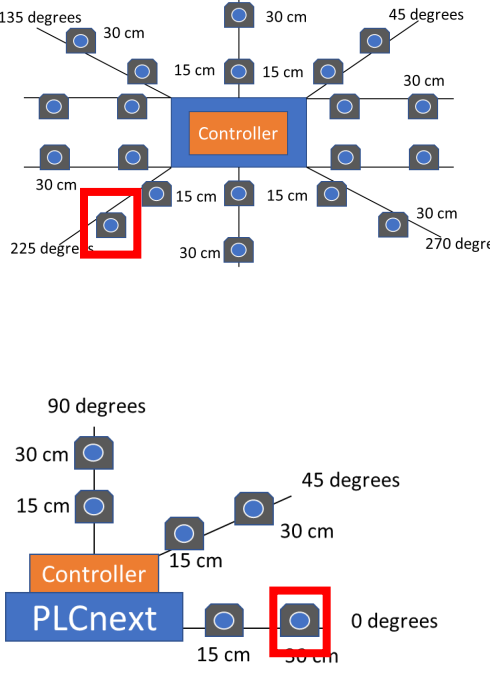
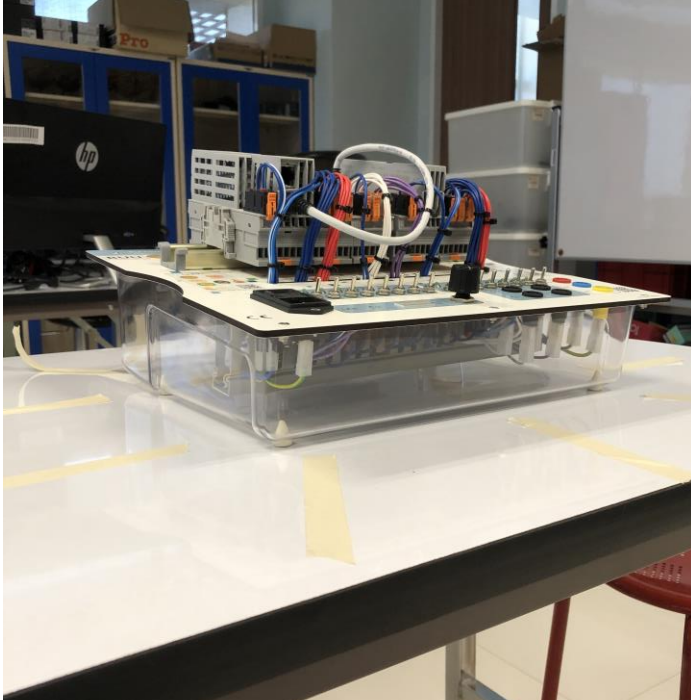
```
# บันทึกภาพที่หมุนแล้วลงในโฟลเดอร์ปลายทาง
rotated_image.save(os.path.join(destination_folder, new_filename))
```

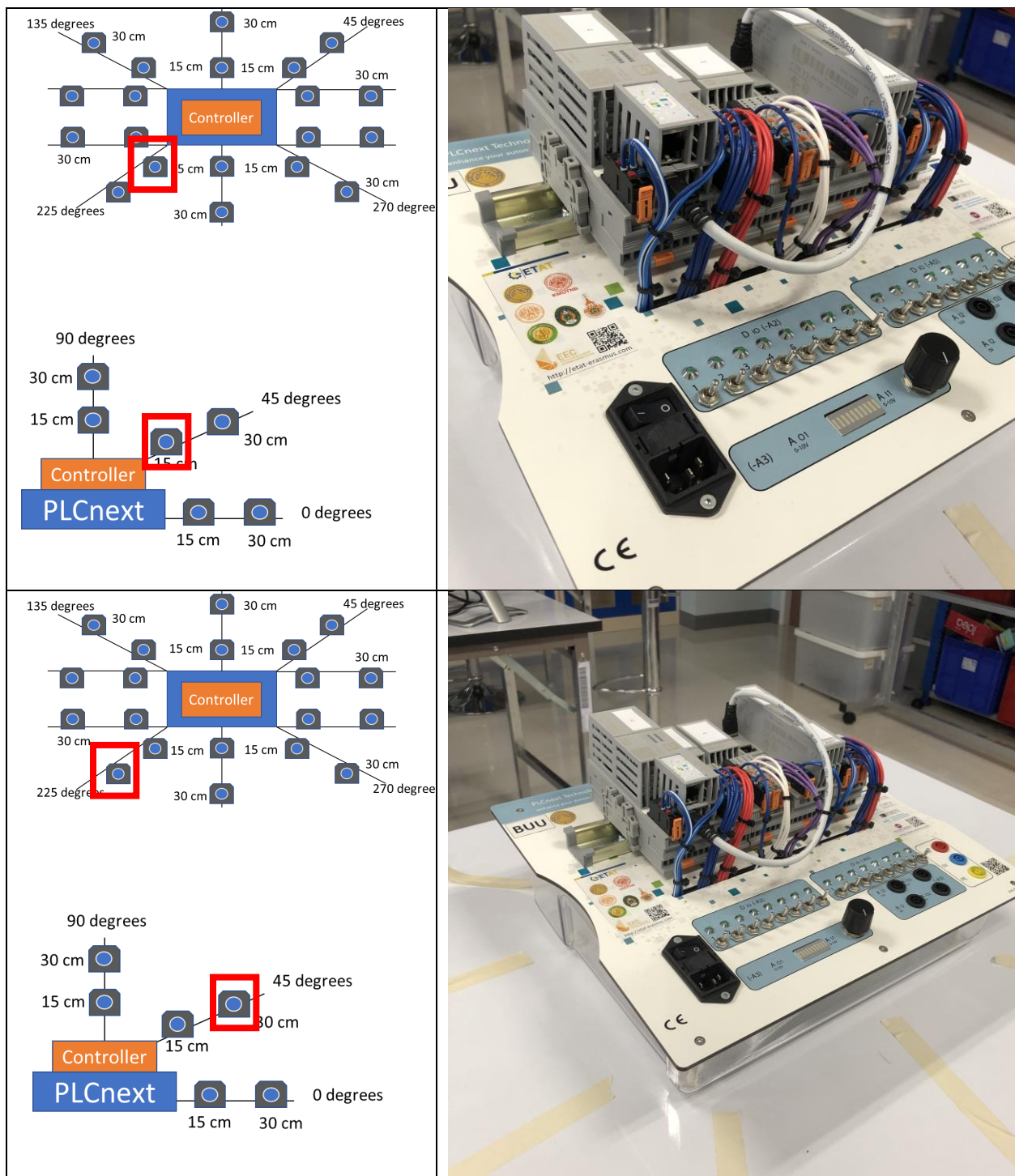
ภาพที่ 33 บันทึกภาพที่หมุนแล้ว

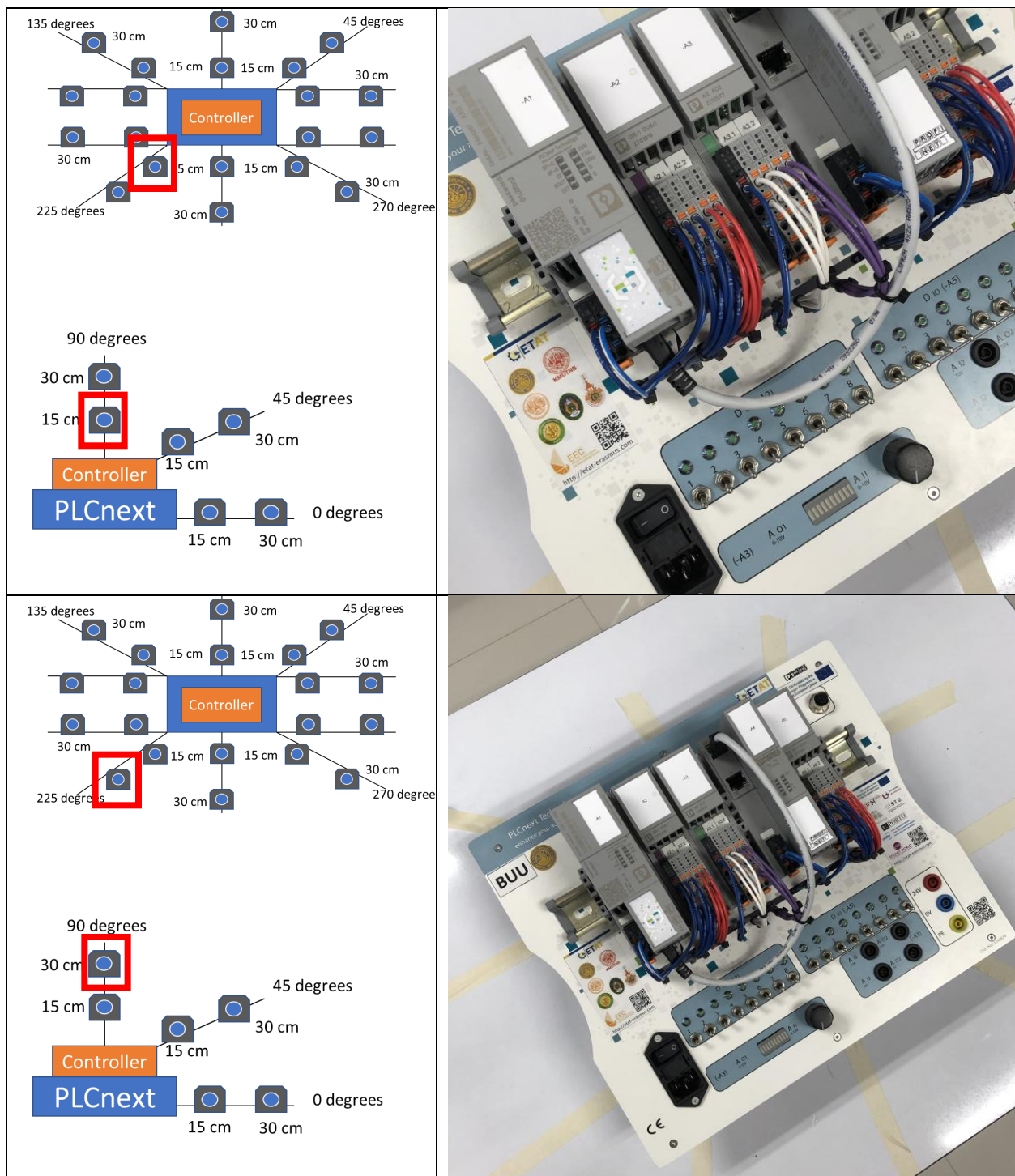
7.2 ตัวอย่างภาพจากชุดข้อมูล

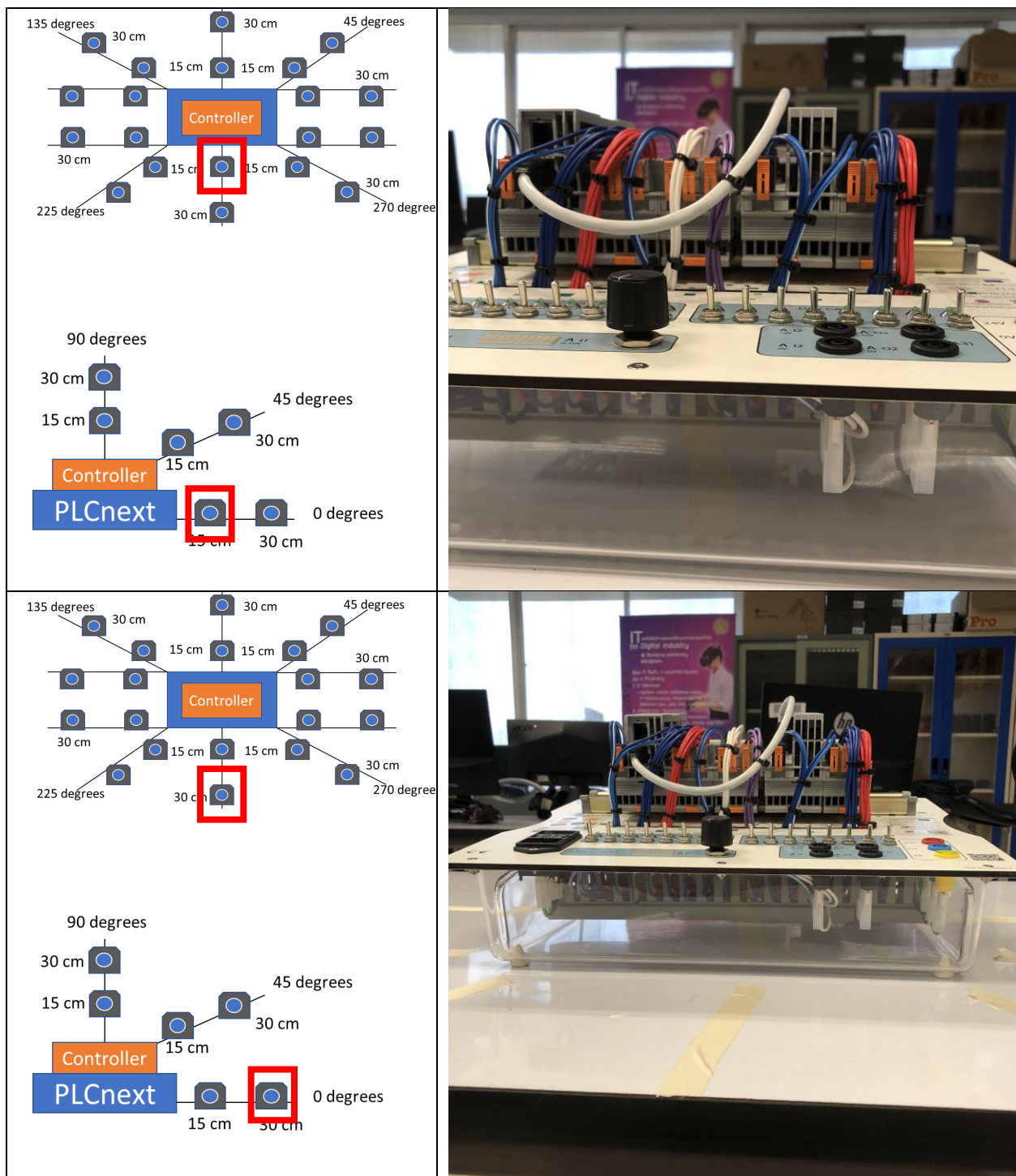
7.2.1 ชุดข้อมูลแบบทั้งเครื่อง PLCnext

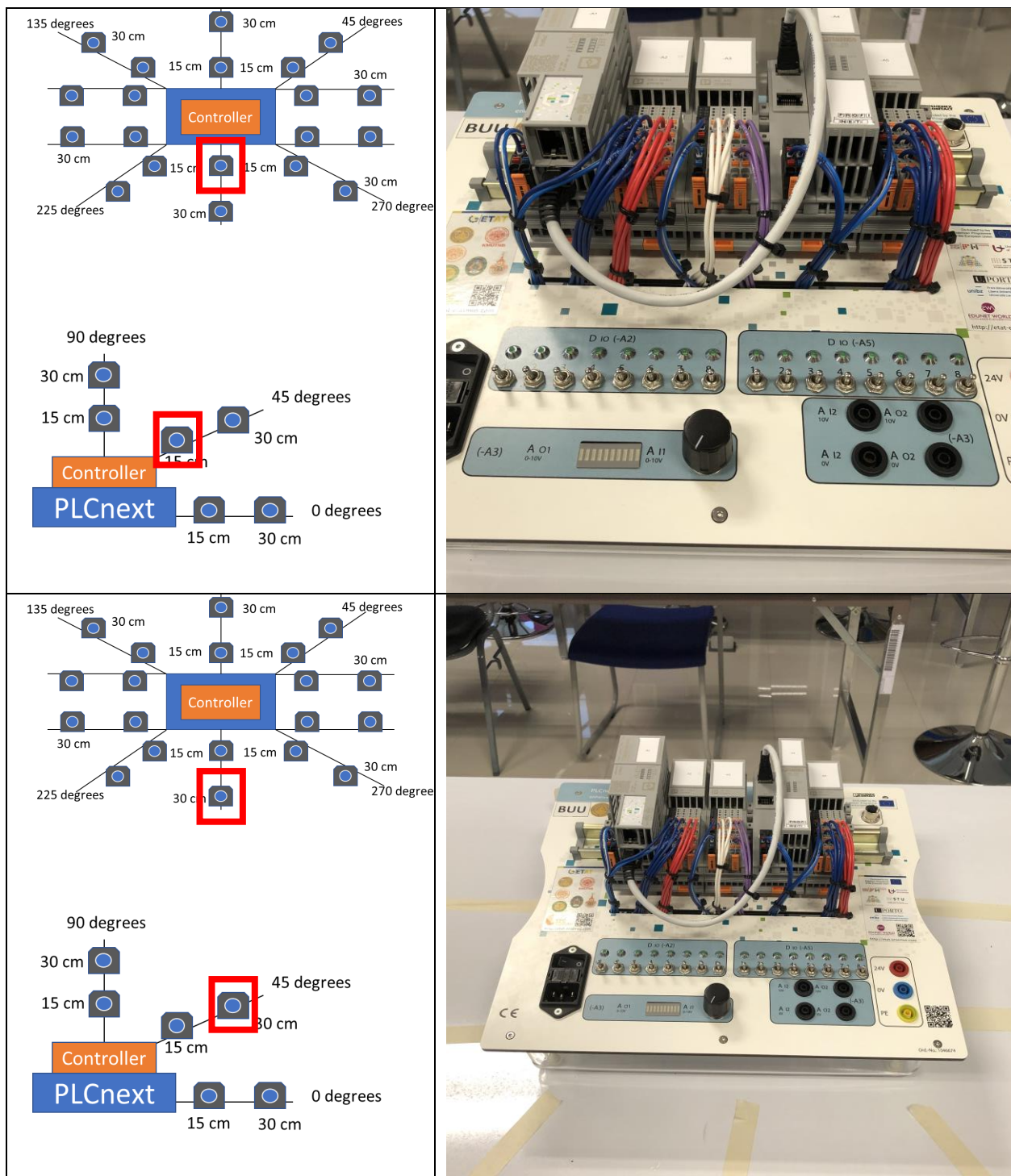
ตารางที่ 3 ชุดข้อมูลแบบทั้งเครื่อง PLCnext

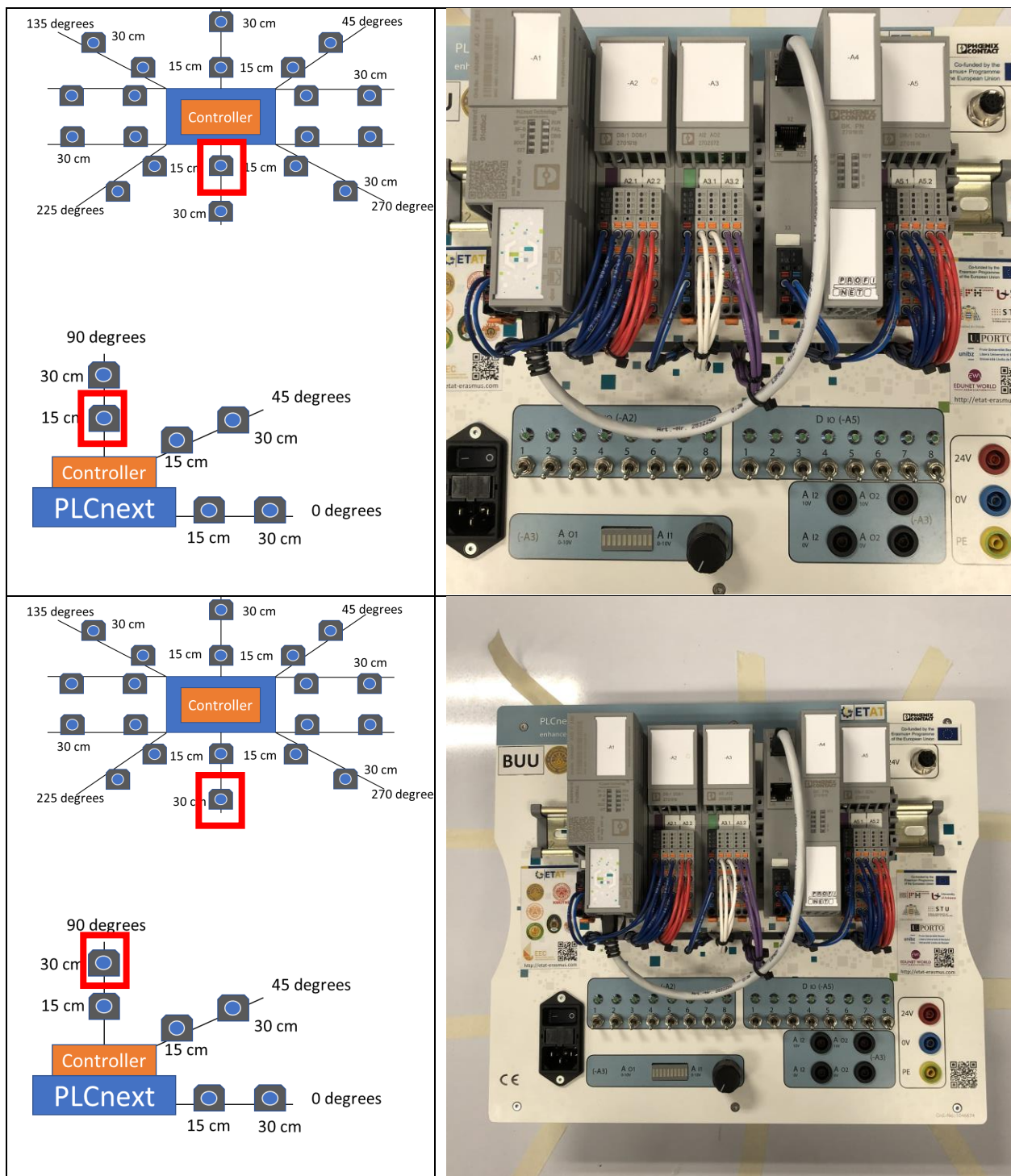
ตำแหน่ง	ภาพตัวอย่าง
	
	





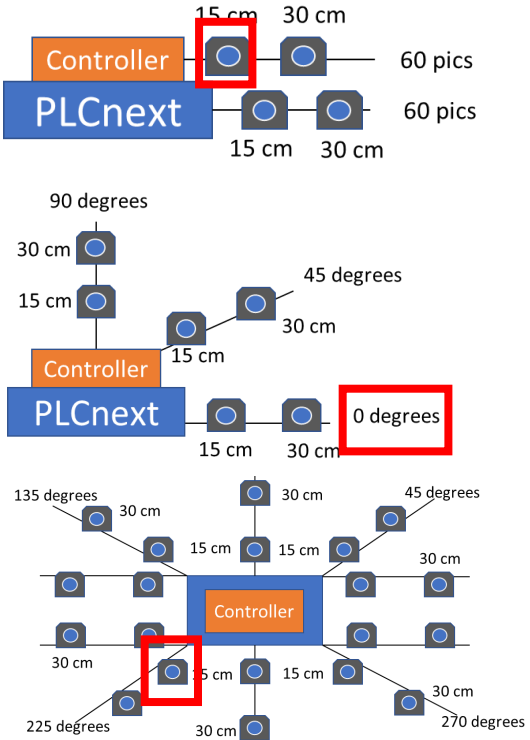
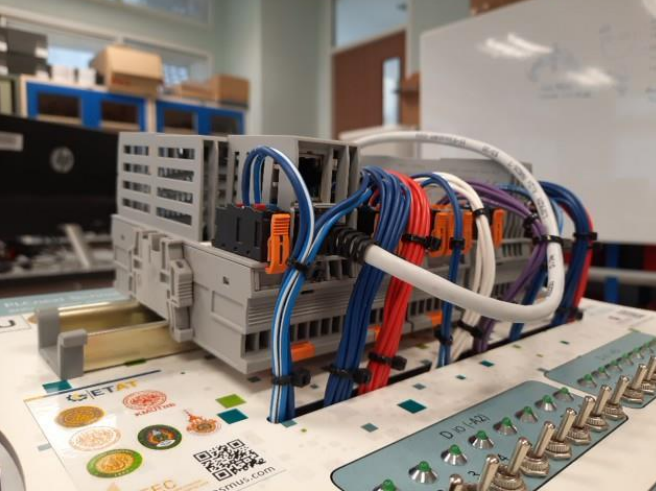
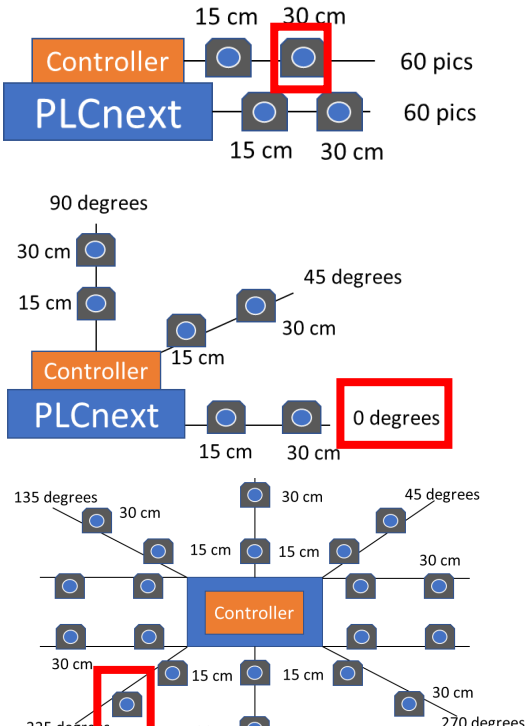
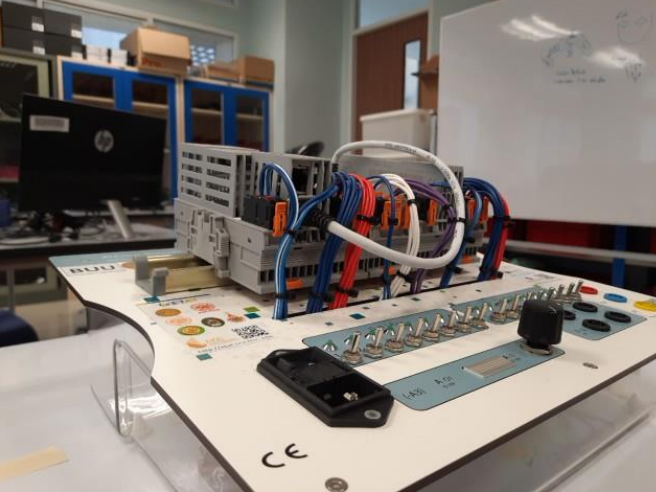


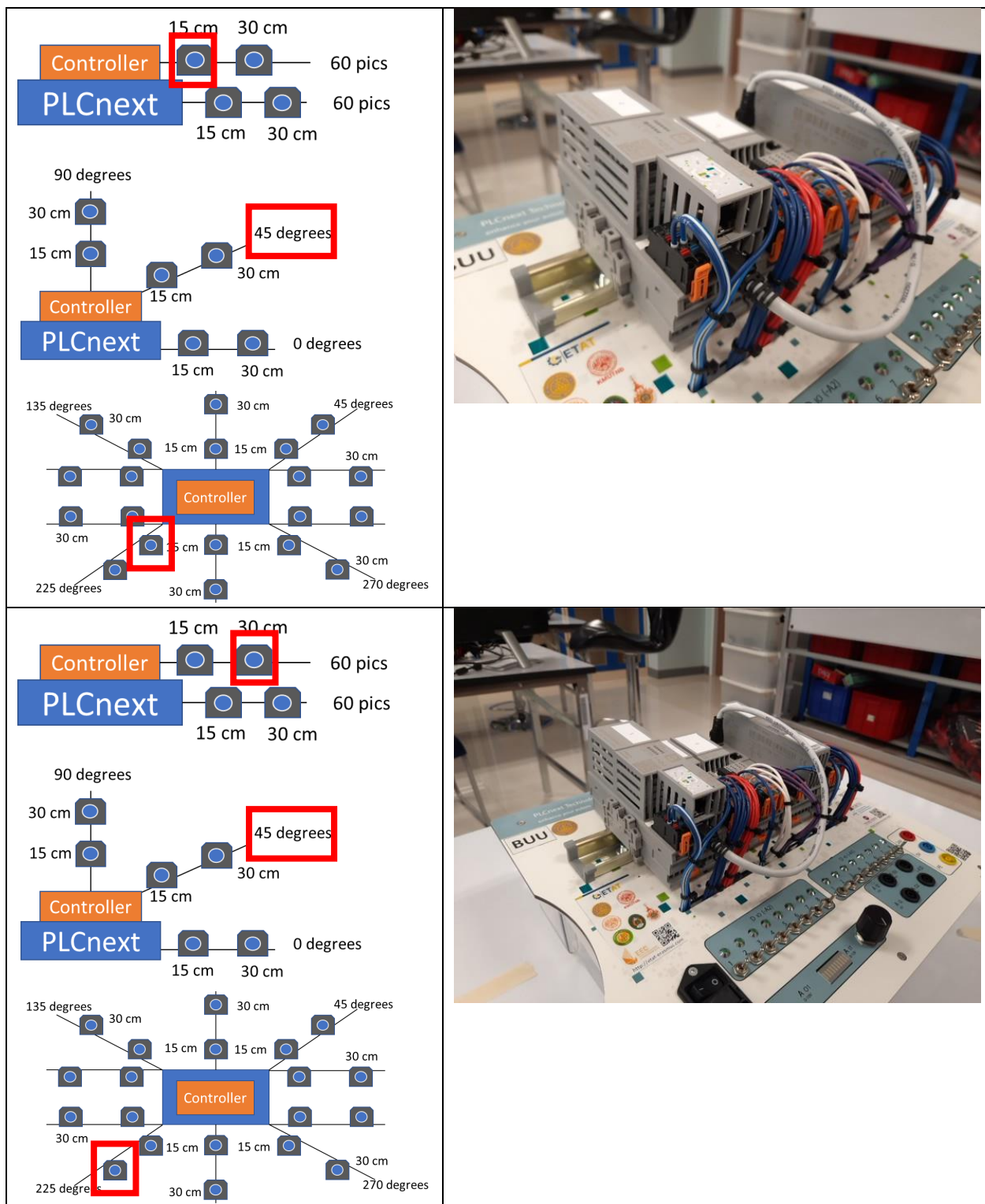


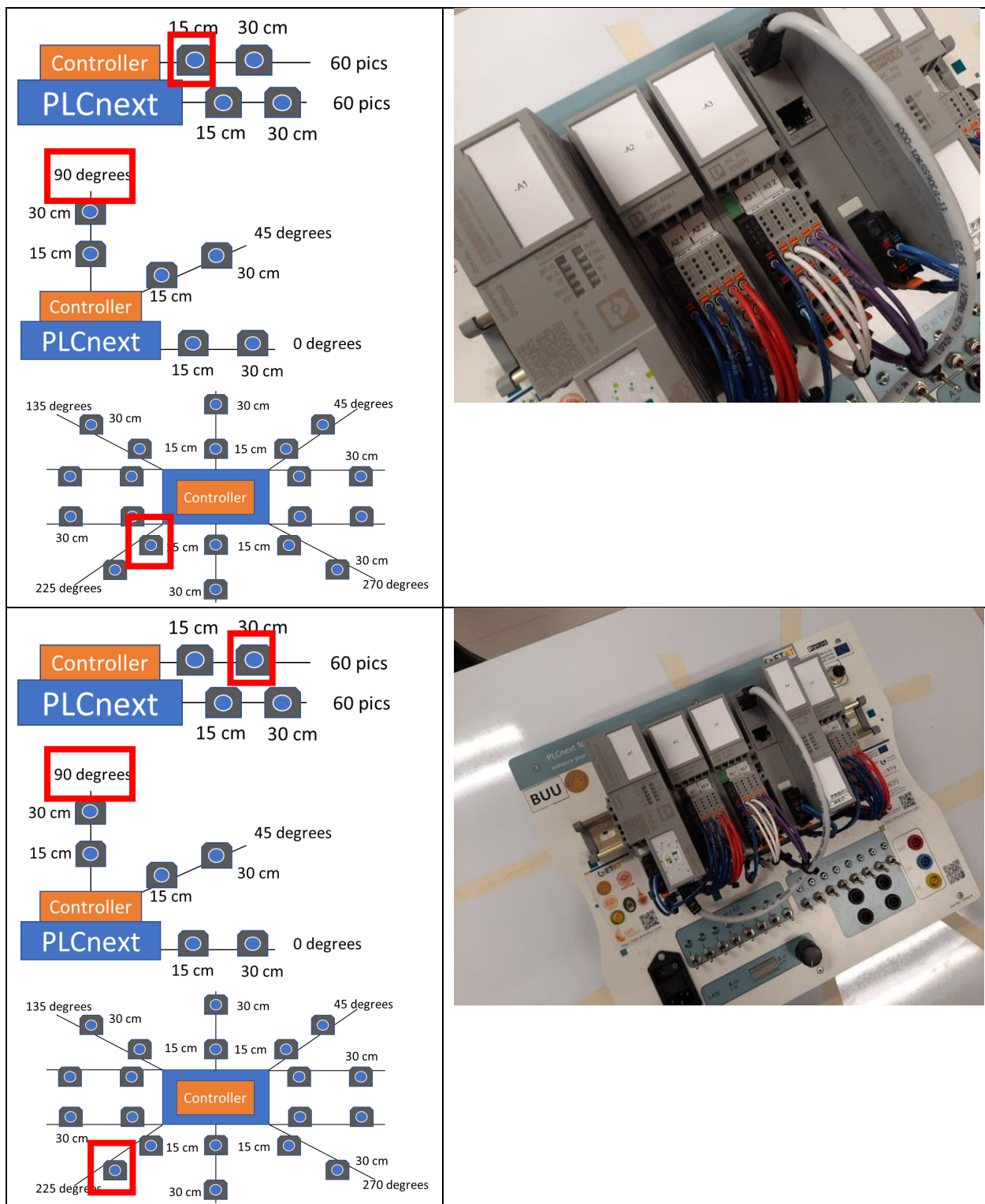


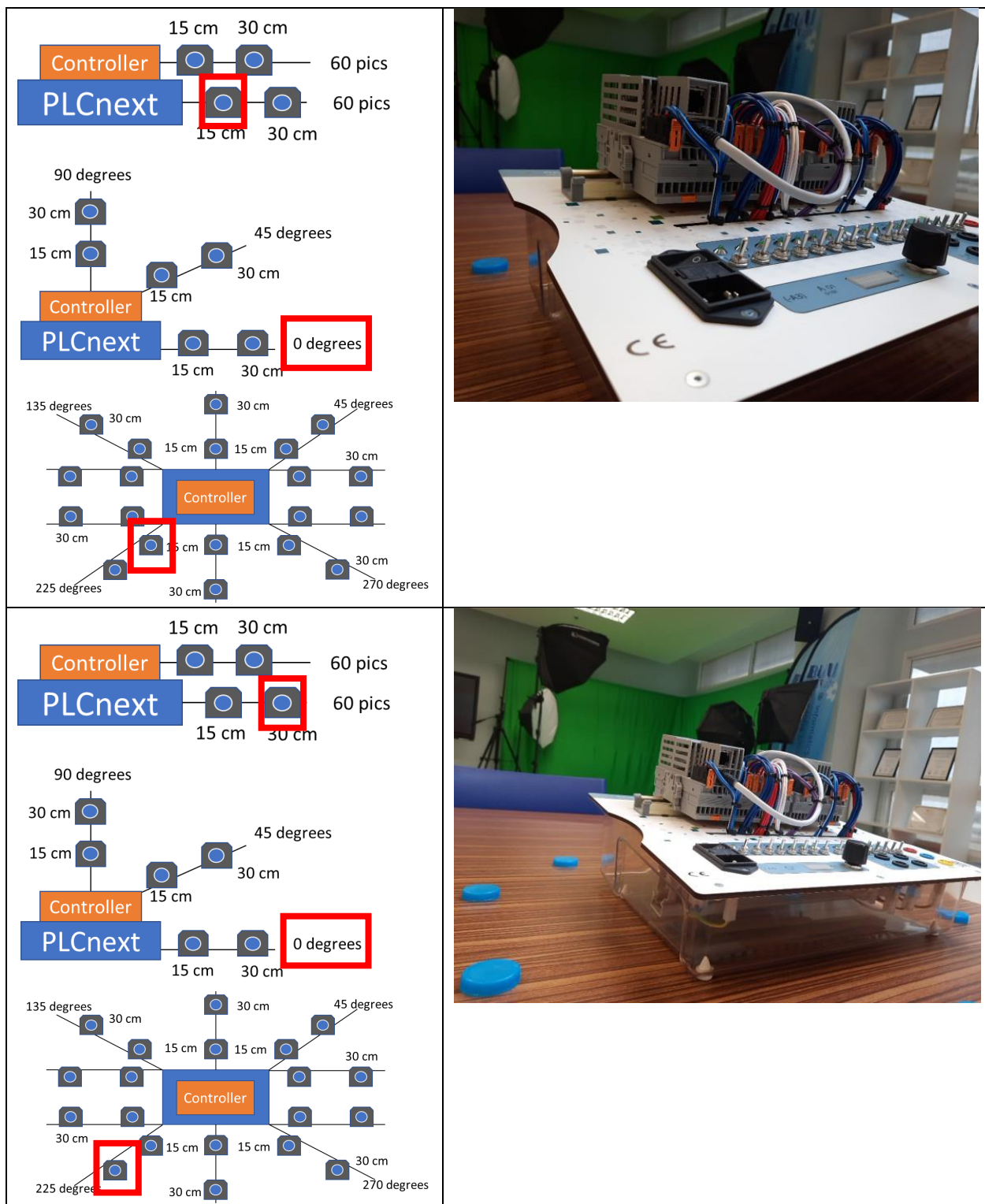
7.2.2 ชุดข้อมูลแบบเฉพาะเจาะจงจากจุดสนใจ

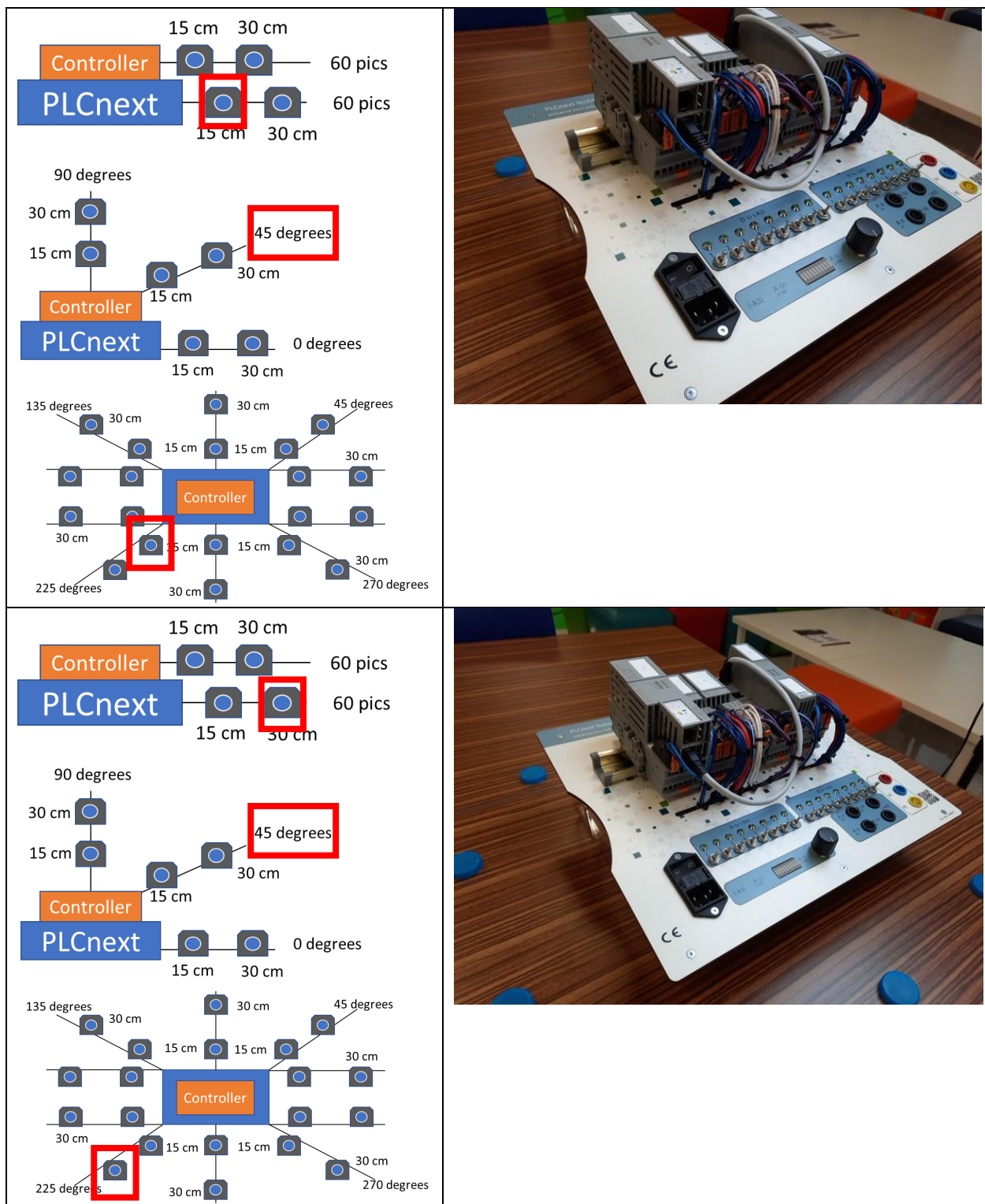
ตารางที่ 4 ชุดข้อมูลแบบเฉพาะเจาะจงจากจุดสนใจ

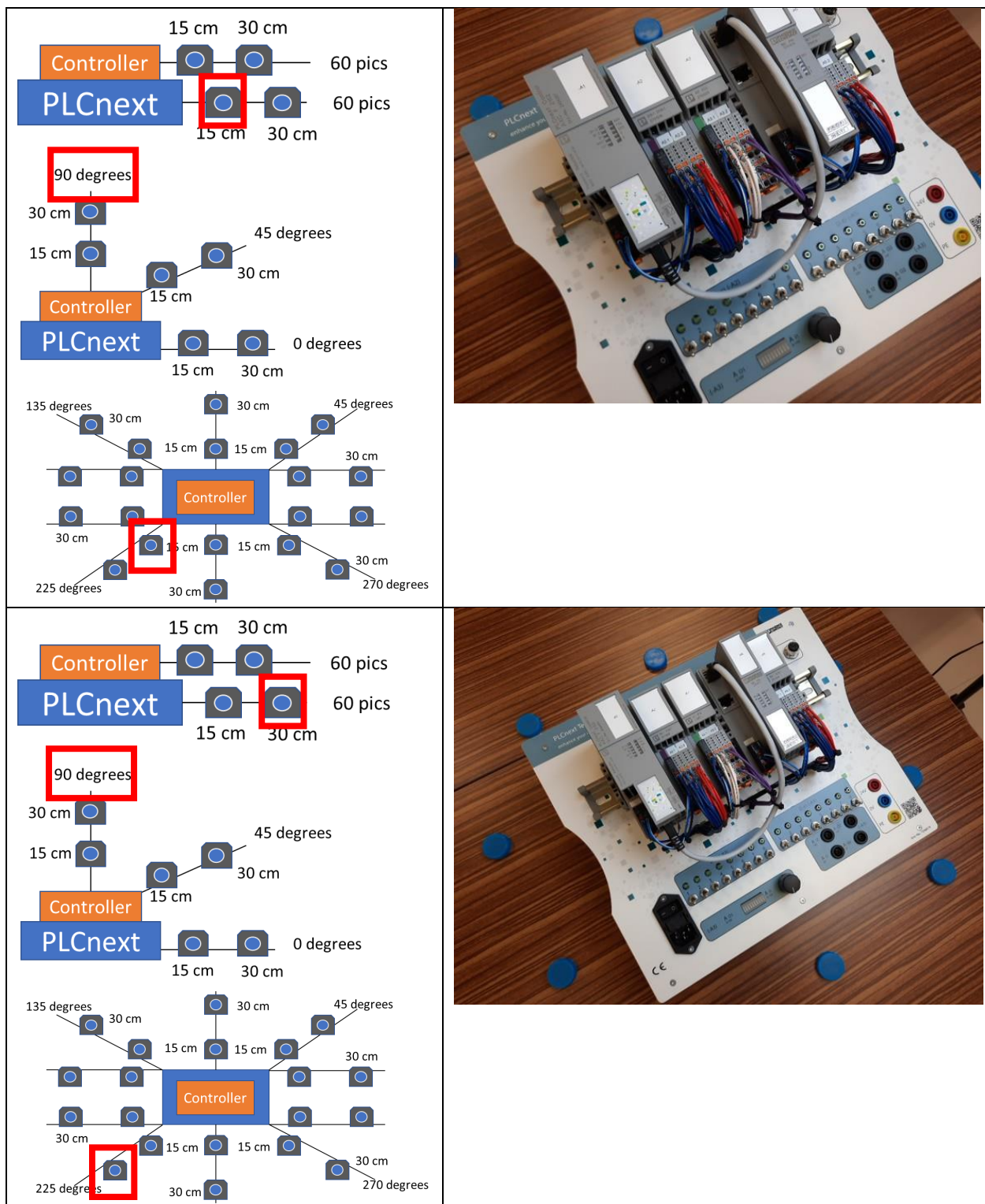
ตำแหน่ง	ภาพตัวอย่าง
	
	











8 ประวัตินักวิจัยและคณะ

8.1 หัวหน้าโครงการวิจัย

1. ชื่อ-นามสกุล ผศ.ดร.ประจักษ์ จิตเงินมะดัน

ชื่อ-นามสกุล Asst.Prof.Dr.techn. Prajaks Jitngernmadan

2. ตำแหน่งปัจจุบัน

อาจารย์ประจำคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

3. ที่อยู่ที่สามารถติดต่อได้

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ต.แสนสุข อ.เมือง จ.ชลบุรี 20131

โทรศัพท์ 038-103061 โทรสาร -

E-mail: prajaks@buu.ac.th

4. ประวัติการศึกษา

ปี	คุณวุฒิ	สถานศึกษา
2560	Doktor der technischen Wissenschaften (Doctor of Engineering Sciences, Doctor technicae)	Johannes Kepler University, Linz, Austria
2550	Master of Science in Electrical Engineering and Information Technologies	University of Applied Sciences Duesseldorf, Germany
2548	Bachelor of Science in Information Technology	University of Applied Sciences Duesseldorf, Germany

5. สาขาที่มีความชำนาญพิเศษ

Accessibility, Usability, Human Computer Interaction, User-centered Design

6. ประสบการณ์ที่เกี่ยวข้องกับการบริหารงานวิจัย

6.1 ผู้อำนวยการแผนงานวิจัย :

6.2 หัวหน้าโครงการวิจัย :

- การออกแบบและพัฒนาระบบตรวจสอบข้อเขียนแบบข้อความยาวโดยอัตโนมัติ

- การออกแบบและประยุกต์ใช้ระบบโทรศัพท์ผ่านเครือข่ายคอมพิวเตอร์โดยใช้ Asterisk เป็น

พื้นฐานสำหรับคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

6.3 โครงการวิจัยที่ทำเสร็จแล้ว :

- การออกแบบและพัฒนาระบบตรวจสอบข้อเขียนแบบข้อความยาวโดยอัตโนมัติ

- การออกแบบและประยุกต์ใช้ระบบโทรศัพท์ผ่านเครือข่ายคอมพิวเตอร์โดยใช้ Asterisk เป็นพื้นฐานสำหรับคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา