

การพัฒนาแอปพลิเคชันนำทางในที่ร่มและระบบแจ้งเตือนสิ่งกีดขวางสำหรับผู้พิการทางสายตา
ภายในสถาบันเทคโนโลยีไทย-ญี่ปุ่น

แพรวพรรณ สิริพลังคานนท์

TNI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

บัณฑิตศึกษา สาขาเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ปีการศึกษา 2568

DEVELOPMENT OF AN INDOOR NAVIGATION APPLICATION AND OBSTACLE ALERT SYSTEM
FOR THE VISUALLY IMPAIRED AT THAI-NICHI INSTITUTE OF TECHNOLOGY

Preawpan Siriphalangkanont

TNI

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate Studies

Thai-Nichi Institute of Technology

Academic Year 2025

หัวข้อวิทยานิพนธ์

การพัฒนาแอปพลิเคชันนำทางในที่ร่มและระบบแจ้งเตือนสิ่ง
กีดขวางสำหรับผู้พิการทางสายตาภายในสถาบันเทคโนโลยี
ไทย-ญี่ปุ่น

โดย

แพรวพรรณ สิริพลังคานนท์

สาขาวิชา

เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ดร.ชัชไชย วรรณบุรณ์

บัณฑิตศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....รองอธิการบดีฝ่ายวิชาการ

(รองศาสตราจารย์ ดร.วรากร ศรีเชวงทรัพย์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ดร.ณัฐกิตติ์ จิตรเอื้อตระกูล)

.....กรรมการ

(ดร.ประมุข บุญเสียง)

.....กรรมการ

(ดร.อภิษฎา นิ่มคุ้มภัย)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ดร.ชัชไชย วรรณบุรณ์)

แพรวพรรณ สิริพลังคานนท์ : การพัฒนาแอปพลิเคชันนำทางในที่ร่มและระบบแจ้งเตือนสิ่งกีดขวางสำหรับผู้พิการทางสายตาภายในสถาบันเทคโนโลยีไทย-ญี่ปุ่น. อาจารย์ที่ปรึกษา : ดร.ชัชชัย วรรณบุรณ์, 95 หน้า.

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาแอปพลิเคชันนำทางในที่ร่ม และระบบแจ้งเตือนสิ่งกีดขวางสำหรับผู้พิการทางสายตาภายในสถาบันเทคโนโลยีไทย-ญี่ปุ่น โดยมีเป้าหมายเพื่อช่วยเพิ่มความสะดวก ความปลอดภัย และความสามารถในการเคลื่อนที่ภายในอาคาร ซึ่งระบบระบุตำแหน่งแบบดั้งเดิม เช่น ระบบดาวเทียม ไม่สามารถใช้งานได้มีประสิทธิภาพในสภาพแวดล้อมภายในอาคาร แอปพลิเคชันที่พัฒนาขึ้นอาศัยการนำทางด้วยมาร์กเกอร์ความเป็นจริงเสริม (Augmented Reality Marker : AR Marker) โดยติดตั้งมาร์กเกอร์ในตำแหน่งต่างๆ ภายในอาคาร เพื่อใช้เป็นจุดอ้างอิงเชิงตำแหน่ง ร่วมกับการประมวลผล ภาพจากกล้องของสมาร์ทโฟนในการระบุตำแหน่งของผู้ใช้งาน ระบบสามารถกำหนดเส้นทางการนำทาง จากตำแหน่งเริ่มต้นไปยังจุดหมายปลายทาง และแสดงคำแนะนำการเคลื่อนที่ในรูปแบบเสียงที่เหมาะสม กับผู้พิการทางสายตา ระบบแจ้งเตือนสิ่งกีดขวาง ซึ่งถูกพัฒนาขึ้นโดยการอาศัยการตรวจจับวัตถุจากภาพ และการประเมินค่าความลึกของวัตถุที่อยู่ในเส้นทางการเคลื่อนที่ จากนั้นนำข้อมูลดังกล่าวมาประมวลผล ร่วมกับระบบลำดับความสำคัญ (Priority Framework) เพื่อกำหนดระดับความสำคัญของสิ่งกีดขวาง และแจ้งเตือนให้เหมาะสมกับสถานการณ์ ช่วยลดการแจ้งเตือนที่ไม่จำเป็น นอกจากนี้ แอปพลิเคชันยังมีความสามารถในการอ่านตัวหนังสือ และแปลงข้อความที่ตรวจพบเป็นเสียง รวมถึงรองรับการใช้งานทั้ง ภาษาไทย และภาษาอังกฤษ และสามารถปรับความเร็วของการแจ้งเตือนได้ ผลการทดสอบพบว่าแอปพลิเคชันสามารถนำทาง แจ้งเตือนสิ่งกีดขวาง และอ่านตัวหนังสือได้อย่างถูกต้องแสดงให้เห็นว่าแอปพลิเคชันที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้งานจริงเพื่อสนับสนุนการดำรงชีวิตประจำวันของผู้พิการทางสายตา ภายในอาคารได้อย่างมีประสิทธิภาพ

บัณฑิตศึกษา

สาขาวิชา เทคโนโลยีสารสนเทศ

ปีการศึกษา 2568

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

PRAWPAN SIRIPHALANGKANONT : DEVELOPMENT OF AN INDOOR NAVIGATION APPLICATION AND OBSTACLE ALERT SYSTEM FOR THE VISUALLY IMPAIRED AT THAI-NICHI INSTITUTE OF TECHNOLOGY. ADVISOR : DR.CHATCHAI WANNABOON, 95 PP.

This research proposes an indoor navigation application and an obstacle warning system for visually impaired users at the Thai-Nichi Institute of Technology, aiming to improve safety, convenience, and mobility in indoor environments where conventional positioning systems are ineffective. The application utilizes Augmented Reality Markers (AR Markers) installed within buildings as positional references, combined with image processing from a mobile device camera to determine the user's location. The system generates navigation paths and provides audio-based guidance for visually impaired users. Obstacle warnings are produced using object detection and depth estimation, with a Priority Framework applied to determine alert importance and reduce unnecessary notifications. In addition, the application supports text-to-speech functionality, bilingual operation in Thai and English, and adjustable audio notification speed. Experimental results indicate that the proposed application effectively performs indoor navigation, obstacle warnings, and text reading, demonstrating its potential for real-world deployment to support the daily activities of visually impaired users in indoor environments.

Graduate Studies

Student's Signature.....

Field of Study Information Technology

Advisor's Signature.....

Academic Year 2025

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือจากหลายฝ่าย ผู้วิจัยขอกราบขอบพระคุณจากอาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.ชัชไชย วรรณบุญ เป็นอย่างสูงที่ได้สละเวลาให้คำแนะนำ คำปรึกษา และข้อเสนอแนะอันเป็นประโยชน์อย่างยิ่ง ทั้งในด้านแนวคิดทางวิชาการ การวางแผนการดำเนินการวิจัย ตลอดจนการตรวจสอบ และปรับปรุงแก้ไขเนื้อหาของวิทยานิพนธ์ฉบับนี้ให้มีความถูกต้อง และสมบูรณ์ยิ่งขึ้น ผู้วิจัยขอขอบพระคุณในความกรุณา และความเอาใจใส่ที่มีให้แก่ผู้วิจัยตลอดระยะเวลาการศึกษา

ขอขอบพระคุณคณาจารย์ทุกท่านในสาขาวิชา ที่ได้ให้ความรู้ คำแนะนำ และถ่ายทอดประสบการณ์อันมีคุณค่า ซึ่งเป็นพื้นฐานสำคัญในการศึกษาค้นคว้า และการดำเนินงานวิจัยครั้งนี้ และช่วยให้ผู้วิจัยสามารถนำความรู้ที่ได้รับไปประยุกต์ใช้ในการพัฒนา และปรับปรุงงานวิจัยได้อย่างมีประสิทธิภาพ

ขอขอบพระคุณเพื่อนร่วมชั้นเรียน เพื่อนร่วมงาน และผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่ได้ช่วยเหลือสนับสนุน และแลกเปลี่ยนความคิดเห็นที่เป็นประโยชน์ รวมถึงให้กำลังใจแก่ผู้วิจัยตลอดระยะเวลาการดำเนินงานวิจัย ซึ่งมีส่วนช่วยให้ผู้วิจัยสามารถก้าวผ่านอุปสรรค และความท้าทายต่างๆ ได้ด้วยดี

ขอขอบพระคุณทางสถาบันเทคโนโลยีไทย-ญี่ปุ่น ที่เอื้อเฟื้อ เครื่องมือ และสถานที่ในการทดลอง และเก็บรวบรวมข้อมูล ซึ่งมีส่วนสำคัญในการทำงานวิจัยฉบับนี้สามารถดำเนินไปได้อย่างครบถ้วน และบรรลุตามวัตถุประสงค์ที่ได้กำหนดไว้

ขอขอบพระคุณบิดา มารดา ที่ให้การสนับสนุนทั้งด้านกำลังใจ และเป็นแรงผลักดันสำคัญที่ทำให้ผู้วิจัยมีกำลังใจในการศึกษา และสามารถจัดทำวิทยานิพนธ์ฉบับนี้จนลุล่วงไปได้ด้วยดี

ท้ายที่สุดนี้ ผู้วิจัยขอกราบขอบพระคุณเงินทุนสนับสนุนการวิจัยจากกองทุนวิจัย Dr.Noriaki KANO สำหรับการสนับสนุนด้านงบประมาณที่ใช้ในการดำเนินงานวิจัยครั้งนี้

แพรวพรรณ สิริพลังคานนท์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูป.....	ญ
บทที่	
1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	3
1.3 ขอบเขตงานวิจัย.....	4
2 ทฤษฎี หลักการ และงานวิจัยที่เกี่ยวข้อง.....	5
2.1 องค์ความรู้ ทฤษฎี หลักการพื้นฐานที่เกี่ยวข้อง.....	5
2.2 งานวิจัยที่เกี่ยวข้อง.....	13
3 ระเบียบและวิธีการวิจัย.....	24
3.1 ภาพรวมของกระบวนการวิจัย.....	24
3.2 การออกแบบ UX/UI.....	27
3.3 การพัฒนาแอปพลิเคชัน.....	28
3.4 การพัฒนาระบบการลำดับความสำคัญ.....	32
3.5 การพัฒนาระบบการนำทางในที่ร่มด้วยมาร์กเกอร์ความเป็นจริงเสริม.....	34
3.6 การจัดเตรียมข้อมูล (Dataset) สำหรับการสร้างโมเดลในการตรวจจับ สิ่งกีดขวาง.....	40
3.7 การพัฒนาการตรวจจับวัตถุ (Object Detection).....	40
3.8 การพัฒนาการตรวจจับอักขระด้วยแสง (Optical Character Recognition)...	41

สารบัญ (ต่อ)

บทที่		หน้า
3	3.9 การพัฒนาการสั่งการด้วยเสียง (Speech Recognition).....	41
	3.10 การทดสอบและประเมินผล.....	41
4	ผลการวิจัย.....	44
	4.1 ผลการพัฒนาแอปพลิเคชันบนสมาร์ตโฟน (Flutter).....	44
	4.2 ผลการทดสอบระบบฝั่งเซิร์ฟเวอร์ (Express.js).....	49
	4.3 ผลการทดสอบระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Flask)	56
	4.4 ผลการประเมินเชิงเปรียบเทียบ (Comparative Evaluation).....	60
5	สรุปและข้อเสนอแนะ.....	66
	5.1 ข้อสรุปของงานวิจัย	66
	5.2 ปัญหา และข้อจำกัดของงานวิจัย	67
	5.3 สำหรับการพัฒนางานวิจัยในอนาคต	67
	บรรณานุกรม	69
	ภาคผนวก	74
	ภาคผนวก ก. สถาปัตยกรรมระบบ (System Architecture).....	75
	ภาคผนวก ข. แอปพลิเคชันบนอุปกรณ์เคลื่อนที่ (Mobile Application).....	77
	ภาคผนวก ค. ระบบฝั่งเซิร์ฟเวอร์ (Backend System)	82
	ภาคผนวก ง. ระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Image Processing and Artificial Intelligence (AI) Module).....	86
	ภาคผนวก จ. ฐานข้อมูล (Database)	90
	ภาคผนวก ฉ. การเข้าถึงซอร์สโค้ด และการรันระบบ (Source Code Access and Execution).....	92
	ประวัติย่อผู้วิจัย.....	95

สารบัญตาราง

ตาราง	หน้า
2.1	14
2.2	16
2.3	19
2.4	21
2.5	
ทางสายตา.....	22
4.1	64



สารบัญรูป

รูป	หน้า
1.1 สถานการณ์คนพิการในประเทศไทย	3
2.1 ภาพตัวอย่างการตรวจจับวัตถุ	6
2.2 ขั้นตอนการตรวจจับอักขระด้วยแสง	7
2.3 ตัวอย่างการกำหนดเส้นทางการนำทางแบบโหนด และมาร์กเกอร์ความเป็นจริงเสริม	11
3.1 ภาพรวมของกระบวนการทำงานของแอปพลิเคชัน	25
3.2 สถาปัตยกรรมการทำงานของระบบ Seeable	26
3.3 แผนภาพภาพรวมการทำงานและสถาปัตยกรรมของระบบ Seeable	27
3.4 ผังอาคาร C ชั้น 3 ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น	36
4.1 ตัวอย่างหน้าจอการเลือกสถานที่ในการนำทาง	45
4.2 ตัวอย่างหน้าจอขณะนำทาง และแจ้งเตือนสิ่งกีดขวาง	46
4.3 ตัวอย่างหน้าจอการตรวจจับวัตถุ	47
4.4 ตัวอย่างหน้าจอการสแกนข้อความ	48
4.5 ตัวอย่างหน้าจอการตั้งค่า และการปรับแต่งแอปพลิเคชัน	49
4.6 ตัวอย่างการส่งรูปภาพเพื่อการตรวจจับสิ่งกีดขวางผ่านโปรแกรม Postman	50
4.7 ตัวอย่างการส่งรูปภาพเพื่อการตรวจจับวัตถุผ่านโปรแกรม Postman	51
4.8 ตัวอย่างการส่งรูปภาพเพื่อการตรวจจับอักขระด้วยแสงผ่านโปรแกรม Postman	52
4.9 ตัวอย่างผลลัพธ์การกำหนด session_id และเส้นทางผ่านโปรแกรม Postman	53
4.10 ตัวอย่างผลลัพธ์การอัปเดตตำแหน่งของผู้ใช้งานผ่านโปรแกรม Postman	54
4.11 ตัวอย่างผลลัพธ์การอัปเดตตำแหน่งของผู้ใช้งาน	54
4.12 ผลการฝึกโมเดลการตรวจจับสิ่งกีดขวางใน Epoch ที่ดีที่สุด	56
4.13 ประสิทธิภาพของโมเดลการตรวจจับสิ่งกีดขวางรายการคลาส	57
4.14 Confusion Matrix ของโมเดลการตรวจจับสิ่งกีดขวาง	58
4.15 ตัวอย่างผลการผสมผสานการตรวจจับวัตถุ การประเมินค่าความลึก และค่าลำดับ ความสำคัญพร้อมแสดง Bounding Box และ Priority Score	59
4.16 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินและระบบในกรณีที่มี วัตถุหลายรายการในแนวทางเดิน	61

สารบัญรูป (ต่อ)

รูป	หน้า
4.17 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินและระบบในสภาพ แวดล้อมทางเดินที่มีระยะลึกและวัตถุซ้อนกัน.....	62
4.18 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินและระบบในกรณีที่มีวัตถุใกล้ และวัตถุไกลในภาพเดียวกัน.....	63
ก.1 ระบบ System Architecture.....	76



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ประเทศไทยมีจำนวนผู้พิการทางสายตาเพิ่มขึ้นอย่างต่อเนื่องในแต่ละปี โดยข้อมูลจากสำนักพัฒนาสังคมและความมั่นคงของมนุษย์ระบุว่าผู้พิการประมาณ 2 ล้านคน ในจำนวนนี้เป็นผู้พิการทางสายตาประมาณ 200,000 คน คิดเป็น 8.06% [1] ของผู้พิการทั้งหมด โดยสามารถดูสถิติได้จากรูปที่ 1.1 ผู้พิการทางสายตาต้องเผชิญกับข้อจำกัดหลายประการในการดำรงชีวิตประจำวัน ไม่ว่าจะเป็นการเดินทาง การเข้าถึงข้อมูล และการมีปฏิสัมพันธ์กับสิ่งแวดล้อมรอบตัว ซึ่งส่งผลกระทบต่อคุณภาพชีวิตและความสามารถในการดำเนินชีวิตอย่างอิสระ ด้วยเหตุนี้ การพัฒนาเทคโนโลยีที่สามารถช่วยเหลือผู้พิการทางสายตาจึงเป็นสิ่งที่มีความสำคัญอย่างยิ่ง

ผู้พิการทางสายตาเป็นกลุ่มบุคคลที่มีข้อจำกัดในการรับรู้สภาพแวดล้อมรอบตัวผ่านการมองเห็น ซึ่งส่งผลกระทบโดยเฉพาะในพื้นที่ภายในอาคาร เนื่องจากการออกแบบสถาปัตยกรรมและโครงสร้างพื้นฐานส่วนใหญ่ไม่ได้คำนึงถึงการรองรับการใช้งานของผู้พิการทางสายตาอย่างเหมาะสม ปัญหาหลักที่พบในอาคาร ได้แก่ การเดินทางภายในพื้นที่ การเข้าถึงข้อมูล และการทำกิจกรรมต่างๆ ซึ่งล้วนเป็นอุปสรรคต่อการดำรงชีวิตของผู้พิการทางสายตา

ในการเดินทางภายในอาคาร ผู้พิการทางสายตามักประสบปัญหาในการค้นหาเส้นทางหรือจุดหมายที่ต้องการ เช่น ห้องน้ำ ทางออกฉุกเฉิน เนื่องจากขาดป้ายบอกทางหรือเครื่องหมายสัญลักษณ์ที่สามารถแปลงเป็นเสียงหรือข้อความดิจิทัลเพื่อให้ผู้พิการสามารถรับรู้ได้ อีกทั้งอุปสรรคทางกายภาพ เช่น มีสิ่งกีดขวางหรือการจัดพื้นที่ที่ไม่เป็นระเบียบ อาจเพิ่มความเสี่ยงต่อการเดินชนหรือสะดุดล้ม นอกจากนี้ การเข้าถึงข้อมูลภายในอาคาร เช่น ป้ายประกาศ หมายเลขห้อง หรือข้อมูลต่างๆ ยังคงอยู่ในรูปแบบตัวอักษรที่ผู้พิการทางสายตาไม่สามารถอ่านได้ ส่งผลให้ต้องพึ่งพาการสอบถามจากเจ้าหน้าที่ ซึ่งอาจไม่สะดวกในบางกรณี

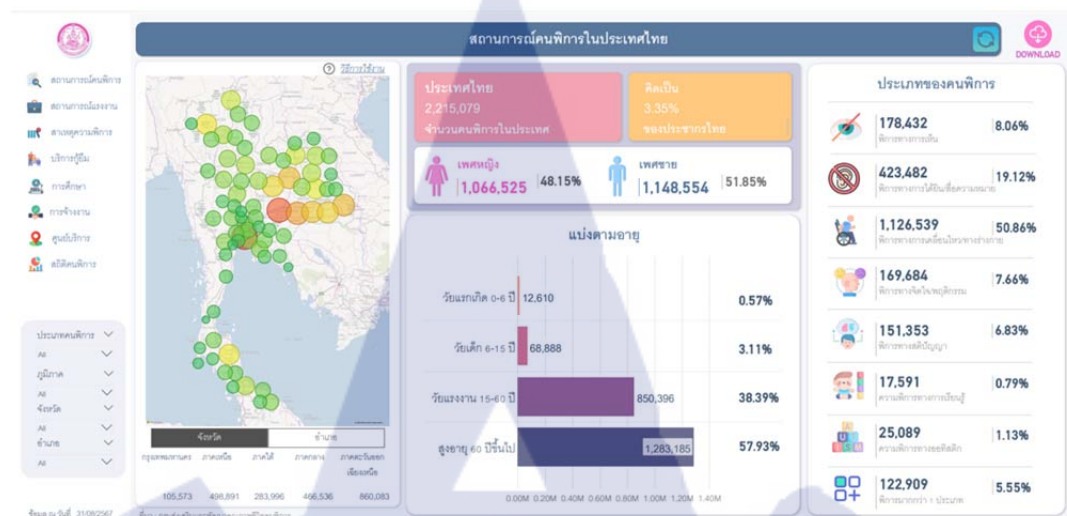
การทำกิจกรรมภายในอาคารยังเป็นอีกหนึ่งความท้าทายสำหรับผู้พิการทางสายตา การเลือกซื้อสินค้าในร้านค้าภายในอาคาร การใช้บริการเครื่องจำหน่ายสินค้าอัตโนมัติ หรือการค้นหาสิ่งของที่ต้องการยังคงเป็นเรื่องยาก เนื่องจากขาดเทคโนโลยีที่สามารถช่วยระบุและบอกตำแหน่งของสิ่งของได้อย่างแม่นยำ แม้ว่าปัจจุบันจะมีแอปพลิเคชันที่ช่วยเหลือผู้พิการทางสายตาในด้านต่างๆ เช่น การแปลงข้อความเป็นเสียง (Text-to-Speech) หรือการระบุวัตถุ (Object Detection) แต่แอปพลิเคชันเหล่านี้ยังไม่สามารถรองรับการใช้งานเฉพาะในอาคารได้อย่างมีประสิทธิภาพ เช่น การนำทางในอาคารที่มีความซับซ้อน การแยกแยะตำแหน่งสิ่งของหรือห้องต่างๆ รวมถึงการสั่งงานด้วยเสียงที่ยังไม่รองรับภาษาไทย หรือขาดความแม่นยำในการตรวจจวัตถุภายในอาคารที่มีลักษณะเฉพาะ

ในปัจจุบันมีการพัฒนาและประยุกต์ใช้เทคโนโลยีเหล่านี้เพื่อช่วยเหลือผู้พิการทางสายตา เพื่อที่จะสามารถใช้ชีวิตได้อย่างมีประสิทธิภาพมากขึ้น เช่น เทคโนโลยีในการตรวจจับวัตถุ (Object Detection) ซึ่งใช้ปัญญาประดิษฐ์ และการเรียนรู้เชิงลึกในการระบุ และกำหนดตำแหน่งของสิ่งของรอบตัว เทคโนโลยีการรู้จำอักขระด้วยแสง (Optical Character Recognition) ที่สามารถแปลงข้อมูลจากเอกสาร หรือป้ายประกาศให้อยู่ในรูปแบบดิจิทัล เพื่อสามารถส่งต่อข้อมูลให้ระบบอ่านข้อความเป็นเสียง เทคโนโลยีการประมาณระยะลึก (Depth Estimation) ที่ช่วยวิเคราะห์ความใกล้ไกลของสิ่งกีดขวาง เพื่อเพิ่มความปลอดภัยในการเดินทาง และเทคโนโลยีการสังเคราะห์เสียง (Text-to-Speech : TTS) และการรู้จำเสียงพูด (Speech-to-Text : STT) ที่ทำให้ผู้ใช้สามารถสื่อสาร และโต้ตอบกับอุปกรณ์ได้อย่างสะดวก และเป็นธรรมชาติ นอกจากนี้ ยังมีเทคโนโลยีการนำทางในที่ร่มที่อาศัยการตรวจจับมาร์กเกอร์ ความเป็นจริงเสริม (AR Marker) เพื่อช่วยผู้พิการทางสายตาในการระบุตำแหน่ง และนำทางภายในอาคาร ที่มีโครงสร้างที่ซับซ้อน การบูรณาการเทคโนโลยีเหล่านี้เข้าด้วยกันจึงเป็นแนวทางสำคัญที่จะช่วยพัฒนาเครื่องมือ หรือแอปพลิเคชันที่ตอบโจทย์การใช้งานจริง และยกระดับคุณภาพชีวิตของผู้พิการทางสายตาได้อย่างมีประสิทธิภาพ

อย่างไรก็ตาม แม้เทคโนโลยีเหล่านี้มีประสิทธิภาพสูง แต่ยังขาดการรวบรวมให้เป็นระบบที่ใช้งานได้จริงภายในอาคาร และเหมาะสมกับบริบทของประเทศไทย ดังนั้น เพื่อแก้ไขปัญหาดังกล่าว แอปพลิเคชัน Seeable จึงถูกพัฒนาขึ้น โดยเป็นแอปพลิเคชันที่ผสานเทคโนโลยีหลายรูปแบบ ได้แก่ การตรวจจับวัตถุ (Object Detection) การรู้จำอักขระด้วยแสง (Optical Character Recognition) การประมาณระยะลึก (Depth Estimation) การสั่งงานด้วยเสียง (Voice Commands) และการโต้ตอบด้วยเสียง (Voice Interaction) เทคโนโลยีเหล่านี้ช่วยเพิ่มความสามารถในการดำรงชีวิตของผู้พิการทางสายตาในประเทศไทยให้เป็นไปอย่างอิสระมากยิ่งขึ้น

แอปพลิเคชันนี้จะช่วยให้ผู้พิการทางสายตาสามารถรับรู้สิ่งแวดล้อมรอบตัวได้อย่างรวดเร็ว และแม่นยำ โดยใช้เทคโนโลยีการตรวจจับวัตถุที่สามารถบอกชื่อวัตถุรอบข้างในทันทีที่พบ อีกทั้งยังรองรับการควบคุมผ่านคำสั่งเสียง ซึ่งช่วยให้ผู้ใช้สามารถสั่งงานและควบคุมแอปพลิเคชันได้โดยไม่ต้องใช้มือสัมผัสหน้าจอ ลดความลำบากในการใช้งานในชีวิตประจำวัน

นอกจากนี้ การพัฒนาแอปพลิเคชัน Seeable ยังช่วยเพิ่มความสะดวกและความมั่นใจในการเดินทางของผู้ใช้ ด้วยการช่วยระบุสิ่งกีดขวางและให้ข้อมูลเกี่ยวกับเส้นทางรอบตัว ส่งผลให้การเดินทางภายในอาคารเป็นไปอย่างปลอดภัยและสะดวกมากขึ้น งานวิจัยและพัฒนาแอปพลิเคชันนี้มุ่งเน้นไปที่การนำเทคโนโลยีใหม่ๆ เช่น การตรวจจับวัตถุและการสั่งงานด้วยเสียง มาใช้เพื่อสนับสนุนความสามารถในการดำรงชีวิตของผู้พิการทางสายตา



รูปที่ 1.1 สํานักการคํานวณประชากรในประเทศไทย [1]

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อออกแบบและพัฒนาแอปพลิเคชันสำหรับการนำทางภายในอาคารที่เหมาะสมกับผู้พิการทางสายตา โดยออกแบบส่วนติดต่อผู้ใช้ (User Interface) ให้เหมาะสมกับลักษณะการใช้งานของผู้พิการทางสายตา รองรับการสั่งการด้วยเสียง และสนับสนุนการทำงานร่วมกับโหมดช่วยเหลือการเข้าถึง (Accessibility Mode) ของระบบปฏิบัติการ เพื่อให้สามารถใช้งานได้สะดวก และมีประสิทธิภาพ

1.2.2 เพื่อพัฒนาระบบการนำทางในที่ร่ม (Indoor Navigation System) โดยใช้มาร์กเกอร์ความเป็นจริงเสริม (Augmented Reality Marker: AR Marker) เป็นจุดอ้างอิงเชิงตำแหน่ง ร่วมกับการวางแผนการนำทางในรูปแบบโหนด (Node-Based Navigation) เพื่อเป็นการกำหนดลำดับเส้นทางการเดินจากจุดเริ่มต้นไปยังเป้าหมายปลายทางที่กำหนดไว้

1.2.3 เพื่อพัฒนาระบบตรวจจับวัตถุ (Object Detection) สำหรับระบุประเภท และตำแหน่งของวัตถุภายในอาคาร เพื่อสนับสนุนการรับรู้สภาพแวดล้อมของผู้ใช้งาน

1.2.4 เพื่อพัฒนาระบบแจ้งเตือนสิ่งกีดขวาง (Obstacle Warning System) โดยผสมผสานการตรวจจับวัตถุร่วมกับการประเมินค่าความลึก (Depth Estimation) และการลำดับความสำคัญด้วย Priority Framework เพื่อคัดเลือกว่าวัตถุที่มีความเสี่ยงสูงสุด และแจ้งเตือนผู้ใช้งานอย่างเหมาะสม

1.2.5 เพื่อพัฒนาฟังก์ชันการตรวจจับอักขระด้วยแสง (Optical Character Recognition: OCR) สำหรับอ่านข้อมูลตัวอักษร และแปลงเป็นเสียงเพื่อสนับสนุนการรับรู้ข้อมูลของผู้ใช้งาน

1.3 ขอบเขตงานวิจัย

1.3.1 ออกแบบและพัฒนาแอปพลิเคชันที่สามารถใช้งานได้ในระบบปฏิบัติการ iOS และ Android โดยใช้ Flutter Framework และเชื่อมต่อกับระบบประมวลผลฝั่งเซิร์ฟเวอร์

1.3.2 ระบบนำทางใช้แนวคิดแบบโหนด (Node-based Navigation) โดยอาศัยมาร์กเกอร์ความเป็นจริงเสริมจำนวน 15 จุด ครอบคลุมห้อง C301-C307 และ C314 ซึ่งมาร์กเกอร์ติดตั้งบริเวณประตูหน้า และประตูหลังของห้องภายในอาคาร C ชั้น 3 สถาบันเทคโนโลยีไทย-ญี่ปุ่น เพื่อใช้เป็นจุดในการอ้างอิงตำแหน่ง

1.3.3 ระบบตรวจจับสิ่งกีดขวางพัฒนาด้วยโมเดล YOLOv8n โดยกำหนดการตรวจจับวัตถุทั้งหมด 14 ประเภท ได้แก่ ตู้, เก้าอี้, ของที่วางอยู่บริเวณทางเดิน, ประตู, ลิฟต์, ป้ายทางออกฉุกเฉิน, ถังดับเพลิง, คน, บันได, โต๊ะ, ป้ายสัญลักษณ์ห้องน้ำ, ถังขยะ, ไวท์บอร์ด และหน้าต่าง ซึ่งเป็นวัตถุที่พบได้ในอาคาร C ชั้น 3

1.3.4 ระบบตรวจจับอักขระด้วยแสง (OCR) รองรับการอ่านข้อความภาษาไทย และภาษาอังกฤษจากป้ายหรือเอกสารภายในอาคาร

1.3.5 การทดสอบ และประเมินผลดำเนินการภายในพื้นที่อาคาร C ชั้น 3 สถาบันเทคโนโลยีไทย-ญี่ปุ่น เท่านั้น

บทที่ 2

ทฤษฎี หลักการ และงานวิจัยที่เกี่ยวข้อง

2.1 องค์ความรู้ ทฤษฎี หลักการพื้นฐานที่เกี่ยวข้อง

2.1.1 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ เป็นกระบวนการที่ใช้ในการปรับปรุง วิเคราะห์ หรือแปลงภาพดิจิทัลให้อยู่ในรูปแบบที่เหมาะสมต่อการนำไปใช้งาน โดยอาศัยหลักการทางคณิตศาสตร์ เช่น การแปลงเชิงเส้น (Linear Transformation), การกรองภาพ (Filtering) และการปรับค่าความเข้มแสง (Intensity Adjustment)

ในบริบทของงานวิจัยนี้ การประมวลผลภาพมีบทบาทสำคัญในขั้นตอนการนำภาพเข้าสู่โมเดลตรวจจับวัตถุ และการประเมินค่าความลึก โดยประกอบด้วยขั้นตอนสำคัญ ได้แก่

1. การปรับขนาดภาพ (Image Resizing) ให้เป็น 640x640 พิกเซล เพื่อให้สอดคล้องกับขนาดของโมเดลที่ใช้ในการตรวจจับวัตถุ
2. การตรวจสอบ และปรับค่าความสว่างด้วย Gamma Correction เพื่อปรับสมดุลของแสงในกรณีที่ภาพมืดหรือสว่างเกินไป
3. การเพิ่มความคมชัดของภาพในบริเวณที่แสงไม่สม่ำเสมอด้วย Contrast Limited Adaptive Histogram Equalization (CLAHE)
4. การลดสิ่งรบกวน (Noise) ด้วย Gaussian Blur เป็นการทำให้ภาพมีความเรียบเนียนมากยิ่งขึ้น
5. การเพิ่มความชัดให้กับขอบของวัตถุโดยใช้ Sharpening Filter เพื่อช่วยให้ระบบสามารถแยกขอบเขตของวัตถุได้ดียิ่งขึ้น

กระบวนการเหล่านี้ช่วยลดสัญญาณรบกวน เพิ่มความชัดเจนของวัตถุ และเพิ่มเสถียรภาพของระบบ โดยเฉพาะในสภาพแสงภายในอาคารที่มีความผันผวน ซึ่งมีผลโดยตรงต่อประสิทธิภาพของระบบตรวจจับวัตถุและการแจ้งเตือนสิ่งกีดขวางในแอปพลิเคชัน Seeable

2.1.2 การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุ เป็นเทคโนโลยีในสาขาปัญญาประดิษฐ์ (Artificial Intelligence: AI) และคอมพิวเตอร์วิทัศน์ (Computer Vision) ที่สามารถระบุประเภท (Classification) และกำหนดตำแหน่ง (Localization) ของวัตถุภายในภาพหรือวิดีโอได้พร้อมกัน โดยผลลัพธ์จะอยู่ในรูปแบบของกรอบสี่เหลี่ยม (Bounding Box) พร้อมค่าความเชื่อมั่น (Confidence Score) ดังแสดงในรูปที่ 2.1

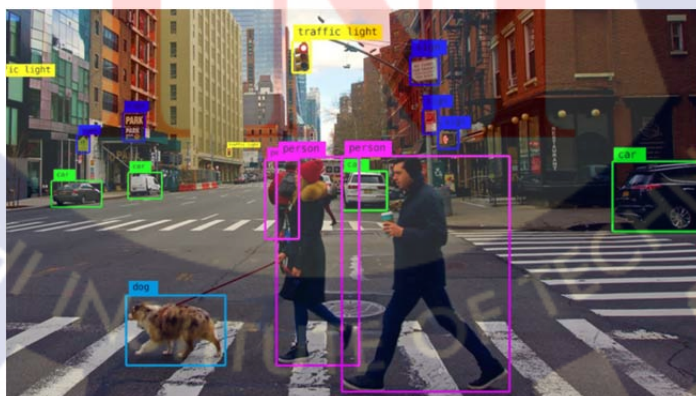
เทคโนโลยีการตรวจจับวัตถุถูกนำไปใช้ในหลากหลายด้าน เช่น ระบบยานยนต์ไร้คนขับ (Autonomous Vehicle) ใช้ในการตรวจจับพาหนะ คนเดินถนน และสิ่งกีดขวาง [2] การวิเคราะห์ภาพทางการแพทย์ ใช้ในการตรวจจับเนื้องอก มะเร็ง และความผิดปกติต่างๆ [3] การตรวจจับใบหน้าเพื่อยืนยันตัวตน ในปัจจุบันการพัฒนาโมเดลตรวจจับวัตถุมุ่งเน้นให้มีความแม่นยำสูง และสามารถประมวลผลแบบเรียลไทม์ได้อย่างมีประสิทธิภาพ โดยอาศัยการประมวลผลด้วยหน่วยประมวลผลกราฟิก (GPU) และเทคนิคเชิงลึก (Deep Learning)

แนวทางการตรวจจับวัตถุสามารถแบ่งออกเป็น 2 ประเภทหลัก ได้แก่

1. Two-Stage Detection เช่น R-CNN Faster และ R-CNN ซึ่งเน้นความแม่นยำสูง โดยทำการเลือกบริเวณที่น่าสนใจก่อน แล้วจึงจำแนกประเภทของวัตถุ แต่ใช้เวลาในการประมวลผลนานกว่า
2. One-Stage Detection เช่น YOLO และ SSD ซึ่งสามารถตรวจจับ และจำแนกวัตถุได้ภายในขั้นตอนเดียว ทำให้มีความรวดเร็ว เหมาะสำหรับงานแบบเรียลไทม์

นอกจากนี้ปัจจุบันมีการพัฒนาแนวทางที่ใช้ Vision Transformers (ViTs) ซึ่งนำโครงสร้างเครือข่ายประสาทเทียมแบบทรานส์ฟอร์เมอร์มาประยุกต์ใช้ในการตรวจจับวัตถุ เพื่อเพิ่มความสามารถในการเรียนรู้ความสัมพันธ์เชิงบริบทของภาพ โดยเฉพาะในการตรวจจับวัตถุขนาดเล็ก [4] และในสภาพแวดล้อมที่มีความซับซ้อนหรือมีแสงไม่เพียงพอ [5]-[7] เทคโนโลยีเหล่านี้มีบทบาทสำคัญในการพัฒนาระบบที่ต้องการทั้งความแม่นยำสูงและการตอบสนองแบบเรียลไทม์

ในงานวิจัยนี้ เลือกใช้แนวคิดแบบ One-Stage Detection เนื่องจากแอปพลิเคชัน Seeable ต้องทำงานแบบเรียลไทม์บนสมาร์ตโฟน และต้องตอบสนองต่อผู้ใช้งานอย่างต่อเนื่องเพื่อให้แอปพลิเคชันสามารถทำงานได้อย่างมีประสิทธิภาพ ไม่มีความล่าช้า



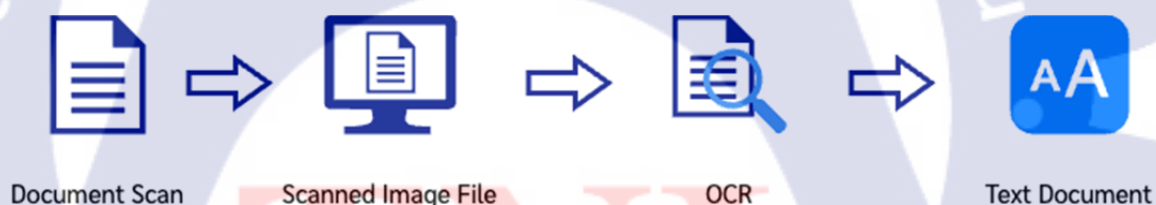
รูปที่ 2.1 ภาพตัวอย่างการตรวจจับวัตถุ [8]

2.1.3 การรู้จำอักขระด้วยแสง (Optical Character Recognition: OCR)

การรู้จำอักขระด้วยแสง เป็นเทคโนโลยีในสาขาคอมพิวเตอร์วิทัศน์ที่ใช้ในการแปลงข้อความจากภาพ เช่น เอกสารสแกน รูปภาพที่มีข้อความ หรือป้ายต่างๆ ให้เป็นข้อความในรูปแบบดิจิทัล ดังแสดงในรูปที่ 2.2 กระบวนการรู้จำอักขระด้วยแสง โดยทั่วไปประกอบด้วยขั้นตอนการตรวจจับบริเวณข้อความ (Text Detection) และการรู้จำตัวอักษร (Text Recognition)

โดยเทคโนโลยี OCR ถูกนำไปประยุกต์ใช้ในหลายด้าน เช่น ระบบจัดการเอกสารอัตโนมัติ [9] ระบบจดจำป้ายทะเบียนรถ และระบบช่วยเหลือผู้พิการทางสายตาสำหรับอ่านข้อความจากสภาพแวดล้อมจริง [10] เครื่องมือ OCR ที่ได้รับความนิยม ได้แก่ Tesseract OCR ของ Google และ Microsoft OCR ซึ่งสามารถรองรับหลายภาษา และมีความสามารถในการจดจำตัวอักษรที่มีรูปแบบและฟอนต์ที่หลากหลาย โดยอาศัยเทคนิคการเรียนรู้เชิงลึกเพื่อเพิ่มความแม่นยำของผลลัพธ์

งานวิจัยของ Q. Nguyen et al. [11] ได้นำเสนอวิธีการแก้ไขข้อผิดพลาดของข้อความจาก OCR โดยใช้แนวทางแบบไม่ต้องมีข้อมูลกำกับ (Unsupervised) ผ่านอัลกอริทึม Hill Climbing ร่วมกับการแก้ไขตัวอักษร เพื่อค้นหาคำที่ถูกต้องจากผลลัพธ์ OCR ที่คลาดเคลื่อน วิธีนี้ช่วยลดข้อผิดพลาดที่เกิดจากคุณภาพภาพหรือข้อจำกัดของระบบ โดยไม่ต้องใช้ข้อมูลฝึกแบบจับคู่ และอาศัยบริบททางภาษา เช่น n-gram และความคล้ายของคำ เพื่อเพิ่มความแม่นยำ ทำให้สามารถปรับปรุงคุณภาพข้อความ OCR ได้อย่างมีประสิทธิภาพ



รูปที่ 2.2 ขั้นตอนการตรวจจับอักขระด้วยแสง

ในงานวิจัยนี้ เทคโนโลยี OCR ถูกนำมาใช้เป็นส่วนหนึ่งของแอปพลิเคชัน Seeable เพื่อสนับสนุนการรับรู้ข้อมูลของผู้พิการทางสายตา ผลลัพธ์ที่ได้จากกระบวนการ OCR จะถูกส่งต่อไปยังระบบสังเคราะห์เสียง (Text-to-Speech: TTS) เพื่อแปลงข้อความเป็นเสียง และแจ้งข้อมูลแก่ผู้ใช้งานในรูปแบบเสียง ซึ่งเป็นช่องทางหลักในการสื่อสารของแอปพลิเคชัน ดังนั้น OCR จึงเป็นองค์ประกอบสำคัญที่ช่วยเพิ่มความสามารถในการรับรู้ข้อมูลจากสภาพแวดล้อมจริง ทำให้ผู้ใช้งานสามารถเข้าถึงข้อมูลที่เป็นตัวอักษรได้โดยไม่ต้องพึ่งพาการมองเห็น

2.1.4 การรู้จำเสียงพูด (Speech Recognition)

การรู้จำเสียงพูด เป็นเทคโนโลยีที่ใช้ในการแปลงสัญญาณเสียงของมนุษย์ให้เป็นข้อความดิจิทัล โดยอาศัยกระบวนการวิเคราะห์สัญญาณเสียง (Speech Signal Processing) ร่วมกับแบบจำลองการเรียนรู้เชิงลึก (Deep Learning Models) เพื่อเรียนรู้รูปแบบของหน่วยเสียง (Phoneme) คำ และโครงสร้างประโยคอย่างมีประสิทธิภาพ

เทคโนโลยีการรู้จำเสียงพูดถูกนำไปประยุกต์ใช้ในหลากหลายแอปพลิเคชัน เช่น ผู้ช่วยเสียงอัจฉริยะ (Google Assistant, Siri, Alexa), ระบบสั่งงานด้วยเสียงในรถยนต์ และซอฟต์แวร์ช่วยเหลือนักพิการ [12] ปัจจุบันการพัฒนามุ่งเน้นไปที่การรองรับหลายภาษา การลดผลกระทบจากเสียงรบกวน และการเพิ่มความแม่นยำในการใช้งานในสภาพแวดล้อมจริง

ในบริบทของงานวิจัยนี้ เทคโนโลยีการรู้จำเสียงพูดถูกนำมาใช้เพื่อสนับสนุนการโต้ตอบของผู้พิการทางสายตา กับแอปพลิเคชัน โดยช่วยให้ผู้ใช้งานสามารถสั่งงานหรือเลือกจุดหมายปลายทาง หรือควบคุมการทำงานของระบบผ่านเสียงได้อย่างสะดวก ซึ่งช่วยลดการพึ่งพาการสัมผัสหน้าจอและเพิ่มประสิทธิภาพในการใช้งาน ทั้งยังสอดคล้องกับแนวคิดการออกแบบเพื่อการเข้าถึง

2.1.5 การสังเคราะห์เสียงพูด (Text-To-Speech : TTS)

การสังเคราะห์เสียงพูด เป็นเทคโนโลยีที่ใช้ในการแปลงข้อความดิจิทัลให้เป็นเสียงพูดที่สามารถเข้าใจได้ โดยกระบวนการทำงานประกอบด้วยขั้นตอนสำคัญ ได้แก่ การวิเคราะห์ข้อความ (Text Analysis) การแปลงเป็นหน่วยเสียง (Phoneme Conversion) และการสร้างสัญญาณเสียง (Speech Synthesis)

ในปัจจุบัน ระบบการสังเคราะห์เสียงพูดได้รับการพัฒนาโดยอาศัยแบบจำลองโครงข่ายประสาทเทียมเชิงลึก เช่น Tacotron และ WaveNet ซึ่งสามารถสร้างเสียงที่มีความเป็นธรรมชาติและใกล้เคียงเสียงมนุษย์มากขึ้น อีกทั้งยังสามารถปรับโทนเสียง ความเร็ว และสำเนียงให้เหมาะสมกับบริบทของการใช้งาน ดังตัวอย่างงานวิจัย [13]-[14] ที่เป็นการนำเทคโนโลยีนี้มาใช้ในการนำทางให้ผู้พิการทางสายตา รวมถึงการพัฒนาผู้ช่วยเสียงอัจฉริยะ

เทคโนโลยีการสังเคราะห์เสียงพูดมีบทบาทสำคัญในงานด้านการเข้าถึงข้อมูล (Accessibility) โดยเฉพาะสำหรับผู้พิการทางสายตา เนื่องจากช่วยแปลงข้อมูลที่เป็นข้อความให้เป็นเสียงได้แบบเรียลไทม์ ในงานวิจัยนี้ ระบบสังเคราะห์เสียงพูดถูกนำมาใช้เป็นช่องทางหลักในการสื่อสารกับผู้ใช้งาน โดยใช้สำหรับแจ้งเตือนสิ่งกีดขวาง ให้คำแนะนำในการนำทาง อ่านผลลัพธ์จากระบบตรวจจับอักขระด้วยแสง และแจ้งสถานะของระบบ เช่น การเริ่มต้นนำทางหรือการถึงจุดหมายปลายทาง

2.1.6 ความเป็นจริงเสริม (Augmented Reality : AR)

ความเป็นจริงเสริม เป็นเทคโนโลยีที่ผสานข้อมูลดิจิทัลหรือวัตถุเสมือนเข้ากับสภาพแวดล้อมจริงแบบเรียลไทม์ โดยอาศัยอุปกรณ์ในการแสดงผล เช่น สมาร์ทโฟน แท็บเล็ต หรืออุปกรณ์สวมใส่อื่นๆ ทำให้ผู้ใช้งานสามารถมองเห็นข้อมูลดิจิทัลซ้อนทับอยู่บนสภาพแวดล้อมจริงได้อย่างเป็นธรรมชาติ เทคโนโลยีนี้ช่วยเพิ่มประสบการณ์การรับรู้ข้อมูล และการโต้ตอบสิ่งแวดล้อมได้อย่างมีประสิทธิภาพ

ตัวอย่างการประยุกต์ใช้เทคโนโลยีความเป็นจริงเสริม ปรากฏในงานวิจัย [15] ที่เป็นการนำเทคโนโลยีความเป็นจริงเสริมมาอยู่บนแอปพลิเคชันบนสมาร์ทโฟน เพื่อยกระดับประสบการณ์การท่องเที่ยวพิพิธภัณฑ์ โดยใช้ AR แสดงข้อมูลดิจิทัลควบคู่กับวัตถุที่จัดแสดงจริง ช่วยให้ผู้ใช้เข้าใจเนื้อหาได้ดียิ่งขึ้น ซึ่งมีความแตกต่างกับความเป็นจริงเสมือน (Virtual Reality: VR) ที่เป็นการสร้างสภาพแวดล้อมเสมือนขึ้นมาแทนที่โลกจริงทั้งหมด ดังตัวอย่างงานวิจัย [16] ที่เป็นการนำเทคโนโลยีความเป็นจริงเสมือนมาสร้างนักบ๊อบเสมือนในสภาพแวดล้อมความเป็นจริงเสมือนสำหรับผู้ป่วยโรคพาร์กินสัน เพื่อเพิ่มแรงจูงใจของผู้ป่วยในการทำกายภาพบำบัดผ่านประสบการณ์เสมือนจริง

2.1.7 มาร์กเกอร์ความเป็นจริงเสริม (Augmented Reality Marker: AR Marker)

มาร์กเกอร์ความเป็นจริงเสริม เป็นสัญลักษณ์หรือรูปแบบที่ถูกออกแบบมาเพื่อใช้เป็นจุดอ้างอิงตำแหน่งต่างๆ ภายในระบบความเป็นจริงเสริม โดยมาร์กเกอร์แต่ละตำแหน่งสามารถกำหนดรหัสหรือข้อมูลเฉพาะที่สามารถระบุพิกัดในสภาพแวดล้อมจริงได้อย่างชัดเจน เมื่อกล้องของอุปกรณ์ตรวจจับมาร์กเกอร์ ระบบจะสามารถนำข้อมูลดังกล่าวไปใช้ในการคำนวณตำแหน่ง และทิศทางของผู้ใช้งานได้อย่างแม่นยำ ดังตัวอย่างงานวิจัย [17] ที่นำเสนอการนำมาร์กเกอร์ความเป็นจริงเสริมมาอ้างอิงตำแหน่งของรถเข็นแต่ละคันภายในโรงงานอุตสาหกรรม

ในงานวิจัยนี้ มาร์กเกอร์ความเป็นจริงเสริมถูกนำมาใช้เป็นองค์ประกอบหลักของระบบนำทางภายในอาคาร โดยทำหน้าที่เป็นจุดอ้างอิงตำแหน่งสำหรับระบุตำแหน่งเริ่มต้นของผู้ใช้งาน อัปเดตตำแหน่งระหว่างการเคลื่อนที่ และตรวจสอบการออกนอกเส้นทาง การใช้มาร์กเกอร์ช่วยลดการพึ่งพาสัญญาณไร้สาย และเพิ่มความแม่นยำของการนำทางภายในอาคาร

2.1.8 การนำทางด้วยมาร์กเกอร์ความเป็นจริงเสริม (Augmented Reality Marker-Based Navigation)

การนำทางด้วยมาร์กเกอร์ความเป็นจริงเสริม เป็นวิธีการนำทางที่ใช้อำศัยมาร์กเกอร์ซึ่งถูกติดตั้งตามตำแหน่งต่างๆ ภายในอาคาร เป็นจุดอ้างอิงตำแหน่ง โดยระบบจะใช้กล้องของสมาร์ทโฟนในการตรวจจับมาร์กเกอร์เพื่อระบุตำแหน่งปัจจุบันของผู้ใช้งาน จากนั้นจึงประมวลผลข้อมูลเพื่อแสดงคำแนะนำในการนำทางหรือทิศทางการเคลื่อนที่ผ่านเทคโนโลยีความเป็นจริงเสริมบนสภาพแวดล้อมจริง

จุดเด่นของการนำทางโดยใช้มาร์กเกอร์ความเป็นจริงเสริม คือการยกระดับความแม่นยำของการนำทางภายในอาคาร (Indoor Navigation) เนื่องจากมาร์กเกอร์ทำหน้าที่เป็นจุดอ้างอิงที่ชัดเจน อีกทั้งช่วยลดข้อจำกัดด้านการพึ่งพาสัญญาณไร้สายที่มีความผันผวนในพื้นที่ปิด ส่งผลให้วิธีนี้เหมาะสมสำหรับอาคารที่มีโครงสร้างซับซ้อน และต้องการความถูกต้องในการนำทางสูง ดังตัวอย่างงานวิจัย [18] ซึ่งนำมาร์กเกอร์ความเป็นจริงเสริมมาใช้เป็นจุดอ้างอิงเชิงตำแหน่งในการระบุตำแหน่งภายในอาคารของโดรน เพื่อประมาณตำแหน่งและทิศทางการเคลื่อนที่ในสภาพแวดล้อมภายในอาคารได้อย่างแม่นยำโดยไม่ต้องพึ่งพาระบบ GPS

ในบริบทงานวิจัยนี้ เทคโนโลยีมาร์กเกอร์ความเป็นจริงเสริมถูกนำมาใช้เป็นกลไกหลักในการระบุตำแหน่งของผู้ใช้งานภายในอาคาร โดยอาศัยการตรวจจับมาร์กเกอร์ผ่านกล้องของสมาร์ทโฟนเพื่อกำหนดตำแหน่งปัจจุบัน จากนั้นข้อมูลตำแหน่งดังกล่าวจะถูกส่งไปยังระบบนำทางของแอปพลิเคชันเพื่อใช้ในการคำนวณเส้นทาง อัปเดตตำแหน่งระหว่างการเดินทางที่ และแสดงคำแนะนำในการเดินทางแก่ผู้ใช้งานในรูปแบบเสียง

2.1.9 การกำหนดเส้นทางในการนำทาง (Path Planning)

การกำหนดเส้นทางในการนำทาง เป็นกระบวนการวางแผนเส้นทางการเคลื่อนที่จากตำแหน่งเริ่มต้นไปยังจุดหมายปลายทาง โดยอาศัยข้อมูลโครงสร้างของพื้นที่ ทางเดิน และตำแหน่งอ้างอิงที่ได้กำหนดไว้ โดยเส้นทางที่ได้ควรมีความเหมาะสมในด้านระยะทาง ความปลอดภัย และความสะดวกในการเคลื่อนที่ เพื่อให้สามารถนำทางได้อย่างมีประสิทธิภาพ [19]-[20]

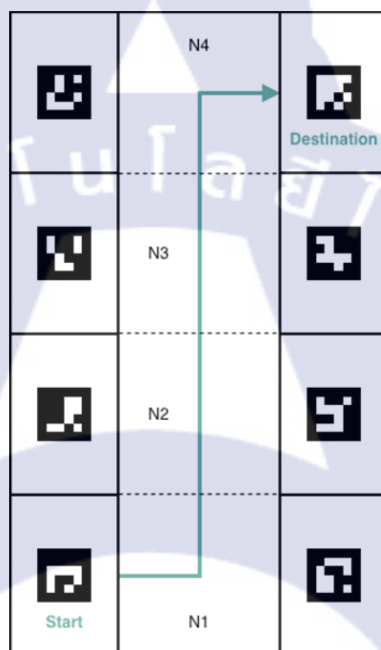
ในระบบนำทางด้วยมาร์กเกอร์ความเป็นจริงเสริม การกำหนดเส้นทางจะอาศัยตำแหน่งของมาร์กเกอร์ที่ติดตั้งภายในอาคารเป็นจุดอ้างอิง โดยมาร์กเกอร์แต่ละตำแหน่งจะถูกกำหนดพิกัดและความสัมพันธ์เชิงพื้นที่ไว้ในรูปแบบของแผนที่เชิงโหนด (Node-based Navigation) เมื่อผู้ใช้งานตรวจพบมาร์กเกอร์ ระบบจะทำการระบุตำแหน่งปัจจุบัน และนำข้อมูลดังกล่าวมาประมวลผลเพื่อกำหนดเส้นทางไปยังจุดหมายปลายทาง ดังเส้นทางในการนำทางในรูปที่ 2.3

การกำหนดเส้นทางในการนำทาง สามารถแบ่งออกเป็นขั้นตอนหลัก ดังต่อไปนี้

1. การเตรียมข้อมูลแผนที่ เป็นการเตรียมข้อมูลโครงสร้างของพื้นที่ภายในอาคาร และกำหนดตำแหน่งของมาร์กเกอร์แต่ละจุดให้สอดคล้องกับสภาพแวดล้อมจริง
2. การระบุตำแหน่งเริ่มต้น และจุดหมายปลายทาง ระบบจะทำการระบุตำแหน่งของผู้ใช้งานจากมาร์กเกอร์ที่ตรวจพบ และกำหนดจุดหมายปลายทางตามที่คุณเลือก
3. การคำนวณเส้นทาง ระบบจะคำนวณเส้นทางที่เหมาะสมระหว่างตำแหน่งเริ่มต้นและจุดหมาย โดยคำนึงถึงความต่อเนื่องของเส้นทาง และข้อจำกัดของพื้นที่

4. การปรับปรุงเส้นทาง เมื่อผู้ใช้งานเคลื่อนที่ และมีการตรวจพบมาร์กเกอร์ใหม่ ระบบ จะทำการปรับปรุงหรือคำนวณเส้นทางใหม่เพื่อให้สอดคล้องกับตำแหน่งปัจจุบัน

การกำหนดเส้นทางทำให้ผู้ใช้งานสามารถเดินทางไปยังจุดหมายได้อย่างมีประสิทธิภาพ และลดความสับสนในการเคลื่อนที่ภายในอาคาร อย่างไรก็ตามข้อจำกัดของการนำทางโดยวิธีนี้ คือ ความสมบูรณ์ของข้อมูลแผนที่ และตำแหน่งของมาร์กเกอร์ หากมีการเปลี่ยนแปลงสภาพแวดล้อม เช่น การปิดปรับปรุงพื้นที่ อาจส่งผลกระทบต่อความแม่นยำของเส้นทางที่กำหนด



รูปที่ 2.3 ตัวอย่างการกำหนดเส้นทางแบบโหนด และมาร์กเกอร์ความเป็นจริงเสริม

2.1.10 การประมาณระยะลึก (Depth Estimation)

การประมาณระยะลึกเป็นกระบวนการในด้านคอมพิวเตอร์วิทัศน์ (Computer Vision) ที่ใช้ในการคาดคะเนระยะทางระหว่างกล้องและวัตถุภายในภาพ หรือการสร้างข้อมูลเชิงลึก (Depth Information) ของฉาก เพื่อให้ระบบสามารถเข้าใจลักษณะเชิงสามมิติของสิ่งแวดล้อมได้ เทคโนโลยีนี้มีบทบาทสำคัญในหลายด้าน เช่น การสร้างแบบจำลองสามมิติ ความเป็นจริงเสริมและความเป็นจริงเสมือน (AR/VR) การนำทางของยานยนต์ไร้คนขับ และหุ่นยนต์

วิธีการประมาณระยะลึกสามารถแบ่งออกได้หลายแนวทาง ได้แก่

1. Stereo Vision เป็นการใช้อุปกรณ์จากกล้อง 2 ตัวที่วางคู่กัน จากนั้นจะคำนวณความแตกต่างเชิงตำแหน่ง (Disparity) ของวัตถุในภาพทั้งสอง และใช้ตรีโกณมิติ (Triangulation) ในการ

ประมาณระยะทาง แต่อย่างไรก็ตามวิธีนี้มีข้อจำกัดในสภาพแวดล้อมที่มีพื้นผิวราบเรียบ สีสันสม่ำเสมอ หรือมีแสงน้อย

2. Structure from Motion เป็นการนำการเคลื่อนที่ของกล้องเพื่อหาความลึกจากความแตกต่างระหว่างภาพหลายมุมมอง โดยที่ภาพจะมาจากกล้องตัวเดียว หรือหลายตัวก็ได้ แต่อย่างไรก็ตามกระบวนการจะค่อนข้างซับซ้อน ต้องใช้หลายภาพ และมีความไวต่อความคลาดเคลื่อนของกล้อง รวมถึงไม่เหมาะกับสภาพแวดล้อมที่มีการเคลื่อนที่ของวัตถุเยอะ

3. Monocular Depth Estimation เป็นการนำภาพจากกล้องเพียงตัวเดียวเพื่อคาดคะเนความลึก โดยอาศัยโมเดลการเรียนรู้เชิงลึก (Deep Learning) ในการคาดคะเนระยะลึกจากลักษณะของภาพ ตัวอย่างโมเดลที่ได้รับความนิยม ได้แก่ MiDaS [20], DenseDepth, Monodepth2

การประมาณระยะลึกแบบ Monocular Depth Estimation สามารถแบ่งได้เป็น 2 ประเภทหลัก ได้แก่

1. Absolute (Metric) Depth Estimation เป็นการประมาณระยะลึกในเชิงปริมาณที่ให้ค่าตัวเลขระยะทางจริง โดยที่ผลลัพธ์จะเป็นแผนที่ความลึก (Depth Map) ที่มีค่าตัวเลขแสดงถึงระยะห่างที่สัมพันธ์กับโลกจริง

2. Relative Depth Estimation เป็นการประมาณระยะลึกในเชิงสัมพัทธ์ โดยไม่จำเป็นต้องระบุค่าระยะทางที่แน่นอน แต่จะบอกได้ว่าวัตถุ หรือบริเวณใดใกล้ หรือไกลกว่ากันในภาพ

2.1.11 การพัฒนาแอปพลิเคชันด้วยเฟรมเวิร์คข้ามแพลตฟอร์ม (Cross-Platform Framework)

การพัฒนาแอปพลิเคชันด้วยเฟรมเวิร์คข้ามแพลตฟอร์ม เป็นแนวทางที่ช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่ทำงานได้ทั้งบน Android และ iOS โดยใช้โค้ดชุดเดียวกัน ลดเวลาและค่าใช้จ่ายในการพัฒนา เฟรมเวิร์คยอดนิยมในปัจจุบัน ได้แก่ Flutter, React Native ซึ่งมีจุดเด่นด้านประสิทธิภาพและความง่ายในการใช้งาน การใช้เฟรมเวิร์คข้ามแพลตฟอร์มช่วยให้สามารถออกแบบแอปพลิเคชันที่มีหน้าตาสวยงาม ทำงานได้รวดเร็ว และรองรับการอัปเดตได้ง่าย เหมาะสำหรับการพัฒนาแอปที่ต้องการรองรับผู้ใช้จำนวนมากในระบบปฏิบัติการที่หลากหลาย

2.2 งานวิจัยที่เกี่ยวข้อง

จากการศึกษางานวิจัยที่เกี่ยวข้องเกี่ยวกับการพัฒนาแอปพลิเคชันสำหรับผู้พิการทางสายตา แบ่งหมวดหมู่ออกเป็น 4 หมวด คือ การศึกษางานวิจัยทางด้านการพัฒนาการนำทางโดยใช้มาร์กเกอร์ ความจริงเป็นเสริม เทคโนโลยีการตรวจจับวัตถุ การตรวจจับอักขระด้วยแสง และงานวิจัยที่เกี่ยวข้องกับ เรื่องการพัฒนาแอปพลิเคชันสำหรับช่วยเหลือผู้พิการทางสายตา

งานวิจัยทางด้านการพัฒนาการนำทางโดยใช้มาร์กเกอร์ความเป็นจริงเสริม

X. Yu and J. Saniie [21] ได้นำเสนอระบบ Visual Impairment Spatial Awareness (VISA) ซึ่งเป็นระบบช่วยเหลือผู้พิการทางสายตาสำหรับการนำทางภายในอาคาร และการทำกิจกรรมในชีวิตประจำวัน โดยมีแนวคิดในการพัฒนาระบบที่ผสานการระบุตำแหน่ง การนำทาง การตรวจจับวัตถุ และการรับรู้สภาพแวดล้อมเข้าไว้ด้วยกัน เพื่อแก้ไขข้อจำกัดของระบบนำทางแบบดั้งเดิมที่ไม่เหมาะสมกับสภาพแวดล้อมภายในอาคาร ระบบ VISA ใช้มาร์กเกอร์ความเป็นจริงเสริมประเภท ArUco Marker เป็นจุดอ้างอิงเชิงตำแหน่ง ร่วมกับการประมวลผลภาพจากกล้อง RGB-D เพื่อระบุตำแหน่ง และทิศทางของผู้ใช้งาน โดยสร้างแผนที่เชิงโหนด และคำนวณเส้นทางด้วยอัลกอริทึม Dijkstra นอกจากนี้ ระบบยังผสานการตรวจจับวัตถุด้วยโมเดล YOLOv8 และการประเมินค่าความลึกของวัตถุ เพื่อแจ้งเตือน และหลีกเลี่ยงสิ่งกีดขวางระหว่างการเคลื่อนที่ ในด้านการใช้งาน ระบบรองรับการโต้ตอบด้วยเสียงผ่านเทคโนโลยีแปลงข้อความเป็นเสียง และแปลงเสียงเป็นข้อความ รวมถึงความสามารถในการอ่านตัวอักษรจากสภาพแวดล้อมจริง ผลการทดสอบแสดงให้เห็นว่าระบบสามารถช่วยเพิ่มความปลอดภัย การรับรู้เชิงพื้นที่ และความเป็นอิสระในการดำรงชีวิตประจำวันของผู้พิการทางสายตาได้อย่างมีประสิทธิภาพ

B. S. C. Putra et al. [22] ได้นำเสนอระบบนำทางภายในอาคารด้วยเทคโนโลยีความเป็นจริงเสริม (Augmented Reality: AR) โดยแก้ไขข้อจำกัดของระบบนำทางแบบเดิมที่ต้องอาศัยการสร้างแผนที่ล่วงหน้า งานวิจัยนี้เสนอแนวคิดการสร้างแผนที่แบบโหนด (Node-Based Mapping) ซึ่งผู้ใช้งานสามารถกำหนดเส้นทางการนำทางได้แบบเรียลไทม์ผ่านสมาร์ทโฟน ระบบที่พัฒนาขึ้นใช้ Google ARCore สำหรับการรับรู้เชิงพื้นที่ ร่วมกับการสร้างโหนดในตำแหน่งที่สามารถเดินได้ และใช้ NavMesh กับอัลกอริทึม A* ในการคำนวณเส้นทาง นอกจากนี้ ยังใช้มาร์กเกอร์ในรูปแบบ QR Code เพื่อกำหนดตำแหน่งเริ่มต้น และจุดหมายปลายทางของผู้ใช้งาน ผลการประเมินความสามารถในการใช้งานด้วยแบบวัด Handheld Augmented Reality Usability Scale (HARUS) พบว่าระบบมีคะแนนความสามารถในการใช้งานเฉลี่ย 81.98 อยู่ในระดับดีมาก แสดงให้เห็นว่าระบบนำทางมีความยืดหยุ่นสามารถใช้งานได้มีประสิทธิภาพ

ตารางที่ 2.1 สรุปงานวิจัยที่เกี่ยวกับเรื่องการพัฒนาการนำทางโดยใช้มาร์กเกอร์ความเป็นจริงเสริม

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Visual Impairment Spatial Awareness System for Indoor Navigation and Daily Activities	พัฒนาระบบนำทางภายในอาคาร โดยใช้ ArUco Marker เป็นจุดอ้างอิงเชิงตำแหน่ง ร่วมกับกล้อง RGB-D เพื่อระบุตำแหน่ง และกำหนดเส้นทาง สามารถแจ้งเตือนสิ่งกีดขวาง อ่านตัวหนังสือ และสั่งการด้วยเสียง	ผลการทดสอบแสดงให้เห็นว่าระบบสามารถช่วยผู้พิการทางสายตาในการนำทางภายในอาคาร ตรวจจับสิ่งกีดขวาง รวมถึงอ่านข้อมูลจากสภาพแวดล้อมได้อย่างมีประสิทธิภาพ
Adaptive AR Navigation: Real-Time Mapping for Indoor Environment Using Node Placement and Marker Localization	พัฒนาระบบนำทางภายในอาคารด้วยเทคโนโลยี AR โดยใช้ Google ARCore สำหรับการรับรู้เชิงพื้นที่ ผู้ใช้งานสามารถวาง และเชื่อมโยงโหนดเพื่อสร้างแผนที่แบบเรียลไทม์ ระบบใช้ NavMesh และ A* ในการกำหนดเส้นทาง และใช้ QR Code เป็นมาร์กเกอร์สำหรับระบุตำแหน่งเริ่มต้น และจุดหมายปลายทาง	ผลการประเมินด้วย HARUS พบว่ามีคะแนนความสามารถในการใช้งานเฉลี่ย 81.98 อยู่ในระดับดีมาก แสดงให้เห็นว่าระบบมีความยืดหยุ่น ใช้งานได้จริง และสามารถนำทางในอาคารด้วย AR อย่างมีประสิทธิภาพ

งานวิจัยทางด้านเทคโนโลยีการพัฒนาระบบจัดลำดับความสำคัญ

F. Vitale and C. Roncoli [23] ได้นำเสนอแนวทางการหลบหลีกสิ่งกีดขวางสำหรับระบบขับเคลื่อนอัตโนมัติ โดยใช้แนวคิด Task Prioritization เพื่อจัดลำดับความสำคัญของงานควบคุมระหว่างการติดตามเส้นทาง (Trajectory Tracking) และการหลบสิ่งกีดขวาง (Obstacle Avoidance) ซึ่งถูกพัฒนาบนพื้นฐานของ Model Predictive Control (MPC) เพื่อให้สามารถคำนวณการควบคุมล่วงหน้าได้ในเวลาจำกัด พร้อมทั้งใช้เทคนิค Null-Space Projection เพื่อผสานหลาย Objective เข้าด้วยกันโดยไม่รบกวน Task ที่มีลำดับความสำคัญสูงกว่า ระบบมีการกำหนดข้อจำกัดด้านความปลอดภัยในรูปแบบ Safe Distance Constraint แบบสองมิติ เพื่อให้สามารถประเมินระยะห่างจากสิ่งกีดขวางได้แม่นยำมาก

ขึ้นเมื่อเทียบกับวิธีแบบหนึ่งมิติ นอกจากนี้ยังมีการทดสอบในหลายสถานการณ์ เช่น การหลบสิ่งกีดขวางแบบคงที่และแบบเคลื่อนที่ในสภาพแวดล้อมเมืองและทางหลวง ผลการทดลองแสดงให้เห็นว่าระบบสามารถตอบสนองต่อสิ่งกีดขวางได้แบบเรียลไทม์ ลดความจำเป็นในการวางแผนเส้นทางใหม่ และเพิ่มเสถียรภาพในการควบคุมยานพาหนะได้อย่างมีประสิทธิภาพ

H. Wang et al. [24] ได้นำเสนอวิธีการนำทางของหุ่นยนต์เคลื่อนที่โดยใช้ Reinforcement Learning (Proximal Policy Optimization: PPO) ร่วมกับ Control Barrier Function (CBF) เพื่อสร้างระบบที่มีทั้งความสามารถในการเรียนรู้และความปลอดภัยในระดับสูง โดย RL จะทำหน้าที่เรียนรู้นโยบายการเคลื่อนที่ที่เหมาะสมจากการโต้ตอบกับสภาพแวดล้อม ขณะที่ CBF ทำหน้าที่เป็น dynamic safety filter เพื่อปรับ action ที่ได้จาก RL ให้เป็นไปตามข้อจำกัดด้านความปลอดภัยแบบเรียลไทม์ผ่านการแก้ปัญหา quadratic programming (QP) ระบบสามารถประเมินความเสี่ยงของการชนจากสถานะของหุ่นยนต์และสิ่งกีดขวาง และปรับพฤติกรรมเคลื่อนที่ให้หลีกเลี่ยงสถานการณ์อันตรายได้โดยไม่สูญเสียประสิทธิภาพของการเรียนรู้ ผลการทดลองในสภาพแวดล้อมจำลอง ROS และ Gazebo แสดงให้เห็นว่าระบบสามารถลดอัตราการชน เพิ่มอัตราความสำเร็จในการนำทาง และรักษาความเสถียรของแนวการเคลื่อนที่ได้ดีกว่าวิธี RL แบบเดิม

Z. Long et al. [25] ได้นำเสนอวิธีการวางแผนเส้นทางสำหรับหุ่นยนต์เคลื่อนที่ในสภาพแวดล้อมที่มีมนุษย์ โดยใช้ Reinforcement Learning บนพื้นฐาน Markov Decision Process (MDP) เพื่อสร้างชุดของเส้นทาง (Multi-Path Planning) ที่เป็นไปได้ จากนั้นใช้ Monte Carlo Simulation เพื่อจำลองพฤติกรรมเคลื่อนที่ของมนุษย์ในอนาคต และประเมินความเสี่ยงของแต่ละเส้นทางโดยพิจารณาความน่าจะเป็นของการชนร่วมกับความไม่แน่นอนของตำแหน่งมนุษย์ ระบบจะเลือกเส้นทางที่มีความเสี่ยงต่ำที่สุดผ่าน Decision Unit ทำให้สามารถลดการเปลี่ยนเส้นทางแบบฉับพลันและลดภาระในการ Re-planning ได้ นอกจากนี้ยังสามารถปรับตัวต่อสภาพแวดล้อมที่มีความไม่แน่นอนสูง เช่น พื้นที่ที่มีคนเดินจำนวนมาก ผลการทดลองแสดงให้เห็นว่าวิธีนี้สามารถลดจำนวนการชนได้อย่างมีนัยสำคัญ และเพิ่มความปลอดภัยในการนำทางของหุ่นยนต์

K. Moller et al. [26] ได้นำเสนอวิธีการวางแผนการเคลื่อนที่สำหรับยานยนต์อัตโนมัติในสภาพแวดล้อมเมืองที่มีความซับซ้อน โดยมุ่งเน้นการหลีกเลี่ยงความเสี่ยงจากคนเดินเท้า ผ่านการใช้ Probabilistic Risk Modeling เพื่อคำนวณค่าความเสี่ยงจากการชน ซึ่งนิยามเป็นผลคูณของ Collision Probability และ Harm โดย Collision Probability ถูกประเมินจากการคาดการณ์ตำแหน่งของคนเดินเท้าในอนาคตด้วยแบบจำลองการกระจายแบบ Bivariate Gaussian distribution ขณะที่ Harm ถูกคำนวณจากปัจจัยทางกายภาพ เช่น ความเร็ว มวล และมุมการชน นอกจากนี้ยังมีการใช้ Logistic Regression Model เพื่อประเมินระดับความรุนแรงของอุบัติเหตุ วิธีการนี้ช่วยให้ระบบสามารถเลือกเส้นทางที่ลดความเสี่ยง

โดยรวมได้อย่างมีประสิทธิภาพ แม้ในสภาพแวดล้อมที่มีความไม่แน่นอนสูง ผลการทดลองแสดงให้เห็นว่าสามารถลดความเสี่ยงจากการชนและเพิ่มความปลอดภัยในการขับขี่ในเขตเมืองได้อย่างมีนัยสำคัญ

G. Sun et al. [27] ได้นำเสนอการปรับปรุงโมเดลตรวจจับวัตถุ YOLOv5 โดยเพิ่ม Convolutional Block Attention Module (CBAM) และเทคนิค Context Feature Fusion เพื่อเพิ่มประสิทธิภาพในการตรวจจับวัตถุในสภาพแวดล้อมที่มีความซับซ้อน โดย CBAM จะทำงานผ่านสองส่วนหลักคือ Channel Attention ซึ่งช่วยเลือก Feature Map ที่สำคัญ และ Spatial Attention ซึ่งช่วยระบุบริเวณที่มีวัตถุสำคัญในภาพ นอกจากนี้ยังมีการผสานข้อมูลเชิงบริบทจากหลายระดับของ Feature Map เพื่อช่วยให้โมเดลสามารถตรวจจับวัตถุขนาดเล็กและวัตถุที่อยู่ในฉากที่มีความหนาแน่นสูงได้ดียิ่งขึ้น ผลการทดลองบนชุดข้อมูล Pascal VOC แสดงให้เห็นว่าโมเดลที่เสนอสามารถเพิ่มค่า Mean Average Precision (mAP) ได้ดีกว่า YOLOv5 แบบดั้งเดิม และยังคงประสิทธิภาพในการประมวลผลแบบเรียลไทม์ แม้จะมีความซับซ้อนของโมเดลเพิ่มขึ้นเล็กน้อย

ตารางที่ 2.2 สรุปงานวิจัยเกี่ยวกับเรื่องเทคโนโลยีการพัฒนาระบบจัดลำดับความสำคัญ

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Reference Tracking Optimization With Obstacle Avoidance via Task Prioritization for Automated Driving	ใช้ Model Predictive Control (MPC) เพื่อควบคุมการเคลื่อนที่ โดยให้ระบบสามารถติดตามเส้นทางและหลบสิ่งกีดขวางได้พร้อมกัน พร้อมใช้ Task Prioritization เพื่อให้การหลบสิ่งกีดขวางมีความสำคัญสูงกว่า และใช้ Null-Space Projection เพื่อรวมการทำงานของหลายเป้าหมาย รวมถึงกำหนดระยะปลอดภัยแบบ 2 มิติ เพื่อเพิ่มความแม่นยำในการหลบหลีก	ระบบสามารถหลบสิ่งกีดขวางได้แบบ Real-time โดยไม่ต้องวางแผนเส้นทางใหม่ ลด Latency ในการคำนวณ เพิ่มเสถียรภาพของ Trajectory และเพิ่มความปลอดภัยในสภาพแวดล้อมที่มีทั้งวัตถุคงที่และเคลื่อนที่

ตารางที่ 2.2 สรุปงานวิจัยเกี่ยวกับเรื่องเทคโนโลยีการพัฒนาระบบจัดลำดับความสำคัญ (ต่อ)

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Risk-Aware Reinforcement Learning with Dynamic Safety Filter for Collision Risk Mitigation in Mobile Robot Navigation	ใช้ Reinforcement Learning (PPO) เพื่อเรียนรู้การเคลื่อนที่ของหุ่นยนต์ และใช้ Control Barrier Function (CBF) เพื่อปรับการเคลื่อนที่ให้ปลอดภัย โดยระบบจะปรับค่าการเคลื่อนที่แบบเรียลไทม์ผ่านการคำนวณแบบ Quadratic Programming (QP) และพิจารณาความเสี่ยงจากระยะและการเคลื่อนที่ของสิ่งกีดขวาง	ระบบสามารถลดการชนได้อย่างชัดเจน และเพิ่มความสำเร็จในการนำทาง หุ่นยนต์สามารถเคลื่อนที่ได้อย่างปลอดภัยมากขึ้น และสามารถทำงานได้ดีในสภาพแวดล้อมที่มีความซับซ้อนและมีสิ่งกีดขวางเคลื่อนที่
Human-Risk-Aware Safe Path Planning Based on Reinforcement Learning for Autonomous Mobile Robots	ใช้ Reinforcement Learning (MDP) เพื่อสร้างเส้นทางหลายทางเลือก จากนั้นใช้ Monte Carlo simulation เพื่อจำลองการเคลื่อนที่ของมนุษย์ในอนาคต และประเมินความเสี่ยงของแต่ละเส้นทาง โดยเลือกเส้นทางที่มีความเสี่ยงต่ำที่สุดผ่านระบบตัดสินใจ	ระบบสามารถลดจำนวนการชนได้มากกว่า 70% เพิ่มอัตราความสำเร็จของภารกิจ และลดความจำเป็นในการเปลี่ยนเส้นทางบ่อยๆ และเหมาะกับสภาพแวดล้อมที่มีความไม่แน่นอน เช่น พื้นที่ที่มีคนจำนวนมาก

ตารางที่ 2.2 สรุปงานวิจัยเกี่ยวกับเรื่องเทคโนโลยีการพัฒนาระบบจัดลำดับความสำคัญ (ต่อ)

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Pedestrian-Aware Motion Planning for Autonomous Driving in Complex Urban Scenarios	ใช้การคำนวณความเสี่ยงแบบ Probabilistic Model โดยประเมินความน่าจะเป็นของการชนด้วย Gaussian Distribution และคำนวณความรุนแรงของอุบัติเหตุด้วย Logistic Regression จากนั้นนำค่าความเสี่ยงไปใช้ในการวางแผนเส้นทางให้หลีกเลี่ยงพื้นที่ที่มีความเสี่ยงสูง	ระบบสามารถประเมินความเสี่ยงจากคนเดินเท้าได้แม่นยำขึ้น และเลือกเส้นทางที่ปลอดภัยมากขึ้น ช่วยลดโอกาสการเกิดอุบัติเหตุ และเพิ่มความน่าเชื่อถือของระบบในสภาพแวดล้อมเมือง
An Image Object Detection Model Based on Mixed Attention Mechanism Optimized YOLOv5	ปรับปรุงโมเดล YOLOv5 โดยเพิ่ม Attention Mechanism (CBAM) ซึ่งช่วยให้โมเดลเลือกข้อมูลสำคัญในภาพได้ดีขึ้น และใช้ Context Feature Fusion เพื่อรวมข้อมูลจากหลายระดับของภาพ ทำให้เข้าใจบริบทของวัตถุได้ชัดเจนขึ้น โดยเฉพาะในภาพที่มีความซับซ้อน	โมเดลสามารถตรวจจับวัตถุได้แม่นยำขึ้น โดยเฉพาะวัตถุขนาดเล็กและหลายวัตถุในภาพ ลดความผิดพลาดในการตรวจจับ และยังสามารถทำงานได้ในระดับใกล้เคียง Real-time

งานวิจัยทางด้านเทคโนโลยีการตรวจจับวัตถุ

H. Yi et al. [28] ได้นำเสนออัลกอริทึมใหม่ชื่อ LAR-YOLOv8 ซึ่งพัฒนาต่อยอดจาก YOLOv8 เพื่อแก้ไขข้อจำกัดในการตรวจจับวัตถุที่มีขนาดเล็กจากภาพถ่าย ปัญหาหลักของการตรวจจับวัตถุที่มีขนาดเล็ก คือภาพมักจะมีรายละเอียดต่ำ พื้นหลังซับซ้อน และมีสัญญาณรบกวน ทำให้ประสิทธิภาพในการตรวจจับลดลง ดังนั้นงานวิจัยนี้จึงมุ่งเน้นการปรับปรุงโมเดลเพื่อเพิ่มความแม่นยำในการดึงคุณลักษณะ และการจำแนกวัตถุ LAR-YOLOv8 มีการปรับปรุงด้วย Local Enhanced Backbone เพื่อเพิ่มประสิทธิภาพในการดึงคุณลักษณะจากภาพที่มีรายละเอียดซับซ้อน ซึ่งทำงานร่วมกับ Attention-Guided Bidirectional

Feature Pyramid Network (AGBiFPN) ที่เป็นตัวช่วยในการตรวจจับวัตถุขนาดเล็กได้ดีขึ้น และ RIOU Loss Function ทำให้การประมาณกรอบของวัตถุมีความแม่นยำมากยิ่งขึ้น จากผลการทดลองพบว่า LAR-YOLOv8 มีประสิทธิภาพเหนือกว่า YOLOv8 ดั้งเดิมอย่างชัดเจน พบว่ามีความแม่นยำเฉลี่ย (mAP) เพิ่มขึ้นระหว่าง 2.7-4.6% มีการลดจำนวนพารามิเตอร์ลงถึง 40% ซึ่งทำให้โมเดลมีขนาดเล็กลง และใช้ทรัพยากรน้อยลง แต่ค่าความเร็วในการประมวลผลต่อเฟรม (FPS) จะลดลงเล็กน้อย แต่โดยรวมแล้วยังคงมีประสิทธิภาพสูง และเหมาะสมสำหรับการตรวจจับวัตถุขนาดเล็กจากภาพได้ดี

Z. Luo and Y. Tian [29] ได้นำเสนอการพัฒนาโมเดลตรวจจับวัตถุในภาพอินฟราเรด โดยปรับปรุงจากโมเดล YOLOv8s เพื่อเพิ่มความแม่นยำในการตรวจจับวัตถุในสภาพแวดล้อมที่ซับซ้อน และมีทัศนวิสัยต่ำ โดยโมเดลที่พัฒนาขึ้นมีชื่อว่า YOLOv8-ITA (Infrared Target Attention) โดยมีการปรับปรุง YOLOv8s ด้วยการเพิ่ม CPCA Attention Mechanism เพื่อให้โมเดลมุ่งเน้นไปยังพื้นที่ที่มีคุณลักษณะเด่นในภาพอินฟราเรด การแทนที่เลเยอร์ Downsampling เดิมด้วย CGBD Module ที่สามารถเก็บรักษาขอบเขต และดึงทั้งคุณลักษณะเฉพาะที่ และคุณลักษณะเชิงบริบทได้ดีกว่า และการใช้ WIoU Loss Function ซึ่งช่วยให้การประเมินกรอบวัตถุแม่นยำขึ้น และเพิ่มประสิทธิภาพการจำแนกประเภท ผลการทดลองแสดงให้เห็นว่า YOLOv8-ITA สามารถเพิ่มประสิทธิภาพความแม่นยำเฉลี่ย (mAP) ได้ 1.4% เมื่อเปรียบเทียบกับ YOLOv8s เดิม พร้อมกับปรับปรุง Precision และ Recall แสดงให้เห็นถึงความเหมาะสมต่อการประยุกต์ใช้จริงในงานระบบขับเคลื่อนอัตโนมัติ และการเฝ้าระวังการจราจร

ตารางที่ 2.3 สรุปงานวิจัยเกี่ยวกับเรื่องเทคโนโลยีการตรวจจับวัตถุ

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Small Object Detection Algorithm Based on Improved YOLOv8 for Remote Sensing	พัฒนา LAR-YOLOv8 เพื่อแก้ปัญหาการตรวจจับวัตถุขนาดเล็กในภาพถ่าย โดยปรับปรุงโครงสร้างด้วย Local Enhanced Backbone, Attention-Guided Bidirectional Feature Pyramid Network (AGBiFPN) และ RIOU Loss Function	เพิ่มความแม่นยำเฉลี่ยของโมเดล (mAP) 2.7-4.6% ลดพารามิเตอร์ลงมากกว่า 40% ทำให้ประสิทธิภาพสูงกว่า YOLOv8 เดิม แม้ค่าความเร็วในการประมวลผลต่อเฟรม (FPS) จะลดลงเล็กน้อย ซึ่งเหมาะสำหรับการตรวจจับวัตถุขนาดเล็ก

ตารางที่ 2.3 สรุปงานวิจัยเกี่ยวกับเรื่องเทคโนโลยีการตรวจจับวัตถุ (ต่อ)

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Improved Infrared Road Object Detection Algorithm Based on Attention Mechanism in YOLOv8	พัฒนา YOLOv8-ITA สำหรับ การตรวจจับวัตถุในภาพ อินฟราเรด เพื่อเพิ่มความแม่นยำในการตรวจจับวัตถุในสภาพแวดล้อมที่ซับซ้อน และมีทัศนวิสัยต่ำ โดยเพิ่ม CPCA Attention Mechanism, CGBD Module และ WIoU Loss Function	เพิ่มความแม่นยำเฉลี่ยของ โมเดล (mAP) 1.4% มีการปรับปรุง precision และ recall เหนือกว่า YOLOv8s และโมเดลอื่นๆ

งานวิจัยทางด้านเทคโนโลยีการตรวจจับอักขระด้วยแสง

P. Kiatphaisansophon et al. [30] ได้นำเสนอการพัฒนาแบบจำลองสำหรับการตรวจจับข้อความในเอกสารภาษาไทย โดยมีเป้าหมายเพื่อเพิ่มประสิทธิภาพของระบบการรู้จำอักขระด้วยแสง (Optical Character Recognition: OCR) งานวิจัยนี้มุ่งเน้นไปที่ขั้นตอนการตรวจจับข้อความ (Text Detection) ซึ่งเป็นขั้นตอนสำคัญของกระบวนการ OCR ที่ประกอบด้วยการตรวจจับข้อความ การรู้จำข้อความ และการสกัดข้อมูล งานวิจัยได้เสนอแนวทางใหม่โดยประยุกต์ใช้โมเดล Mask R-CNN ซึ่งเป็นโมเดลด้าน Instance Segmentation ที่สามารถตรวจจับข้อความแต่ละส่วนเป็นวัตถุอิสระ ส่งผลให้สามารถระบุกรอบข้อความ (Bounding Box) ได้อย่างแม่นยำและลดปัญหาการรวมหลายบรรทัดไว้ในกรอบเดียว (Multiline Problem) นอกจากนี้ แนวทางดังกล่าวยังช่วยลดความจำเป็นในการทำ Post-Processing ซึ่งเป็นข้อจำกัดของโมเดลเดิม เช่น CRAFT และ PMTD การทดลองดำเนินการกับเอกสารหลายประเภท เช่น สมุดบัญชีธนาคาร บัตรประชาชน ใบแจ้งหนี้ ใบเสร็จรับเงิน พาสปอร์ต และเอกสารราชการอื่นๆ ผลการทดลองแสดงให้เห็นว่าโมเดลที่นำเสนอมีความแม่นยำสูง สามารถปรับใช้กับเอกสารที่มีโครงสร้างหลากหลายได้อย่างมีประสิทธิภาพ อีกทั้งยังลดเวลาในการประมวลผลและการใช้ทรัพยากรของระบบ ทำให้เหมาะสมต่อการใช้งานจริง

T. Chumwatana and W. Rattana-umnuaychai [31] ได้นำเสนอเทคนิคการใช้การตรวจจับอักขระด้วยแสง (Optical Character Recognition : OCR) พร้อมด้วยการสกัดข้อมูล (Information Extraction) เพื่อแปลงข้อมูลจากเอกสารภาษาไทยให้อยู่ในรูปแบบดิจิทัลที่สามารถค้นหา และแก้ไขได้อย่างมีประสิทธิภาพ โดยมุ่งเน้นการประยุกต์ใช้กับเอกสารที่อยู่ในรูปแบบกระดาษ หรือไฟล์สแกนงานวิจัยนี้ใช้ Tesseract OCR สำหรับการรู้จำอักขระ พร้อมกับกระบวนการประมวลผลภาพ (Image Processing)

เช่น การทำภาพไบนารี การกรองสัญญาณรบกวน และการแบ่งบรรทัดข้อความ เพื่อที่จะสามารถรู้จำตัวอักษรได้ชัดเจนยิ่งขึ้น จากนั้นข้อมูลที่ถูกแปลงจะผ่านกระบวนการสกัดข้อมูล (Information Extraction) และการจัดระเบียบสารสนเทศ (Information Organization) เพื่อดึงข้อมูล และจัดเก็บลงในฐานข้อมูลด้วยโครงสร้างที่เหมาะสม ผลการทดลองพบว่าระบบสามารถดึงข้อมูลได้ด้วยความแม่นยำเฉลี่ย 75.38% สำหรับคุณลักษณะ (Attributes) และ 66.92% สำหรับค่าของข้อมูล (Values) โดยความถูกต้องจะขึ้นอยู่กับคุณภาพของเอกสารต้นฉบับ เช่น ความคมชัด ความสว่าง และระดับสัญญาณรบกวน

ตารางที่ 2.4 สรุปงานวิจัยเกี่ยวกับการตรวจจับอักขระด้วยแสง

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Efficient Text Bounding Box Identification Using Mask R-CNN: Case of Thai Documents	พัฒนาแบบจำลองตรวจจับกรอบข้อความสำหรับระบบ OCR โดยใช้ Mask R-CNN เพื่อตรวจจับข้อความเป็นวัตถุอิสระ ลดปัญหา Multiline และจัดขั้นตอน Post-Processing โดยทดสอบกับเอกสารหลากหลายประเภท เช่น บัตรประชาชน ใบเสร็จ ใบแจ้งหนี้ และหนังสือเดินทาง	ระบบสามารถตรวจจับกรอบข้อความได้อย่างแม่นยำ รองรับเอกสารหลายรูปแบบ ลดเวลาในการประมวลผลและการใช้ทรัพยากร โดยมีเวลาในการประมวลผลน้อยกว่า 1 วินาทีต่อภาพ และเหมาะสมต่อการใช้งานในระบบ OCR จริง
Using OCR Framework and Information Extraction for Thai Documents Digitalization	ใช้ Tesseract OCR ร่วมกับเทคนิคประมวลผลภาพ (Image Processing) เพื่อแยกข้อความ จากนั้นใช้การสกัดข้อมูล (Information Extraction) และการจัดระเบียบสารสนเทศ (Information Organization) เพื่อดึงข้อมูลจากเอกสารภาษาไทย	ระบบสามารถแปลงข้อความจากเอกสารภาษาไทยแล้วแปลงเป็นข้อความดิจิทัลด้วยความแม่นยำเฉลี่ย 75.38% สำหรับคุณลักษณะ (attributes) และ 66.92% สำหรับค่าของข้อมูล (values)

งานวิจัยทางด้านการพัฒนาแอปพลิเคชันสำหรับช่วยเหลือผู้พิการทางสายตา

M. Awad et al. [32] ได้นำเสนอแอปพลิเคชันบนระบบปฏิบัติการ Android ที่ออกแบบมาเพื่อช่วยเหลือผู้พิการทางการมองเห็น โดยรวบรวมฟังก์ชันที่สำคัญไว้ในแอปพลิเคชันเดียว ได้แก่ การตรวจจับแสง (Light Detection), การตรวจจับสี (Color Detection), การรู้จำวัตถุ (Object Recognition) และการรู้จำธนบัตร (Banknote Recognition) แอปพลิเคชันนี้ใช้ OpenCV สำหรับการประมวลผลภาพ (Image Processing), CNNdroid สำหรับการทำงานของโครงข่ายประสาทเทียม (Neural Network) บนอุปกรณ์สมาร์ตโฟน และ CraftAR SDK เพื่อรองรับการรู้จำธนบัตรแบบไม่ต้องเชื่อมต่อกับอินเทอร์เน็ต ฟังก์ชันทั้งหมดถูกออกแบบให้สามารถใช้งานได้ง่าย และทำงานร่วมกับ Text-to-Speech (TTS) เพื่อแปลงผลลัพธ์เป็นเสียงให้กับผู้ใช้งาน ผลการทดสอบแสดงให้เห็นว่าแอปพลิเคชันสามารถทำงานได้อย่างมีประสิทธิภาพทั้งการตรวจจับแสง การรู้จำวัตถุ และการแยกแยะธนบัตร โดยผู้ใช้งานซึ่งเป็นผู้พิการทางสายตาได้ให้คะแนนความพึงพอใจในระดับสูง

S. Vaidya et al. [33] ได้นำเสนอการพัฒนาแอปพลิเคชันบนสมาร์ตโฟน และเว็บแอปพลิเคชัน เพื่อช่วยเหลือผู้พิการทางการมองเห็น โดยมุ่งเน้นการตรวจจับวัตถุแบบเรียลไทม์ และสื่อสารข้อมูลด้วยเสียงเพื่อให้ผู้ใช้งานสามารถรับรู้สิ่งรอบตัวได้ทันที แอปพลิเคชันจะนำภาพที่บันทึกได้มาผ่านโมเดล YOLOv3 และ Tiny YOLOv3 ซึ่งเป็นอัลกอริทึมหลักในการตรวจจับวัตถุแบบเรียลไทม์ ผลลัพธ์จากการตรวจจับจะผ่าน Text-to-Speech เพื่อแปลงข้อมูลดังกล่าวเป็นเสียง ทำให้ผู้พิการทางการมองเห็นสามารถรับรู้สิ่งรอบตัวได้อย่างต่อเนื่อง โดยไม่ต้องอาศัยการสัมผัสหน้าจอ ผลการทดลองพบว่าสามารถตรวจจับวัตถุโดยมีความแม่นยำสูงสุด 85.5% บนสมาร์ตโฟน และ 89% บนเว็บแอปพลิเคชัน อย่างไรก็ตาม ระบบยังมีข้อจำกัด เช่น วัตถุอยู่ไกลเกินไป หรืออยู่ในสภาพแสงน้อย แต่โดยรวมแล้วแอปพลิเคชันสามารถทำงานได้ดีโดยไม่ต้องใช้อินเทอร์เน็ต และช่วยเพิ่มความสะดวก ความมั่นใจ และความปลอดภัยแก่ผู้พิการทางการมองเห็นในการดำเนินชีวิตประจำวัน

ตารางที่ 2.5 สรุปงานวิจัยที่เกี่ยวข้องกับเรื่องการพัฒนาแอปพลิเคชันสำหรับช่วยเหลือผู้พิการทางสายตา

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Intelligent Eye: A Mobile Application for Assisting Blind People	พัฒนาแอปพลิเคชัน Android ที่มีหลายฟังก์ชัน ได้แก่ Light Detection, Color Detection, Object Recognition และ Banknote Recognition โดยแอปพลิเคชันใช้ OpenCV ในการประมวลผลภาพ (Image	แอปพลิเคชันสามารถตรวจจับ แสง สี วัตถุ และธนบัตรได้อย่างมีประสิทธิภาพ ผู้พิการทางการมองเห็นที่เข้าร่วมทดสอบมีความพึงพอใจสูง ช่วยเพิ่มความสะดวก ความมั่นใจ และความเป็นอิสระในการดำเนินชีวิตประจำวัน

ตารางที่ 2.5 สรุปงานวิจัยที่เกี่ยวข้องกับเรื่องการพัฒนาแอปพลิเคชันสำหรับช่วยเหลือผู้พิการทางสายตา (ต่อ)

งานวิจัย	กระบวนการที่ใช้	ผลลัพธ์
Intelligent Eye: A Mobile Application for Assisting Blind People (ต่อ)	Processing), CNNdroid สำหรับการทำงานของโครงข่ายประสาทเทียม (Neural Network) บนอุปกรณ์สมาร์ทโฟน และ CraftAR SDK เพื่อรองรับการรู้จำธนบัตร และทำงานร่วมกับ Text-to-Speech (TTS)	
Real-Time Object Detection for Visually Challenged People	พัฒนาแอปพลิเคชันบนสมาร์ทโฟน และเว็บแอปพลิเคชันเพื่อตรวจจับวัตถุแบบเรียลไทม์ โดยใช้โมเดล YOLOv3 และ Tiny YOLOv3 และทำงานร่วมกับ Text-to-Speech (TTS)	ความแม่นยำเฉลี่ยในการตรวจจับวัตถุบนสมาร์ทโฟน 85.5% และบนเว็บแอปพลิเคชัน 89% ระบบสามารถทำงานได้โดยไม่ต้องเชื่อมต่ออินเทอร์เน็ต นอกจากนี้ยังช่วยเพิ่มความปลอดภัย และความมั่นใจแก่ผู้พิการทางการมองเห็นในการดำเนินชีวิตประจำวัน

บทที่ 3

ระเบียบและวิธีการวิจัย

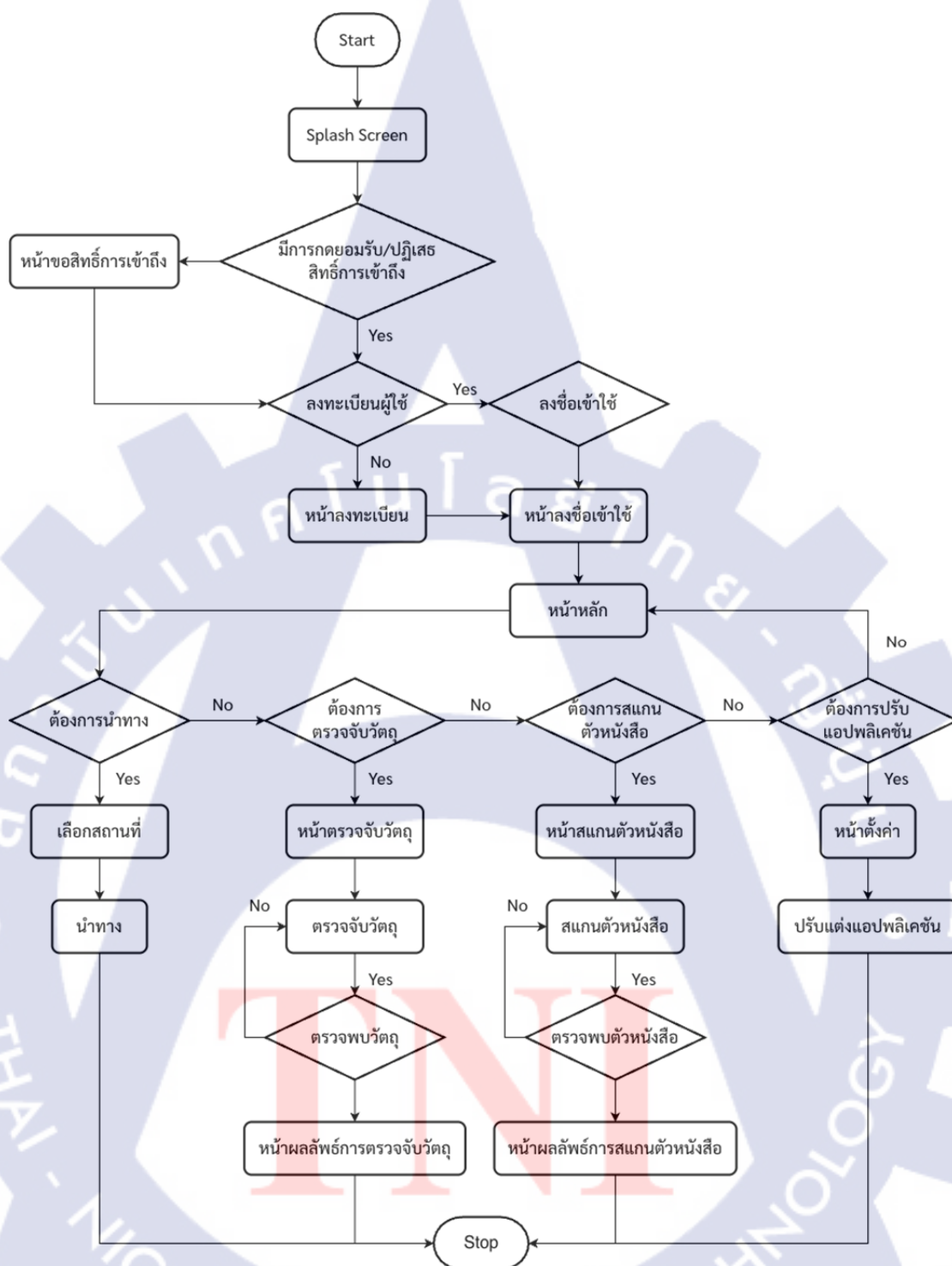
3.1 ภาพรวมของกระบวนการวิจัย

งานวิจัยนี้มุ่งเน้นการพัฒนาแอปพลิเคชัน Seeable ซึ่งเป็นแอปพลิเคชันสำหรับผู้พิการทางสายตา ภายในแอปพลิเคชันจะประกอบไปด้วยระบบนำทางภายในอาคารพร้อมระบบแจ้งเตือนสิ่งกีดขวาง การตรวจจับวัตถุ และการอ่านตัวหนังสือจากสภาพแวดล้อมเพื่อช่วยเพิ่มความปลอดภัย และความสามารถในการรับรู้สิ่งรอบตัวของผู้ใช้งาน

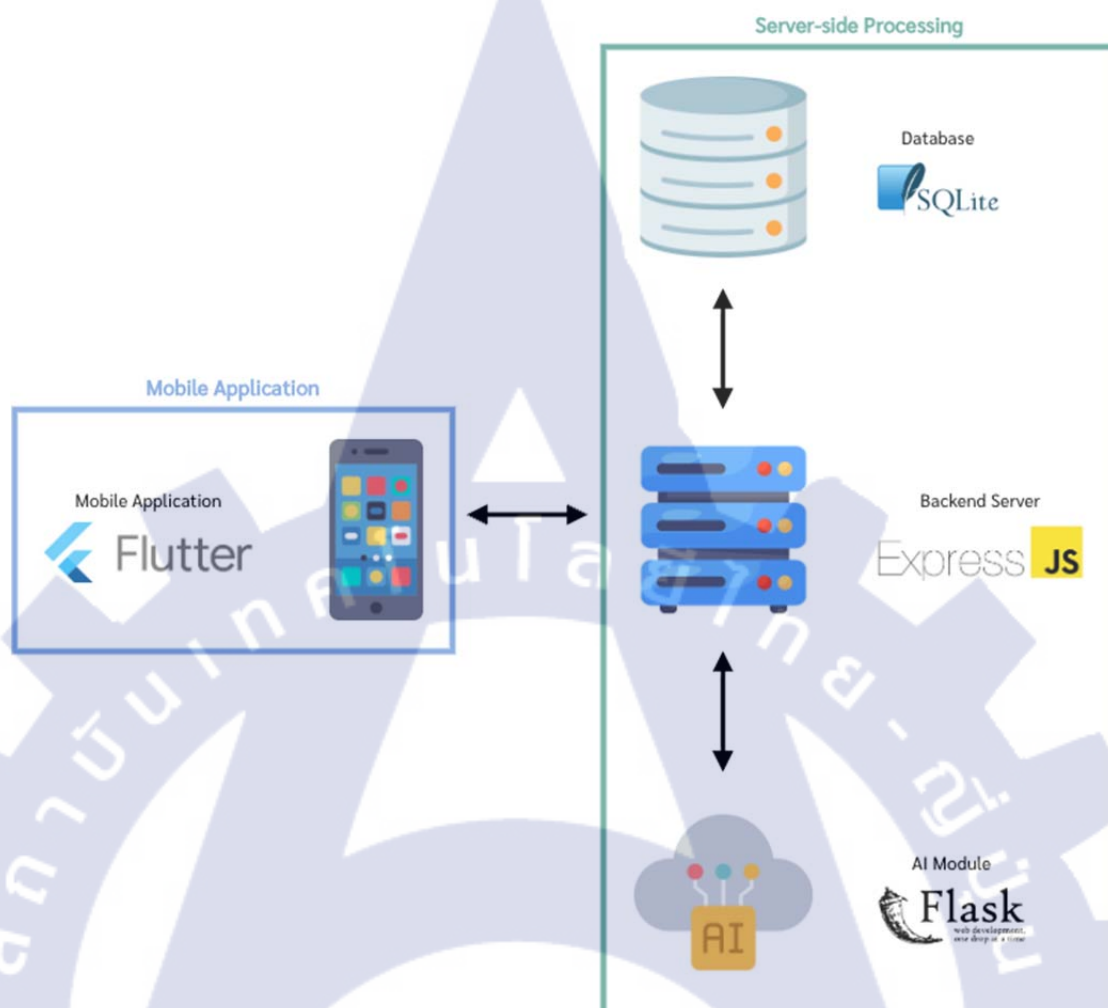
รูปที่ 3.1 แสดงภาพรวมของกระบวนการทำงานของแอปพลิเคชัน โดยเริ่มตั้งแต่ขั้นตอนการเข้าสู่ระบบ การตรวจสอบสิทธิ์ผู้ใช้งาน และเข้าสู่หน้าจอหลัก ซึ่งผู้ใช้สามารถเลือกใช้งานฟังก์ชันต่างๆ ได้แก่ การนำทาง การตรวจจับวัตถุ การสแกนตัวหนังสือ และการปรับแต่งการตั้งค่า ระบบจะประมวลผลข้อมูลจากกล้องแบบเรียลไทม์ เพื่อตรวจจับวัตถุหรือสิ่งกีดขวาง และแสดงผลลัพธ์หรือแจ้งเตือนกลับไปยังผู้ใช้งานในรูปแบบที่เหมาะสม เช่น เสียงหรือข้อความ

ในขณะที่รูปที่ 3.2 แสดงสถาปัตยกรรมของระบบ Seeable ซึ่งแบ่งออกเป็นสองส่วนหลัก ได้แก่ ส่วนของ Mobile Application ที่พัฒนาด้วย Flutter และส่วนของ Server-Side Processing ซึ่งประกอบด้วย Backend Server ที่พัฒนาด้วย Express.js ฐานข้อมูล SQLite และ AI Module ที่พัฒนาด้วย Flask โดยแอปพลิเคชันจะทำหน้าที่รับข้อมูลจากผู้ใช้งานและส่งไปยังเซิร์ฟเวอร์เพื่อประมวลผล เช่น การตรวจจับวัตถุหรือการวิเคราะห์ภาพ จากนั้นผลลัพธ์จะถูกส่งกลับมายังแอปพลิเคชันเพื่อแสดงผลแบบเรียลไทม์

จากทั้งสองภาพสามารถสรุปได้ว่า ระบบ Seeable ถูกออกแบบให้ทำงานร่วมกันระหว่างฝั่งผู้ใช้งานและฝั่งเซิร์ฟเวอร์อย่างมีประสิทธิภาพ โดยมีการจัดการลำดับการทำงานอย่างเป็นระบบ และรองรับการประมวลผลข้อมูลแบบเรียลไทม์ ซึ่งสอดคล้องกับวัตถุประสงค์ของงานวิจัยในการพัฒนาแอปพลิเคชันที่ช่วยเพิ่มความสะดวกและความปลอดภัยให้กับผู้พิการทางสายตา

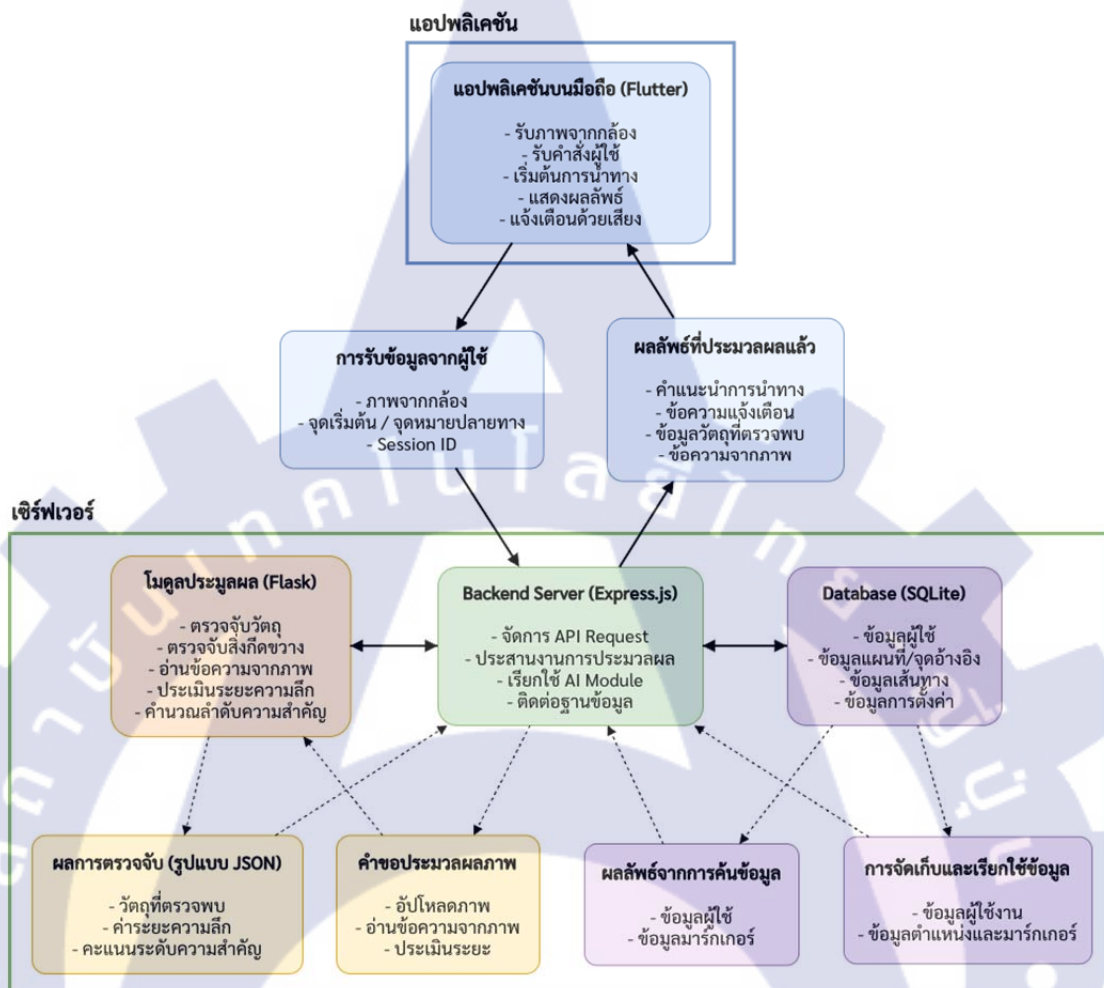


รูปที่ 3.1 ภาพรวมของกระบวนการทำงานของแอปพลิเคชัน



รูปที่ 3.2 สถาปัตยกรรมการทำงานของระบบ Seeable

ภาพรวมการทำงานของแอปพลิเคชัน Seeable



รูปที่ 3.3 แผนภาพภาพรวมการทำงานและสถาปัตยกรรมของระบบ Seeable

3.2 การออกแบบ UX/UI

การออกแบบแอปพลิเคชันสำหรับผู้พิการทางสายตา ไม่ควรจำกัดการให้เป็นเพียงแอปพลิเคชันที่รองรับเฉพาะกลุ่มผู้พิการทางสายตาเท่านั้น แต่ควรออกแบบโดยใช้หลักการ Inclusive Design ซึ่งเป็นแนวคิดที่ช่วยให้ทุกคนสามารถเข้าถึงและใช้งานแอปพลิเคชันได้อย่างสะดวกไม่ว่าจะเป็นผู้ใช้งานกลุ่มใดก็ตาม นอกเหนือจากใช้หลักการดังกล่าวแล้วก็ต้องมีการออกแบบการรองรับฟีเจอร์การช่วยเหลือการเข้าถึง หรือที่เรียกว่า Accessibility เพื่อให้ผู้ใช้งานกลุ่มผู้พิการทางสายตาสามารถใช้งานการอ่านหน้าจอได้ (Screen Reader)

นอกจากนี้แอปพลิเคชันยังออกแบบมาให้สามารถใช้งานการสั่งการด้วยเสียง ซึ่งช่วยเพิ่มความสะดวกในการใช้งาน โดยเฉพาะอย่างยิ่งผู้ที่ไม่สามารถสัมผัสหน้าจอได้โดยตรง การผสานเทคโนโลยีดังกล่าว จะช่วยให้แอปพลิเคชันสามารถรองรับการใช้งานได้อย่างครอบคลุมและตอบสนองต่อความต้องการของผู้ใช้ทุกกลุ่มได้อย่างมีประสิทธิภาพ

ภายในแอปพลิเคชันจะแบ่งออกเป็น 4 หน้าหลัก โดยแบ่งตามฟีเจอร์ของแอปพลิเคชัน คือ หน้านำทาง หน้าตรวจจับวัตถุ หน้าการอ่านตัวหนังสือ และหน้าตั้งค่าแอปพลิเคชัน

1. หน้านำทาง ต้องสามารถระบุตำแหน่งได้อย่างเรียลไทม์ และเสียงในการแจ้งเตือนตำแหน่งต้องทำการแจ้งเตือนเป็นช่วงๆ เช่น ตรงไปอีก 5 เมตร -> ตรงไปอีก 3 เมตร -> เลี้ยวขวา และสามารถแจ้งเตือนสิ่งกีดขวางได้ โดยจะแจ้งเตือนเป็นเสียง เช่น พบสิ่งกีดขวางเป็นเก้าอี้อยู่ทางขวามือ
2. หน้าตรวจจับวัตถุ จะมีการบอกผลลัพธ์เป็นสถานการณ์ตรงหน้า เช่น มีเก้าอี้อยู่ทางด้านขวามือของคุณประมาณ 2 เมตร
3. หน้าการอ่านตัวหนังสือ จะมีการแจ้งเตือนผู้ใช้งานว่าภาพมีความชัดมากพอหรือไม่ แสงของภาพจ้าเกินไปหรือมืดเกินไป หรือภาพไม่อยู่ในเฟรม เพื่อที่จะได้สแกนข้อความได้อย่างถูกต้องและแม่นยำ
4. หน้าตั้งค่า ผู้ใช้จะสามารถปรับแต่งส่วนต่างๆ ภายในแอปพลิเคชันได้ เช่น ภาษา, ความเร็วในการแจ้งเตือน, ธีม, การเปิดหรือปิดโหมดการสั่งการด้วยเสียง และการติดต่อกับพัฒนา

3.3 การพัฒนาแอปพลิเคชัน

แอปพลิเคชัน Seeable พัฒนาด้วย Flutter ซึ่งเป็นเฟรมเวิร์กข้ามแพลตฟอร์ม (Cross-Platform Framework) ที่สามารถรองรับทั้งระบบปฏิบัติการ iOS และ Android โดย Flutter รองรับการใช้งานฟีเจอร์ช่วยเหลือการเข้าถึง (Accessibility) ที่สามารถใช้งานร่วมกับการอ่านหน้าจอ (Screen Reader) และการสั่งการด้วยเสียง ซึ่งเหมาะสมกับผู้พิการทางสายตา นอกจากนี้ แอปพลิเคชันยังสามารถเชื่อมต่อกับระบบประมวลผลภายนอก เช่น การตรวจจับวัตถุ การวิเคราะห์สิ่งกีดขวาง การอ่านข้อความจากภาพ และการประมาณระยะ ซึ่งเป็นฟีเจอร์หลักของระบบ Seeable

โครงสร้างของระบบแบ่งออกเป็น 2 ส่วนหลัก คือ แอปพลิเคชันบนสมาร์ทโฟน (Mobile Application) และระบบประมวลผลฝั่งเซิร์ฟเวอร์ (Server-Side Processing) ซึ่งมีการทำงานตามแผนภาพภาพรวมของระบบ ดังรูปที่ 3.3 โดยมีการส่งข้อมูลจากผู้ใช้งานไปยังเซิร์ฟเวอร์เพื่อประมวลผล และส่งผลลัพธ์กลับมาแสดงผลแบบเรียลไทม์

3.3.1 แอปพลิเคชันบนสมาร์ทโฟน (Mobile Application)

แอปพลิเคชันเป็นส่วนที่ติดต่อกับผู้ใช้งานโดยตรง ทำหน้าที่รับข้อมูลจากผู้ใช้งาน เช่น ภาพจากกล้อง จุดเริ่มต้นและจุดหมายปลายทาง รวมถึงคำสั่งต่างๆ จากผู้ใช้งาน จากนั้นส่งข้อมูลไปยังเซิร์ฟเวอร์

เพื่อประมวลผล และรับผลลัพธ์กลับมาแสดงผลในรูปแบบที่เหมาะสม เช่น การแจ้งเตือนด้วยเสียง ข้อความ หรือคำแนะนำการนำทาง องค์ประกอบของแอปพลิเคชันประกอบด้วย

1. ส่วนติดต่อกับผู้ใช้ (User Interface) รองรับการแสดงผลข้อมูล เช่น คำแนะนำการนำทาง ข้อความแจ้งเตือน และข้อมูลวัตถุที่ตรวจพบ โดยออกแบบให้ใช้งานง่ายและเหมาะกับผู้ใช้พิการทางสายตา พร้อมทั้งรองรับการจัดการสถานะของแอปพลิเคชัน (State Management) และการขอสิทธิ์การเข้าถึงอุปกรณ์ เช่น กล้อง และไมโครโฟน

2. ส่วนรับข้อมูลจากผู้ใช้ (Input Handling) ทำหน้าที่รับข้อมูลจากกล้องและคำสั่งผู้ใช้ เช่น การกำหนดจุดเริ่มต้นและปลายทาง รวมถึงข้อมูล Session เพื่อใช้ในการประมวลผล

3. ส่วนแสดงผลลัพธ์ (Output Handling) แสดงผลลัพธ์ที่ได้จากการประมวลผลจากเซิร์ฟเวอร์ เช่น คำแนะนำการนำทาง ข้อความแจ้งเตือน ข้อมูลวัตถุ และข้อความจากภาพ พร้อมทั้งแจ้งเตือนผู้ใช้งานด้วยเสียง

4. ส่วนเชื่อมต่อกับเซิร์ฟเวอร์ (API Communication) ใช้สำหรับส่งข้อมูล เช่น ภาพหรือคำขอ ไปยังเซิร์ฟเวอร์ และรับผลลัพธ์กลับมา โดยใช้ไลบรารี Dio ในการจัดการการสื่อสารข้อมูล

นอกจากนี้แอปพลิเคชันยังมีการใช้ไลบรารีภายนอก เพื่อเสริมการทำงานให้ครอบคลุมและตอบสนองต่อความต้องการของผู้ใช้งาน ดังนี้

1. การสั่งการด้วยเสียง ใช้ Speech_to_text สำหรับการแปลงเสียงพูดเป็นข้อความ เพื่อนำมาใช้ควบคุมการทำงานของแอปพลิเคชัน และใช้ Flutter_tts สำหรับแปลงข้อความเป็นเสียงเพื่อตอบสนอง หรือเป็นการแจ้งเตือนสำหรับผู้ใช้งานแอปพลิเคชัน

2. การเปิดกล้อง ใช้ Camera สำหรับการนำภาพไปใช้ในการตรวจจับวัตถุ และการตรวจจับมาร์กเกอร์ความเป็นจริงเสริมในการระบุตำแหน่งผู้ใช้ และการนำทางภายในอาคาร

3. การนำเข้าภาพจากคลังรูปภายในอุปกรณ์สมาร์ทโฟน ใช้ Image_picker สำหรับการถ่ายภาพหรือนำภาพที่บันทึกไว้ไปตรวจจับวัตถุ และการตรวจจับอักขระด้วยแสง

4. การเชื่อมต่อกับเซิร์ฟเวอร์ ใช้ dio สำหรับการรับและส่งข้อมูล เพื่อนำไปประมวลผลและนำผลที่ได้มาแสดงหรือแจ้งเตือนภายในแอปพลิเคชัน

5. การเชื่อมต่อแดชบอร์ดสำหรับการตรวจสอบความผิดปกติของแอปพลิเคชัน ใช้ Firebase_crashlytics เพื่อที่นักพัฒนาจะได้รับอีเมลแจ้งเตือนทุกครั้งที่แอปพลิเคชันมีความผิดปกติเกิดขึ้น สามารถทำให้นักพัฒนาสามารถระบุสาเหตุและดำเนินการแก้ไขปัญหาได้อย่างทันท่วงที

6. การเชื่อมต่อกับแดชบอร์ดแสดงพฤติกรรมของผู้ใช้งาน ใช้ Firebase_analytics เพื่อเป็นการเก็บข้อมูลการใช้งานแอปพลิเคชันในฟีเจอร์ต่างๆ ข้อมูลที่ได้จะถูกนำมาวิเคราะห์เพื่อเข้าใจพฤติกรรม และความต้องการของผู้ใช้งาน เพื่อจะได้มีข้อมูลสำหรับการวางแผนพัฒนา และปรับปรุงประสิทธิภาพของแอปพลิเคชันให้ดีขึ้นในอนาคต

3.3.2 เซิร์ฟเวอร์ (Server-Side Processing)

ระบบฝั่งเซิร์ฟเวอร์ทำหน้าที่เป็นศูนย์กลางในการประมวลผลข้อมูลที่ส่งมาจากแอปพลิเคชันบนสมาร์ตโฟน โดยรับข้อมูลนำเข้า เช่น ภาพจากกล้อง จุดเริ่มต้น จุดหมายปลายทาง และข้อมูลคำขอจากผู้ใช้งาน แล้วดำเนินการประมวลผลผ่านส่วนประกอบต่างๆ ของระบบ ก่อนส่งผลลัพธ์กลับไปยังแอปพลิเคชันในรูปแบบที่พร้อมใช้งาน เช่น คำแนะนำการนำทาง ข้อความแจ้งเตือน ข้อมูลวัตถุที่ตรวจพบ และข้อความที่อ่านได้จากภาพ

การแยกงานประมวลผลออกจากสมาร์ตโฟนไปยังฝั่งเซิร์ฟเวอร์ช่วยลดภาระของอุปกรณ์พกพา โดยเฉพาะงานที่ใช้ทรัพยากรสูง เช่น การประมวลผลภาพ การตรวจจับวัตถุ การอ่านข้อความจากภาพ และการประมาณระยะ ซึ่งหากดำเนินการทั้งหมดบนอุปกรณ์พกพาโดยตรงอาจส่งผลกระทบต่อความเร็วในการตอบสนอง การใช้หน่วยความจำ และพลังงานของอุปกรณ์ ดังนั้น เซิร์ฟเวอร์จึงมีบทบาทสำคัญในการเพิ่มประสิทธิภาพและความยืดหยุ่นของระบบ

จากรูปที่ 3.3 แผนภาพภาพรวมการทำงานของระบบ สามารถแบ่งการทำงานของฝั่งเซิร์ฟเวอร์ออกเป็น 3 ส่วนหลัก ได้แก่ เซิร์ฟเวอร์ส่วนกลาง (Backend Server), โมดูลประมวลผลปัญญาประดิษฐ์ (AI Module) และฐานข้อมูล (Database) ซึ่งทำงานร่วมกันดังนี้

เซิร์ฟเวอร์ส่วนกลาง (Backend Server)

เซิร์ฟเวอร์ส่วนกลางที่พัฒนาด้วย Express.js ทำหน้าที่เป็นศูนย์กลางในการจัดการคำขอจากแอปพลิเคชัน และควบคุมลำดับการทำงานของระบบ โดยมีบทบาทสำคัญในการรับส่งข้อมูลประสานงานกับโมดูลประมวลผล และเชื่อมต่อกับฐานข้อมูล

เมื่อแอปพลิเคชันส่งข้อมูลเข้ามา เช่น ภาพจากกล้อง หรือข้อมูลสำหรับการนำทาง เซิร์ฟเวอร์จะทำการตรวจสอบข้อมูลเบื้องต้น และจัดประเภทของคำขอ จากนั้นจะดำเนินการตามลักษณะของงาน ดังนี้

1. การส่งต่อข้อมูลไปประมวลผล (Processing Dispatch) ตัวอย่างเช่นงานทางด้านการประมวลผลภาพ เช่น การตรวจจับวัตถุ การอ่านข้อความ เซิร์ฟเวอร์จะส่งข้อมูลไปยัง โมดูลประมวลผลปัญญาประดิษฐ์ (AI Module) เพื่อดำเนินการประมวลผล และรอรับผลลัพธ์กลับมา
2. การจัดการข้อมูล และการเชื่อมต่อฐานข้อมูล (Data Management) เช่น ข้อมูลผู้ใช้ ตำแหน่ง Marker หรือข้อมูลแผนที่ เซิร์ฟเวอร์จะทำการเชื่อมต่อกับ ฐานข้อมูล (Database) เพื่อดึงข้อมูล เปรียบเทียบข้อมูล หรือบันทึกข้อมูลตามที่กำหนด เพื่อใช้ประกอบการประมวลผลของระบบ
3. การคำนวณเส้นทางการนำทาง (Navigation Processing) เซิร์ฟเวอร์ทำหน้าที่ประมวลผลข้อมูลตำแหน่งโดยการรับข้อมูลตำแหน่งเริ่มต้น และจุดหมายปลายทาง เพื่อคำนวณเส้นทางการนำทางภายในอาคาร และนำผลลัพธ์ไปจัดรูปแบบเป็นคำแนะนำที่เหมาะสมสำหรับผู้ใช้งาน นอก

จากนี้ยังสามารถรับข้อมูลมาร์กเกอร์ที่ตรวจจับได้ เพื่อนำมาใช้ในการคำนวณตำแหน่งปัจจุบัน เส้นทาง การนำทาง และระยะทางคงเหลือ

4. การรวม และจัดรูปแบบผลลัพธ์ (Result Aggregation) หลังจากได้รับข้อมูลจาก โมดูล AI และฐานข้อมูล เซิร์ฟเวอร์จะทำการรวมผลลัพธ์ทั้งหมด เช่น ข้อมูลวัตถุ ค่าระยะความลึก และ ข้อมูลตำแหน่ง จากนั้นแปลงให้อยู่ในรูปแบบที่เข้าใจง่าย เช่น ข้อความแจ้งเตือน หรือคำแนะนำการนำทาง ก่อนส่งกลับไปยังแอปพลิเคชัน

โมดูลประมวลผลปัญญาประดิษฐ์ (AI Module)

โมดูลประมวลผลปัญญาประดิษฐ์ที่พัฒนาด้วย Flask ทำหน้าที่ในการประมวลผลข้อมูลเชิงลึกจากภาพ โดยรองรับการตรวจจับวัตถุ การตรวจจับสิ่งกีดขวาง การอ่านข้อความจากภาพ การประมาณระยะ และการคำนวณลำดับความสำคัญของวัตถุ เมื่อได้รับภาพจากเซิร์ฟเวอร์ส่วนกลาง โมดูลนี้จะประมวลผลตามประเภทคำขอ ดังนี้

1. การตรวจจับวัตถุ (Object Detection) ใช้โมเดล YOLOv8n ในการตรวจจับวัตถุภายในอาคาร เช่น ประตู โต๊ะ เก้าอี้ บันได คน และวัตถุอื่นๆ ที่อยู่ในชุดข้อมูลที่ใช้ฝึกสอน โมเดลจะส่งผลลัพธ์ในรูปแบบของตำแหน่งกรอบวัตถุ (Bounding Box) ประเภทของวัตถุ และค่าความเชื่อมั่น (Confidence Score)

2. การตรวจจับสิ่งกีดขวาง (Obstacle Detection) หลังจากตรวจพบวัตถุแล้ว ระบบจะวิเคราะห์ว่าวัตถุนั้นมีแนวโน้มเป็นสิ่งกีดขวางต่อผู้ใช้หรือไม่ โดยพิจารณาจากประเภทของวัตถุ ตำแหน่งของวัตถุในภาพ และข้อมูลระยะที่ได้จากการประมาณความลึก เพื่อใช้ประกอบการแจ้งเตือน

3. การประมาณค่าความลึก (Depth Estimation) ใช้โมเดล MiDaS ในการประมาณความลึกจากภาพ เพื่อช่วยระบุว่าวัตถุที่ตรวจพบอยู่ใกล้หรือไกลจากผู้ใช้ ข้อมูลนี้มีความสำคัญต่อการประเมินความเสี่ยงของสิ่งกีดขวาง และช่วยให้ระบบสามารถแจ้งเตือนได้เหมาะสมมากขึ้น

4. การลำดับความสำคัญ (Priority Framework) หลังจากได้ข้อมูลประเภทวัตถุและค่าระยะความลึก ระบบจะนำข้อมูลดังกล่าวมาคำนวณลำดับความสำคัญของวัตถุ (Priority Score) เพื่อพิจารณาว่าวัตถุใดควรถูกแจ้งเตือนก่อน เช่น วัตถุที่อยู่ใกล้ผู้ใช้และอยู่ในเส้นทางการเดินจะมีความสำคัญสูงกว่าวัตถุที่อยู่ไกลหรืออยู่นอกเส้นทาง

5. การอ่านข้อความจากภาพ (Optical Character Recognition) ใช้ Tesseract OCR สำหรับอ่านข้อความจากภาพ เช่น ป้ายบอกทาง ป้ายห้อง หรือเอกสารที่ผู้ใช้ต้องการอ่าน จากนั้นข้อความที่ได้จะถูกส่งกลับไปยังแอปพลิเคชันเพื่อนำไปแสดงผลหรืออ่านออกเสียงให้ผู้ใช้

ฐานข้อมูล (Database)

ฐานข้อมูลของระบบเลือกใช้ SQLite เนื่องจากมีขนาดเล็ก ติดตั้งง่าย และเหมาะสำหรับระบบที่ต้องการโครงสร้างฐานข้อมูลไม่ซับซ้อนมาก ข้อมูลที่จัดเก็บในฐานข้อมูลประกอบด้วยข้อมูลผู้ใช้งาน และข้อมูลมาร์กเกอร์ความเป็นจริงเสริมสำหรับประกอบการระบุตำแหน่งและการนำทาง

1. ข้อมูลผู้ใช้งาน ใช้สำหรับจัดเก็บข้อมูลพื้นฐานของผู้ใช้ ข้อมูลการเข้าสู่ระบบ หรือข้อมูลที่เกี่ยวข้องกับการใช้งานแอปพลิเคชัน

2. ข้อมูลมาร์กเกอร์ และจุดอ้างอิงเชิงตำแหน่ง โดยที่จะเก็บข้อมูลว่ามาร์กเกอร์นี้ มี id อะไร และเป็นจุดอ้างอิงของห้องใด เช่น {"marker_id": 1, "marker_name": "C301-F"} คือ มาร์กเกอร์ที่มี id เท่ากับ 1 ใช้เป็นจุดอ้างอิงของห้อง C301 ประตูด่าน

3.4 การพัฒนาระบบการลำดับความสำคัญ

การพัฒนาระบบการลำดับความสำคัญ (Priority Framework) มีเป้าหมายเพื่อยกระดับการทำงานของงานของการตรวจจับสิ่งกีดขวางให้สามารถจัดเรียงความสำคัญของวัตถุที่ตรวจพบตามลำดับความเสี่ยง และบริบทของสถานการณ์จริง ก่อนทำการแจ้งเตือนแก่ผู้ใช้งาน แนวทางดังกล่าวเป็นการลดการแจ้งเตือนที่ไม่จำเป็น และทำให้ผู้ใช้งานได้รับข้อมูลที่สำคัญ และเกี่ยวข้องมากที่สุดในแต่ละช่วงเวลา

โดยระบบประกอบด้วยองค์ประกอบหลัก 4 ส่วน ได้แก่ การตรวจจับวัตถุด้วยโมเดล YOLOv8n การประเมินค่าความลึกด้วยโมเดล MiDaS (Mixed Datasets for Monocular Depth Estimation) การคำนวณค่าน้ำหนักวัตถุ (Object Weight) และการประเมินตำแหน่งของวัตถุในภาพ (Direction Factor) โดยค่าที่ได้จากแต่ละองค์ประกอบจะถูกนำมาผสมผสานรวมกันเพื่อคำนวณค่าลำดับความสำคัญ (Priority Score)

3.4.1 การคำนวณค่าน้ำหนักวัตถุ (Object Weight)

การคำนวณค่าน้ำหนักวัตถุ (Object Weight) เป็นการคำนวณหาค่าที่ใช้สะท้อนถึงความสำคัญของวัตถุแต่ละประเภท โดยจะพิจารณาจาก 3 ปัจจัยหลัก ได้แก่ ความถี่ในการพบเจอในชุดข้อมูลฝึก (Frequency) ระดับความปลอดภัย (Safety Risk) และความสามารถในการหลบหลีก (Avoidability)

1. ความถี่ในการพบเจอในชุดข้อมูลฝึก (Frequency) เพื่อสะท้อนความแพร่หลายของวัตถุในสภาพแวดล้อมจริง ค่าดังกล่าวถูกปรับให้อยู่ในช่วงมาตรฐานด้วยกระบวนการ Normalization ดังสมการ

$$\text{freq_score} = 0.3 + 0.7 \times \frac{\text{count} - \text{min}}{\text{max} - \text{min}}$$

โดยที่

count คือ จำนวนตัวอย่างของคลาสนั้น

min และ max จำนวนตัวอย่างของคลาสนั้น

การกำหนดค่าเริ่มต้นที่ 0.3 เพื่อป้องกันไม่ให้วัตถุที่มีจำนวนข้อมูลน้อยมีค่าน้ำหนักต่ำจนเกินไป

2. ระดับความปลอดภัย (Safety Risk) เป็นการประเมินระดับความอันตรายของวัตถุต่อผู้ใช้งาน โดยแบ่งออกเป็น 3 ระดับ ได้แก่ ระดับสูง ระดับปานกลาง และระดับต่ำ ตัวอย่างเช่น บันไดและลิฟท์จัดอยู่ในกลุ่มความเสี่ยงสูง ส่วนโต๊ะและเก้าอี้จัดอยู่ในกลุ่มความเสี่ยงปานกลาง เป็นต้น ค่าความเสี่ยงถูกนำมาพหุคูณกับค่าความถี่ในการพบเจอในชุดข้อมูลฝึก (Frequency) ดังสมการ

$$\text{risk_score} = 0.5 \times \text{risk} + 0.4 \times \text{freq_score}$$

โดยให้น้ำหนักด้านความเสี่ยงมากกว่า เนื่องจากความปลอดภัยของผู้ใช้งานเป็นปัจจัยที่สำคัญที่สุด

3. ความสามารถในการหลบหลีก (Avoidability) หมายถึง ความง่ายในการหลบหลีกวัตถุ แบ่งออกเป็น 3 ระดับ ได้แก่ หลบยาก หลบได้ปานกลาง และหลบง่าย วัตถุขนาดใหญ่หรือวัตถุที่อยู่กับที่มักจัดอยู่ในกลุ่มที่หลบยาก

3.4.2 การคำนวณค่าระยะลึก (Depth Score)

การประเมินระยะลึกใช้โมเดล MiDaS ซึ่งเป็นโมเดลประเภท Monocular Depth Estimation สามารถประมาณค่าความลึกจากภาพเพียงภาพเดียว หลังจากประมวลผลจะได้แผนภาพความลึก (Depth Map) ซึ่งแต่ละพิกเซลมีค่าความลึกสัมพันธ์กับระยะของวัตถุ ในการคำนวณค่าระยะลึก (Depth Score) ใช้ค่าจาก 2 ตัวชี้วัด ได้แก่

D_c (Close-depth Percentile) คือ ค่าความลึกของพิกเซลที่ใกล้ที่สุดภายในภาพ

D_m (Median Depth) คือ ค่ากลางของความลึกภายในภาพ

โดยที่ค่าความลึกที่ต่ำหมายถึงวัตถุอยู่ใกล้ จึงทำการกลับค่าก่อนการคำนวณ โดยให้น้ำหนักกับค่าความลึกที่ใกล้ที่สุดมากกว่า เพื่อสะท้อนความเสี่ยงที่เกิดจากวัตถุที่อยู่ใกล้ผู้ใช้งานมากที่สุด

$$DS = 0.7(1 - D_c) + 0.3(1 - D_m)$$

3.4.3 การประเมินตำแหน่งในภาพ (Direction Factor)

ตำแหน่งของวัตถุในภาพมีผลต่อความเสี่ยงในการชน โดยจะแบ่งภาพออกเป็น 3 ส่วน ได้แก่ ด้านซ้าย กึ่งกลาง และด้านขวา หากวัตถุอยู่บริเวณกึ่งกลางภาพ ซึ่งมีแนวโน้มขวางเส้นทางเดินโดยตรง จะกำหนดค่า Direction Factor เท่ากับ 1 ส่วนวัตถุที่อยู่ทางด้านซ้ายหรือขวาจะกำหนดค่าเท่ากับ 0.8 เพื่อสะท้อนความเสี่ยงที่ลดลง

3.4.4 การคำนวณค่าลำดับความสำคัญ (Priority Score)

ค่าลำดับความสำคัญขั้นต้น (Raw Priority) คำนวณได้จากผลคูณของปัจจัยทั้งสาม คือ ค่าน้ำหนักวัตถุ (Object Weight: OW) ค่าระยะลึก (Depth Score : DS) และการประเมินตำแหน่งในภาพ (Direction Factor : DF) ดังสมการ

$$P_{raw} = DS \times OW \times DF$$

จากนั้นทำการปรับค่าให้อยู่ในช่วง [0, 1] ด้วยกระบวนการ Normalization เพื่อให้สามารถเปรียบเทียบวัตถุหลายรายการในภาพเดียวกันได้ โดยที่ถ้าค่าลำดับความสำคัญมีค่ามากกว่าหรือเท่ากับ 0.4 ระบบจะทำการแจ้งเตือนผู้ใช้งานทันที

$$p = \frac{P_{raw} - P_{min}}{P_{max} - P_{min}}$$

3.5 การพัฒนาระบบการนำทางในที่ร่มด้วยมาร์กเกอร์ความเป็นจริงเสริม

ในงานวิจัยนี้ใช้มาร์กเกอร์ความเป็นจริงเสริมประเภท ArUco Marker เป็นจุดอ้างอิงเชิงตำแหน่งภายในอาคารสำหรับการนำทางในที่ร่ม เนื่องจากมาร์กเกอร์มีลักษณะเฉพาะที่สามารถตรวจจับ และระบุรหัสมาร์กเกอร์แต่ละจุดได้อย่างชัดเจน จึงเหมาะสำหรับใช้เป็นจุดอ้างอิงในการระบุตำแหน่งของผู้ใช้งานภายในอาคาร

กระบวนการตรวจจับมาร์กเกอร์จะดำเนินการที่ฝั่งแอปพลิเคชันบนสมาร์ตโฟนที่พัฒนาด้วย Flutter ผ่านการประมวลผลของระบบปฏิบัติการ (Native Layer) เนื่องจาก Flutter ไม่สามารถเรียกใช้ความสามารถของไลบรารี OpenCV สำหรับการตรวจจับ ArUco Marker ได้โดยตรง จึงต้องมีการเชื่อมต่อระหว่าง Flutter และ Native เพื่อให้สามารถตรวจจับมาร์กเกอร์ความเป็นจริงเสริมประเภท ArUco ได้อย่างมีประสิทธิภาพ ในระบบปฏิบัติการ Android และ iOS

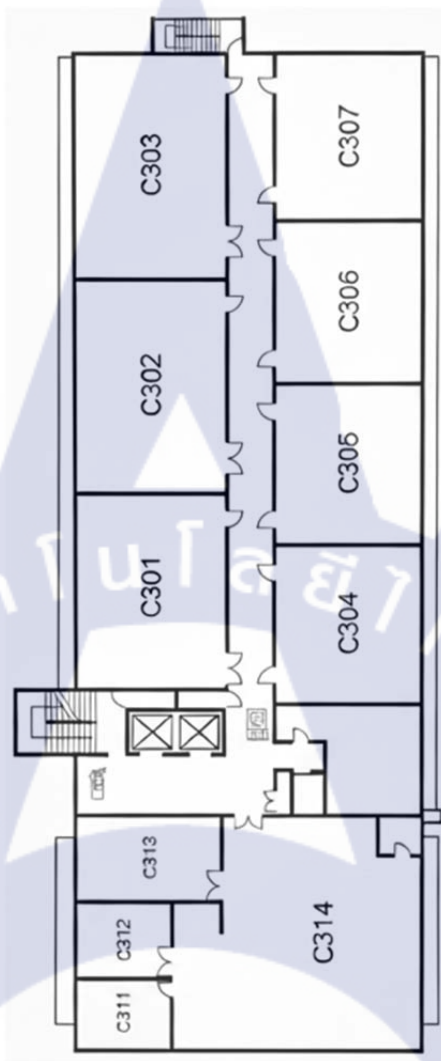
เมื่อแอปพลิเคชันตรวจจับพบมาร์กเกอร์ ระบบจะได้ค่า marker_id เพื่อนำไปเปรียบเทียบกับข้อมูลในฐานข้อมูลเพื่อระบุว่ามาร์กเกอร์ดังกล่าวอ้างอิงถึงตำแหน่งในภายในอาคาร จากนั้นแอปพลิเคชันจะส่งข้อมูลดังกล่าวไปยังระบบเซิร์ฟเวอร์ เพื่อใช้ในการระบุตำแหน่งปัจจุบันของผู้ใช้งาน คำนวณเส้นทางการนำทาง (Path Planning) และประเมินระยะทางคงเหลือจนถึงจุดหมาย

ในการทดลองนี้ ใช้ผังอาคาร C ชั้น 3 ของสถาบันเทคโนโลยีไทย-ญี่ปุ่นเป็นพื้นที่ต้นแบบในการพัฒนาระบบนำทาง ดังแสดงในภาพที่ 3.4 โดยผังดังกล่าวประกอบด้วยห้องหลักที่ใช้ในการนำทาง ได้แก่ C301-C303 ทางฝั่งซ้ายของโถงทางเดิน และ C304-C307 ทางฝั่งขวา รวมถึงห้อง C314 ที่เป็นส่วนแยกออกจากโถงหลัก ลักษณะของอาคารเป็นโถงทางเดินตรงที่มีห้องอยู่สองฝั่ง ทำให้สามารถกำหนดโครงสร้างการนำทางในรูปแบบเชิงเส้นได้อย่างชัดเจน

ผังอาคารนี้ถูกนำมาแปลงเป็นโครงสร้างเชิงตรรกะสำหรับการคำนวณเส้นทาง ในรูปแบบของโหนด (Node) เพื่อใช้เป็นจุดอ้างอิงในการนำทาง เช่น C301F, C301B, C304F, C305F เป็นต้น โดย F (Front) แทนประตูหน้าของห้อง และ B (Back) แทนประตูหลังของห้อง เพื่อให้สามารถจำลองการเคลื่อนที่ภายในอาคารได้อย่างละเอียด

นอกจากนี้ ยังมีการกำหนดความสัมพันธ์ระหว่างโหนด โดยแต่ละโหนดจะเชื่อมต่อกันตามลำดับของห้อง และทิศทางการเดิน เช่น การเชื่อมต่อระหว่างด้านหน้าและด้านหลังของห้องเดียวกัน การเชื่อมต่อระหว่างห้องที่อยู่ติดกันในแนวทางเดิน และการเชื่อมต่อระหว่างฝั่งซ้ายและฝั่งขวาของอาคาร ทั้งนี้ การกำหนดโครงสร้างดังกล่าวช่วยให้ระบบสามารถคำนวณเส้นทาง ระยะทาง และลำดับการเคลื่อนที่ของผู้ใช้งานได้อย่างเป็นระบบ

เพื่อให้การคำนวณระยะทางสอดคล้องกับสภาพแวดล้อมจริงของอาคาร ระบบยังมีการกำหนดค่าระยะเชิงกายภาพ เช่น ระยะความลึกของห้อง ระยะห่างระหว่างห้อง และระยะเชื่อมต่อระหว่างโถงหลักกับพื้นที่อื่น โดยค่าดังกล่าวถูกนำมาใช้ร่วมกับโครงสร้างโหนดในการคำนวณเส้นทางการนำทาง (Path Planning) และระยะทางคงเหลือ (Remaining Distance) อย่างมีประสิทธิภาพ



รูปที่ 3.4 ผังอาคาร C ชั้น 3 ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

3.5.1 การกำหนดตำแหน่งอ้างอิงในผังอาคาร

เพื่อให้สามารถคำนวณเส้นทางได้อย่างเป็นระบบ ผังอาคารถูกแปลงให้อยู่ในรูปแบบของจุดอ้างอิงเชิงตำแหน่ง (โหนด) โดยกำหนดให้แต่ละห้องมีอย่างน้อย 2 จุด คือ

1. F (Front) ใช้แทนตำแหน่งบริเวณหน้าห้องหรือจุดที่ใกล้แนวทางเดินหลัก
2. B (Back) ใช้แทนตำแหน่งอีกด้านหนึ่งของแนวห้อง เพื่อใช้สำหรับคำนวณระยะต่อเนื่องภายในแนวทางเดิน

จากโครงสร้างของระบบ ห้องถูกแบ่งออกเป็น 2 ผังหลัก ได้แก่

1. ผังซ้าย: C301, C302, C303
2. ผังขวา: C304, C305, C306, C307

นอกจากนี้ ยังมีการกำหนดค่าระยะทางที่ใช้ในการคำนวณเส้นทางให้สอดคล้องกับสภาพแวดล้อมจริงของอาคาร ได้แก่

1. ฝั่งซ้าย มีระยะลิคห้องเท่ากับ 8 เมตร
2. ฝั่งขวา มีระยะลิคห้องเท่ากับ 6 เมตร
3. ระยะระหว่างห้องที่อยู่ติดกันตามแนวทางเดินเท่ากับ 3 เมตร
4. ระยะเชื่อมจากโถงหลักไปยังห้อง C314 เท่ากับ 8 เมตร

ค่าระยะทางเหล่านี้ถูกกำหนดไว้ในระบบ เพื่อนำไปใช้ในการคำนวณระยะทางรวมและเส้นทางการนำทางให้มีความใกล้เคียงกับการใช้งานจริงมากที่สุด

3.5.2 กระบวนการคำนวณเส้นทางการนำทาง

เมื่อผู้ใช้งานเลือกจุดเริ่มต้นและจุดหมายปลายทาง ระบบจะส่งข้อมูลดังกล่าวไปยังเซิร์ฟเวอร์ และคำนวณระยะทาง โดยพิจารณาตามเงื่อนไขของตำแหน่งเริ่มต้น และตำแหน่งปลายทาง การคำนวณเส้นทางในงานวิจัยนี้สามารถจำแนกได้เป็น 5 กรณีหลัก ดังนี้

กรณีที่ 1 : จุดเริ่มต้นและจุดหมายเป็นจุดเดียวกัน ระบบจะกำหนดระยะทางเท่ากับ 0 เมตร เนื่องจากถือว่าผู้ใช้งานอยู่ที่ตำแหน่งเป้าหมายแล้ว

กรณีที่ 2 : อยู่ภายในห้องเดียวกัน แต่เป็นคนละด้าน (F/B) เช่น C301F ไป C301B ระบบจะกำหนดระยะทางเท่ากับค่าความลึกของห้องนั้น เช่น ห้องฝั่งซ้ายมีความลึก 8 เมตร ส่วนห้องฝั่งขวามีความลึก 6 เมตร

กรณีที่ 3 : จุดเริ่มต้นและจุดหมายอยู่ในฝั่งเดียวกัน เช่น C301F ไป C303B หรือ C304F ไป C307B ระบบจะคำนวณระยะตามลำดับของห้องในแนวทางเดิน โดยมีหลักการคือ

1. จาก F ไป B ภายในห้องเดียวกัน จะบวกระยะลิคของห้อง
 2. จาก B ของห้องหนึ่งไป F ของห้องถัดไป จะบวกระยะระหว่างห้องเท่ากับ 3 เมตร
- ระบบจะวนลำดับห้องทีละห้องจนถึงจุดหมายปลายทาง โดยคำนึงถึงทิศทางการเดิน

ตัวอย่าง จาก C301F ไป C303B

C301F -> C301B = 8 เมตร

C301B -> C302F = 3 เมตร

C302F -> C302B = 8 เมตร

C302B -> C303F = 3 เมตร

C303F -> C303B = 8 เมตร

รวมระยะทางทั้งหมด = 30 เมตร

กรณีที่ 4 : จุดเริ่มต้นและจุดหมายอยู่คนละฝั่ง เช่น การนำทางจากฝั่งซ้ายไปฝั่งขวา ระบบจะพิจารณาความสัมพันธ์ระหว่างโหนดโดยใช้ 2 กลไกหลัก ได้แก่ โหนดที่อยู่ตรงข้ามกัน (Opposite Node) และ โหนดที่ไม่ได้อยู่ตรงข้ามกันโดยตรง (Cross Offset)

1. โหนดที่อยู่ตรงข้ามกัน (Opposite Node) ระบบจะถือว่าระยะข้ามฝั่งเป็น 0 เมตร และปรับจุดเริ่มต้นไปยังโหนดฝั่งตรงข้ามทันที ก่อนดำเนินการคำนวณระยะทางต่อไปยังจุดหมายปลายทาง

ตัวอย่างคู่โหนดที่อยู่ตรงข้ามกัน ได้แก่

C301F <-> C304F

C301B <-> C305F

C303F <-> C306B

C303B <-> C307B

2. ไม่ได้อยู่ตรงข้ามกันโดยตรง (Cross Offset) ซึ่งเป็นกรณีที่ต้องข้ามฝั่งแต่มีระยะเพิ่มตามรูปแบบของผังอาคาร ได้แก่

C302F -> C305F = 3 เมตร

C302F -> C306F = 6 เมตร

C302F -> C306B = 12 เมตร

C302B -> C305F = 12 เมตร

กรณีที่ 5: เส้นทางที่เกี่ยวข้องกับห้อง C314 เนื่องจากห้อง C314 อยู่ในตำแหน่งที่แยกออกจากแนวห้องหลัก จึงถูกกำหนดเป็นกรณีพิเศษในการคำนวณระยะ หากจุดเริ่มต้นหรือจุดหมายเป็น C314 ระบบจะใช้ระยะเชื่อมคงที่เท่ากับ 8 เมตร เพื่อเชื่อมระหว่าง C314 กับจุดตั้งต้นของฝั่งซ้ายหรือฝั่งขวา ได้แก่ C301F หรือ C304F แล้วจึงคำนวณระยะต่อไปยังห้องปลายทางตามเงื่อนไขของฝั่งนั้น

3.5.3 การสร้างเส้นทางสำหรับติดตามการนำทาง

ในการนำทาง ระบบจะทำการคำนวณระยะทางรวม และสร้างลำดับโหนดของเส้นทางเพื่อใช้ติดตามสถานะของผู้ใช้งานระหว่างการเดินทาง โดยเส้นทางจะถูกจัดเก็บในรูปแบบลำดับของโหนดที่ผู้ใช้งานควรผ่าน เช่น [C301F, C301B, C302F, C302B, C303F]

ข้อมูลดังกล่าวจะถูกจัดเก็บภายใต้ session_id ร่วมกับข้อมูลจุดเริ่มต้น จุดหมายปลายทาง และระยะทางรวม เพื่อให้ระบบสามารถติดตามความคืบหน้าของผู้ใช้งาน และประเมินสถานะการนำทางได้อย่างต่อเนื่องตลอดกระบวนการ

3.5.4 การอัปเดตตำแหน่งระหว่างการเดินทาง

ระหว่างการใช้งาน แอปพลิเคชันจะสแกนหามาร์กเกอร์ความเป็นจริงเสริมประเภท ArUco Marker อย่างต่อเนื่อง เมื่อพบมาร์กเกอร์ใหม่ ระบบจะส่งชื่อโหนดที่ตรวจพบ แล้วนำมาเปรียบเทียบกับฐานข้อมูลเพื่อระบุตำแหน่งปัจจุบัน หลังจากได้ตำแหน่งปัจจุบัน ระบบจะตรวจสอบสถานะการนำทางได้ 4 กรณี ได้แก่

1. Arrived หมายถึง ผู้ใช้ถึงจุดหมายปลายทางแล้ว ระบบจะแจ้งเตือนผู้ใช้งานด้วยเสียงและข้อความ เช่น “คุณถึงจุดหมายปลายทางแล้ว” พร้อมหยุดการนำทาง และสิ้นสุดการนำทาง
2. On_Path หมายถึง ผู้ใช้ยังอยู่บนเส้นทางที่ระบบกำหนด ระบบจะคำนวณระยะทางคงเหลือจากตำแหน่งปัจจุบันไปยังจุดหมายปลายทาง และแจ้งเตือนเป็นระยะ เช่น “เหลือระยะทางอีก 5 เมตร ให้เดินตรงไป” หรือ “อีก 3 เมตรให้เลี้ยวขวา” เพื่อช่วยให้ผู้ใช้งานสามารถเดินตามเส้นทางได้อย่างถูกต้อง
3. OFF_PATH หมายถึง ผู้ใช้ออกนอกเส้นทางที่กำหนด ระบบจะแจ้งเตือนผู้ใช้งานทันที เช่น “คุณออกนอกเส้นทาง กรุณากลับเข้าสู่เส้นทางที่กำหนด” พร้อมแนะนำทิศทางเพื่อกลับเข้าสู่เส้นทางเดิม เช่น “ให้หันกลับและเดินตรงไป” หรือ “ให้เลี้ยวซ้ายเพื่อกลับเข้าสู่เส้นทาง”
4. Overshoot หมายถึง ผู้ใช้เดินเลยจุดหมายปลายทาง ระบบจะแจ้งเตือนผู้ใช้งานว่า “คุณเดินเลยจุดหมาย กรุณาเดินย้อนกลับ” และอาจจะระบุระยะทางย้อนกลับโดยประมาณ เช่น “ย้อนกลับประมาณ 2 เมตร” เพื่อช่วยให้ผู้ใช้งานสามารถกลับไปยังตำแหน่งเป้าหมายได้อย่างถูกต้อง

3.5.5 ลำดับขั้นตอนการทำงานของระบบนำทาง

กระบวนการทำงานของระบบนำทางด้วยมาร์กเกอร์ความเป็นจริงเสริมสามารถสรุปเป็นลำดับขั้นได้ดังนี้

1. ผู้ใช้งานทำการเลือกตำแหน่งเริ่มต้นและจุดหมายปลายทางจากรายการสถานที่ที่จัดเก็บอยู่ในฐานข้อมูล โดยข้อมูลดังกล่าวจะถูกส่งไปยังเซิร์ฟเวอร์เพื่อเริ่มกระบวนการนำทาง
2. เซิร์ฟเวอร์นำข้อมูลจุดเริ่มต้นและจุดหมายปลายทางมาแปลงเป็นโหนด (Node) และคำนวณเส้นทางนำทาง รวมถึงระยะทางรวม โดยอ้างอิงจากโครงสร้างผังอาคารในรูปแบบโหนดและค่าระยะเชิงกายภาพที่กำหนดไว้
3. ระบบจะสร้างรหัส session_id เพื่อใช้ติดตามสถานะการนำทางของผู้ใช้งาน พร้อมจัดเก็บข้อมูล เช่น จุดเริ่มต้น จุดหมายปลายทาง เส้นทาง (path) และระยะทางรวม
4. แอปพลิเคชันจะเริ่มการนำทาง และทำการสแกนหา ArUco Marker อย่างต่อเนื่องผ่านกล้องของสมาร์ทโฟน เพื่อใช้ในการตรวจจับตำแหน่งปัจจุบันของผู้ใช้งานแบบเรียลไทม์

5. เมื่อพบมาร์กเกอร์ใหม่ แอปพลิเคชันจะส่งข้อมูลมาร์กเกอร์ที่ตรวจพบไปยังเซิร์ฟเวอร์เพื่อประมวลผล และใช้ในการระบุตำแหน่งปัจจุบันของผู้ใช้งานโดยอ้างอิงจากฐานข้อมูล จากนั้นเซิร์ฟเวอร์จะเปรียบเทียบตำแหน่งปัจจุบันกับเส้นทางที่กำหนดไว้ เพื่อประเมินสถานะการนำทางของผู้ใช้งาน ได้แก่ อยู่บนเส้นทาง (On_Path) ออกนอกเส้นทาง (Off_Path) เดินเลยจุดหมาย (Overshoot) หรือถึงจุดหมายแล้ว (Arrived)

6. หากผู้ใช้อยู่บนเส้นทาง ระบบจะคำนวณระยะทางคงเหลือจากตำแหน่งปัจจุบันไปยังจุดหมายปลายทางใหม่อีกครั้ง ในกรณีที่ผู้ใช้ออกนอกเส้นทาง ระบบอาจทำการประเมินเส้นทางใหม่เพื่อให้ผู้ใช้กลับเข้าสู่เส้นทางที่ถูกต้อง

7. การแจ้งเตือนและแนะนำการนำทาง เช่น ระยะทางคงเหลือ ทิศทางการเดิน หรือข้อความแจ้งเตือน จะถูกส่งกลับไปยังแอปพลิเคชัน และแสดงผลในรูปแบบข้อความ พร้อมทั้งแปลงเป็นเสียงเพื่อแจ้งเตือนผู้ใช้งานแบบเรียลไทม์

3.6 การจัดเตรียมข้อมูล (Dataset) สำหรับการสร้างโมเดลในการตรวจจับสิ่งกีดขวาง

การจัดเตรียมชุดข้อมูลส่วนของการตรวจจับวัตถุ จะเน้นไปที่วัตถุที่อยู่ในอาคาร C ของสถาบันเทคโนโลยีไทย-ญี่ปุ่นเป็นหลัก ประกอบด้วยวัตถุ 14 ประเภท ได้แก่ ตู้ (Cabinet), เก้าอี้ (Chair), สิ่งของที่วางอยู่บริเวณทางเดิน (Clutter), ประตู (Door), ลิฟต์ (Elevator), ป้ายทางออกฉุกเฉิน (Exit Sign), ถังดับเพลิง (Fire Extinguisher), คน (Person), บันได (Stairs), โต๊ะ (Table), ป้ายสัญลักษณ์ห้องน้ำ (Toilet Sign), ถังขยะ (Trash Bin), ไวท์บอร์ด (Whiteboard), หน้าต่าง (Window) โดยภาพของวัตถุจะต้องเตรียมในหลายมุมมอง หลายสภาพแวดล้อม และหลายพื้นหลัง เพื่อความแม่นยำในการตรวจจับวัตถุ

3.7 การพัฒนาการตรวจจับวัตถุ (Object Detection)

หลังจากเตรียมชุดของข้อมูลที่เหมาะสมแล้ว ขั้นตอนต่อมา คือ การเลือกโมเดลที่มีขนาดสอดคล้องกับข้อจำกัดของสมาร์ทโฟนซึ่งมีทรัพยากรจำกัด ซึ่งงานวิจัยนี้เลือกใช้ YOLOv8n ซึ่งเป็นโมเดลขนาดเล็กที่มีความรวดเร็วในการประมวลผล แต่ยังคงมีความแม่นยำเพียงพอสำหรับการใช้งานจริง โมเดลดังกล่าวได้รับการฝึก (Training) ด้วยชุดข้อมูลที่เก็บจากสภาพแวดล้อมภายในอาคาร โดยครอบคลุมวัตถุที่ผู้พิการทางสายตาพบได้บ่อย ทำให้สามารถแยกแยะวัตถุที่เป็นสิ่งกีดขวางได้ดียิ่งขึ้น

3.8 การพัฒนาการตรวจจับอักขระด้วยแสง (Optical Character Recognition)

ในงานวิจัยนี้เลือกใช้ Tesseract OCR ซึ่งเป็น Open Source ที่ได้รับความนิยม รองรับการตรวจจับอักขระหลายภาษา รวมถึงภาษาไทย และภาษาอังกฤษ ซึ่งเป็นภาษาหลักที่ใช้ในงานวิจัยนี้ ก่อนจะนำรูปภาพเข้ามาประมวลผลด้วย OCR จะนำภาพมาผ่านกระบวนการประมวลผลภาพ (Image Processing) เพื่อเพิ่มความแม่นยำ โดยการปรับความคมชัด และความสว่าง (Image Enhancement) การแปลงภาพเป็นสีเทา (Grayscale) การกรองสัญญาณรบกวน (Noise Reduction) และการทำ Thresholding เพื่อลดผลกระทบจากสภาพแสง และพื้นหลังที่ซับซ้อน เมื่อภาพที่มีคุณภาพเหมาะสมแล้วจึงนำไปประมวลผลด้วย Tesseract OCR เพื่อแปลงข้อความในภาพเป็นข้อมูลดิจิทัล โดยผลลัพธ์ดังกล่าวจะถูกส่งต่อไปยังแอปพลิเคชัน และอ่านออกเสียงให้ผู้ใช้งานผ่านระบบ Text-to-Speech (TTS)

3.9 การพัฒนาการสั่งการด้วยเสียง (Speech Recognition)

การพัฒนาการสั่งการด้วยเสียงจะเลือกใช้ Library ของ Flutter ที่ชื่อว่า `speech_to_text` โดยจะมีการสร้างชุดคำถามและคำตอบที่เกี่ยวข้องกับการใช้งานแอปพลิเคชัน เพื่อให้ผู้ใช้งานสามารถสั่งการแอปพลิเคชันผ่านคำพูดได้ เช่น “นำทางไปที่ห้องเรียน A302” ระบบจะรับเสียงผ่านทางไมโครโฟนของมือถือและแปลงเสียงพูดเป็นข้อความโดยใช้เทคโนโลยีแปลงเสียงเป็นข้อความ (Speech-to-Text) จากนั้นก็นำไปประมวลผลเพื่อตอบสนองต่อคำสั่ง นอกจากนี้มีการเลือกใช้ `flutter_tts` ซึ่งเป็นเทคโนโลยีการแปลงข้อความเป็นเสียง (Text-To-Speech) ในการแปลงผลลัพธ์เป็นเสียงแจ้งเตือนผู้ใช้งาน

3.10 การทดสอบและประเมินผล

การทดสอบและการประเมินผลการทำงานของระบบในงานวิจัยนี้มุ่งเน้นการประเมินทั้งความถูกต้อง และประสิทธิภาพของระบบนำทางในที่ร่ม และระบบแจ้งเตือนสิ่งกีดขวาง โดยแบ่งการทดสอบออกเป็น 2 ส่วนหลัก ได้แก่ การทดสอบทางเทคนิค และการทดสอบการใช้งานของในสถานการณ์จำลอง

3.10.1 การทดสอบทางเทคนิค

การทดสอบทางเทคนิคจะเป็นการประเมินประสิทธิภาพของระบบในเชิงตัวเลข โดยครอบคลุมทั้งความแม่นยำ ความรวดเร็ว และความเสถียรของระบบ โดยมีรายละเอียดดังนี้

1. การตรวจสอบความแม่นยำของระบบนำทางในที่ร่ม โดยทำการเปรียบเทียบตำแหน่งที่ระบบระบุได้จากมาร์กเกอร์ความเป็นจริงเสริมกับตำแหน่งจริงภายในอาคาร เพื่อประเมินความคลาดเคลื่อนของการระบุตำแหน่ง

2. การประเมินความแม่นยำของการตรวจจับวัตถุ และมาร์กเกอร์ประเมินผลการตรวจจับวัตถุ มาร์กเกอร์ และข้อความจากภาพในสภาพแวดล้อมที่แตกต่างกัน เช่น แสงน้อย แสงมาก หรือมีวัตถุซ้อนกัน โดยใช้ตัวชี้วัด เช่น ความแม่นยำโดยรวม (Accuracy), ความแม่นยำในการตรวจจับเฉพาะสิ่งที่ควรตรวจพบ (Precision) และความสามารถในการตรวจจับสิ่งที่ควรตรวจพบ (Recall)

3. การประเมินความแม่นยำของการแจ้งเตือนสิ่งกีดขวาง ในการประเมินส่วนนี้จะมีการจัดเตรียมชุดข้อมูลภาพ และกำหนดค่ามาตรฐาน (Ground Truth) โดยให้ผู้ประเมินทำการตีกรอบวัตถุ (Bounding Box) และระบุวัตถุที่ควรได้รับการแจ้งเตือนจากนั้นนำผลลัพธ์ของระบบมาเปรียบเทียบกับค่ามาตรฐาน โดยพิจารณาทั้งความถูกต้องของการตรวจจับวัตถุ และความถูกต้องของการเลือกวัตถุที่ควรแจ้งเตือน

4. การประเมินความล่าช้าในการทำงานของระบบ (Latency Evaluation) วัดระยะเวลาที่ใช้ในแต่ละขั้นตอนของระบบ ได้แก่ เวลาการส่งข้อมูล เวลาประมวลผลของโมเดล และเวลาการตอบสนองของระบบ เพื่อประเมินความเหมาะสมของระบบสำหรับการใช้งานแบบเรียลไทม์

5. การทดสอบระบบสั่งการและแจ้งผลด้วยเสียง เป็นการประเมินความถูกต้องของการแปลงเสียงเป็นข้อความ (Speech-to-Text) และความชัดเจนของการแจ้งเตือนด้วยเสียง (Text-to-Speech) เพื่อให้มั่นใจว่าผู้ใช้งานสามารถรับรู้ข้อมูลได้อย่างถูกต้อง

6. การประเมินความถูกต้องของเส้นทางการนำทางด้วยการเปรียบเทียบกับการเดินทางจริง ในการประเมินส่วนนี้จะให้ผู้ทดลองทำการเดินทางจากจุดเริ่มต้นไปยังจุดหมายปลายทางภายในอาคารด้วยตนเอง เพื่อใช้เป็นเส้นทางอ้างอิง จากนั้นนำเส้นทางที่ระบบคำนวณได้มาเปรียบเทียบกับเส้นทางจริง การประเมินพิจารณาจากความสอดคล้องของเส้นทาง ระยะทางรวมที่แตกต่างกัน และความสามารถของระบบในการนำทางผู้ใช้งานไปยังจุดหมายได้อย่างถูกต้อง การประเมินในลักษณะนี้ช่วยสะท้อนความเหมาะสมของระบบในการใช้งานจริง และความถูกต้องของอัลกอริทึมการนำทาง

3.10.2 การทดสอบการทำงานของระบบในสถานการณ์จำลองการใช้งานจริง

เนื่องจากข้อจำกัดด้านเวลา และขอบเขตของงานวิจัย การทดสอบในส่วนนี้มุ่งเน้นการประเมินความสามารถของระบบในการทำงานตามฟังก์ชันที่ออกแบบไว้ ภายใต้สถานการณ์จำลองการใช้งานจริง โดยมีรายละเอียดดังนี้

1. การทดสอบการนำทางภายในอาคาร จะทำการทดสอบการนำทางจากตำแหน่งเริ่มต้นไปยังจุดหมายปลายทาง เพื่อประเมินความถูกต้องของเส้นทางที่ระบบคำนวณ และความสามารถในการนำผู้ไปยังจุดหมายได้อย่างถูกต้อง

2. การทดสอบการอัปเดตตำแหน่งระหว่างการเคลื่อนที่ จะทำการประเมินความสามารถของระบบในการอัปเดตตำแหน่งผู้ใช้งานเมื่อมีการตรวจพบมาร์กเกอร์ใหม่ ว่าสามารถปรับปรุงข้อมูลตำแหน่งและระยะทางคงเหลือได้อย่างต่อเนื่อง

3. การทดสอบการแจ้งเตือนสิ่งกีดขวาง จะทำการประเมินความสามารถของระบบในการตรวจจับและแจ้งเตือนสิ่งกีดขวางในเส้นทางการเดิน โดยพิจารณาทั้งความถูกต้องของการแจ้งเตือนและความเหมาะสมของวัตถุที่ถูกเลือกมาแจ้งเตือน

4. การทดสอบฟังก์ชันการปรับแต่งการใช้งานของระบบ เป็นการทดสอบความสามารถของระบบในการปรับแต่งการใช้งาน เช่น การปรับความเร็วของเสียงแจ้งเตือน และการตั้งค่าภาษาของแอปพลิเคชัน เพื่อประเมินความยืดหยุ่นของระบบในการรองรับผู้ใช้งานที่หลากหลาย

ผลจากการทดสอบทั้งหมดถูกนำมาใช้ในการประเมินความพร้อมของระบบสำหรับการนำไปใช้งานจริง โดยพิจารณาทั้งด้านความแม่นยำ ประสิทธิภาพ และความเหมาะสมในการใช้งาน นอกจากนี้ ยังใช้เป็นข้อมูลในการวิเคราะห์ข้อจำกัดของระบบ และเป็นแนวทางสำหรับการพัฒนาและปรับปรุงระบบในอนาคต



TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

บทที่ 4

ผลการวิจัย

บทนี้นำเสนอผลการพัฒนา และทดสอบระบบ Seeable ซึ่งเป็นแอปพลิเคชันนำทางในที่ร่ม และระบบแจ้งเตือนสิ่งกีดขวางสำหรับผู้พิการทางสายตา โดยการนำเสนอผลลัพธ์แบ่งตามองค์ประกอบหลักของระบบ ได้แก่ แอปพลิเคชันบนสมาร์ทโฟน ระบบฝังเซิร์ฟเวอร์ และระบบประมวลผลภาพและปัญญาประดิษฐ์ เพื่อสะท้อนทั้งผลการทำงานของแต่ละส่วนและผลการทำงานร่วมกันของระบบโดยรวม

4.1 ผลการพัฒนาแอปพลิเคชันบนสมาร์ทโฟน (Flutter)

4.1.1 ผลการทำงานโดยรวมของแอปพลิเคชัน Seeable

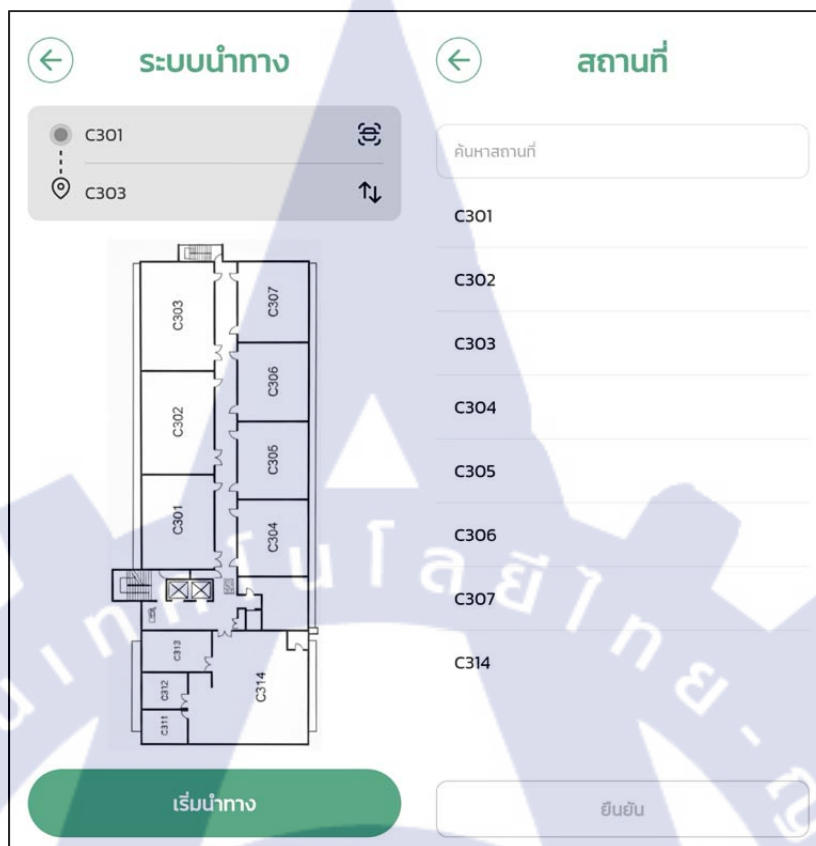
แอปพลิเคชัน Seeable ถูกพัฒนาด้วยเฟรมเวิร์ค Flutter ซึ่งรองรับการทำงานแบบข้ามแพลตฟอร์ม (Cross-Platform) โดยสามารถใช้งานได้ทั้งบนระบบปฏิบัติการ iOS และ Android ตามขอบเขตที่กำหนดไว้ในงานวิจัย

จากการทดสอบการทำงานโดยรวม พบว่าแอปพลิเคชันสามารถเชื่อมต่อกับระบบฝังเซิร์ฟเวอร์ (Express.js) และโมดูลประมวลผลภาพและปัญญาประดิษฐ์ (Flask) ได้อย่างถูกต้อง ผ่านการสื่อสารแบบ RESTful API โดยสามารถส่งข้อมูลภาพ รับผลลัพธ์ในรูปแบบ JSON และนำข้อมูลดังกล่าวมาแสดงผลในรูปแบบข้อความและเสียง (Text-to-Speech) ได้อย่างครบถ้วน

ระบบสามารถทำงานได้อย่างต่อเนื่อง ไม่มีการค้างของแอปพลิเคชัน และรองรับการสลับใช้งานหลายฟังก์ชัน เช่น การตรวจจับวัตถุ การแจ้งเตือนสิ่งกีดขวาง การนำทาง และการอ่านข้อความจากภาพ

4.1.2 ผลการแจ้งเตือนสิ่งกีดขวาง และนำทางภายในอาคาร

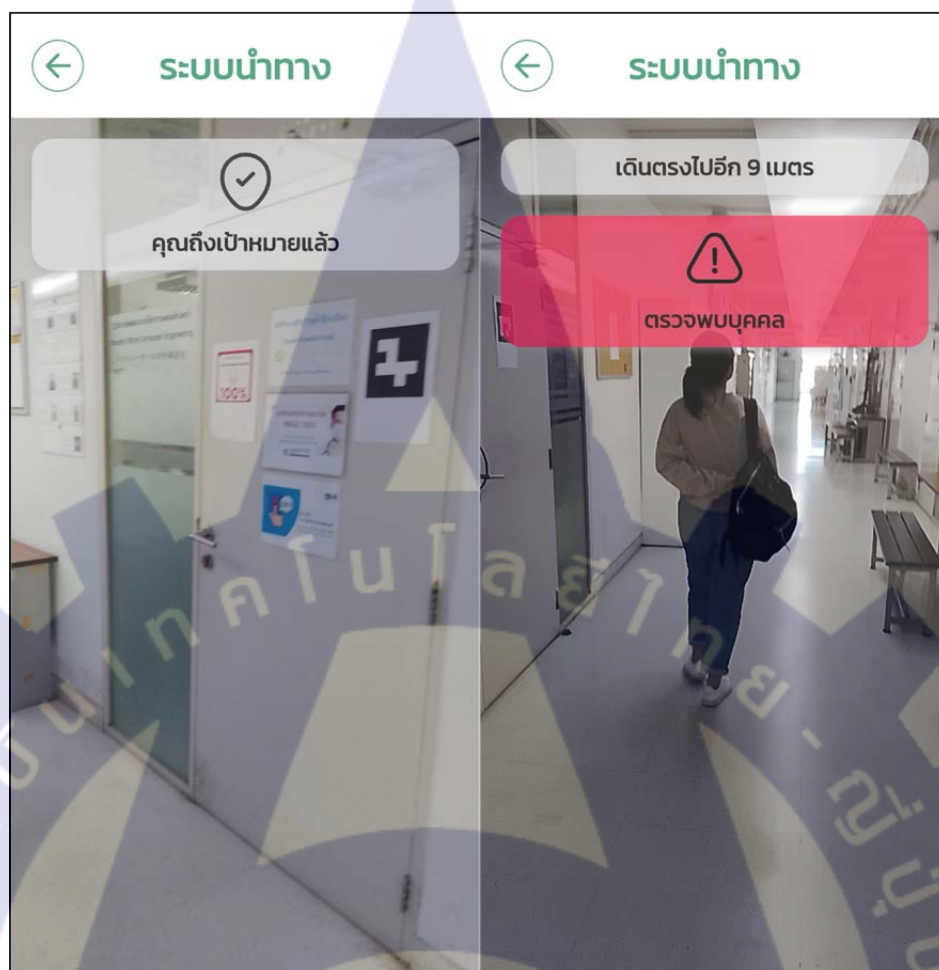
ฟังก์ชันการนำทางถูกออกแบบให้ทำงานร่วมกับระบบการแจ้งเตือนสิ่งกีดขวาง เพื่อเพิ่มความปลอดภัยในการเคลื่อนที่ภายในอาคาร โดยใช้มาร์กเกอร์ความเป็นจริงเสริมเป็นจุดอ้างอิงเชิงตำแหน่ง และการนำทางแบบโหนด (Node-Based Navigation) เพื่อนำทางผู้ใช้งานจากจุดเริ่มต้นไปยังจุดหมายปลายทางได้อย่างถูกต้อง



รูปที่ 4.1 ตัวอย่างหน้าจอการเลือกสถานที่ในการนำทาง

กระบวนการทำงานเริ่มจากผู้ใช้งานเลือกจุดเริ่มต้น และจุดหมายปลายทางเพื่อให้ระบบทำการคำนวณเส้นทาง ดังตัวอย่างในรูปที่ 4.1 จากนั้นระบบจะทำการสแกนมาร์กเกอร์เพื่อระบุตำแหน่งปัจจุบันของผู้ใช้งาน และนำไปคำนวณเส้นทาง พร้อมทั้งคำนวณระยะทางคงเหลืออย่างต่อเนื่องระหว่างการเดินทาง หากระบบตรวจพบว่าผู้ใช้งานเคลื่อนที่ออกนอกเส้นทางที่กำหนด หรือเดินเลยเป้าหมาย จะมีการแจ้งเตือน และคำนวณเส้นทางใหม่ เพื่อให้ผู้ใช้งานกลับเข้าสู่เส้นทางหลักได้อย่างถูกต้อง

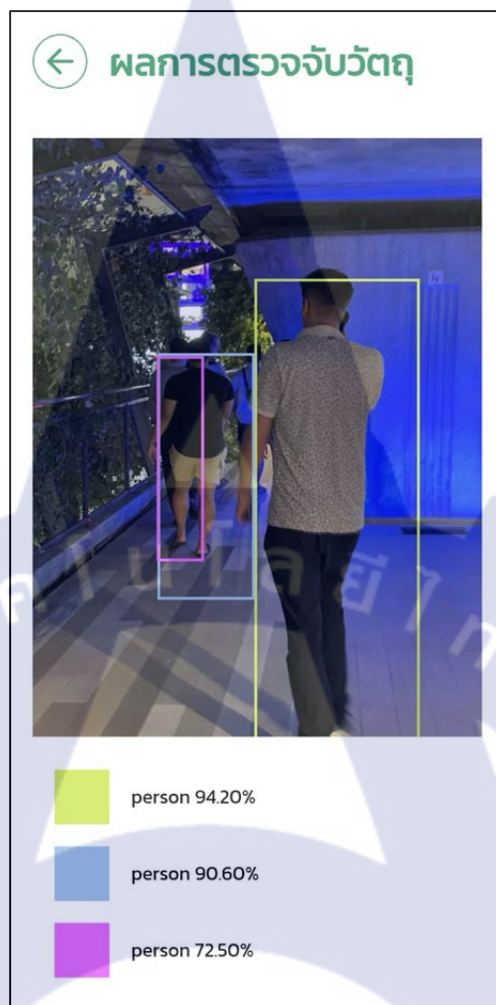
ในส่วนของการแจ้งเตือนสิ่งกีดขวาง ระบบผสานผลการตรวจจับวัตถุ ร่วมกับการประมาณค่าความลึก และคำนวณค่าลำดับความสำคัญ เพื่อเลือกแจ้งเตือนเฉพาะวัตถุที่อยู่ในแนวการเดินและมีความเสี่ยงสูง โดยระบุทิศทาง เช่น ด้านหน้า ด้านซ้าย หรือด้านขวา และแจ้งเตือนผ่านเสียงแบบเรียลไทม์จากการทดลองเดินจริงภายในอาคาร C ชั้น 3 ดังตัวอย่างในรูปที่ 4.2 พบว่า ระบบสามารถนำผู้ใช้งานไปยังจุดหมายได้ถูกต้อง อัปเดตตำแหน่งได้อย่างต่อเนื่อง และลดการแจ้งเตือนที่ซ้ำซ้อน ทำให้การนำทาง มีความปลอดภัยและเหมาะสมต่อการใช้งานของผู้พิการทางสายตา



รูปที่ 4.2 ตัวอย่างหน้าจอขณะนำทาง และแจ้งเตือนสิ่งกีดขวาง

4.1.3 ผลการทำงานตรวจจับวัตถุ

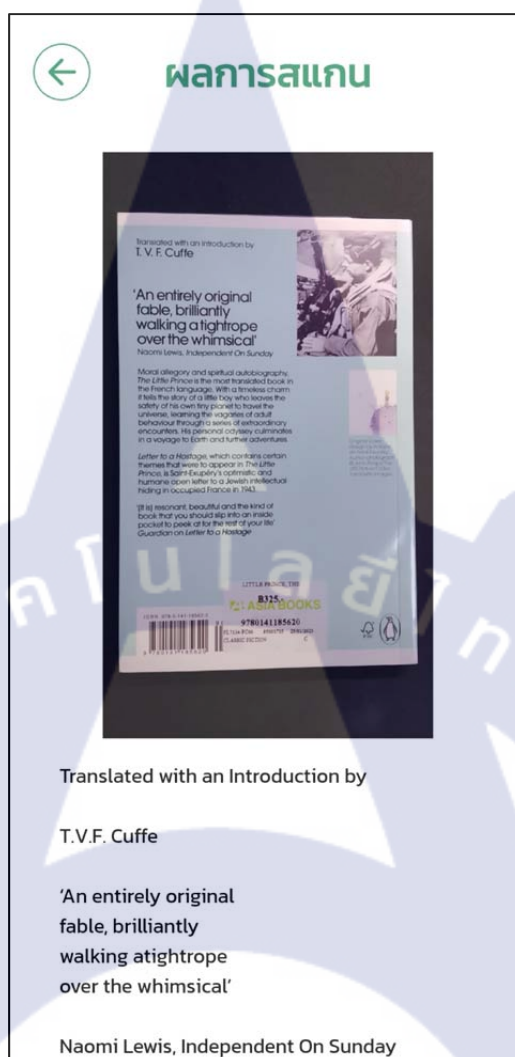
ฟังก์ชันการตรวจจับวัตถุทำงานโดยการรับภาพจากกล้องของรับภาพจากกล้องหรือคลังรูปภาพของสมาร์ทโฟนไปยังโมดูลการตรวจจับวัตถุ และรับข้อมูลของวัตถุที่ตรวจพบพร้อมตำแหน่งของวัตถุในภาพมาแสดงยังหน้าจอ และแจ้งเตือนผู้ใช้งานผ่านเสียง ดังตัวอย่างในรูปที่ 4.3



รูปที่ 4.3 ตัวอย่างหน้าจอการตรวจจับวัตถุ

4.1.4 ผลการตรวจจับอักขระด้วยแสง

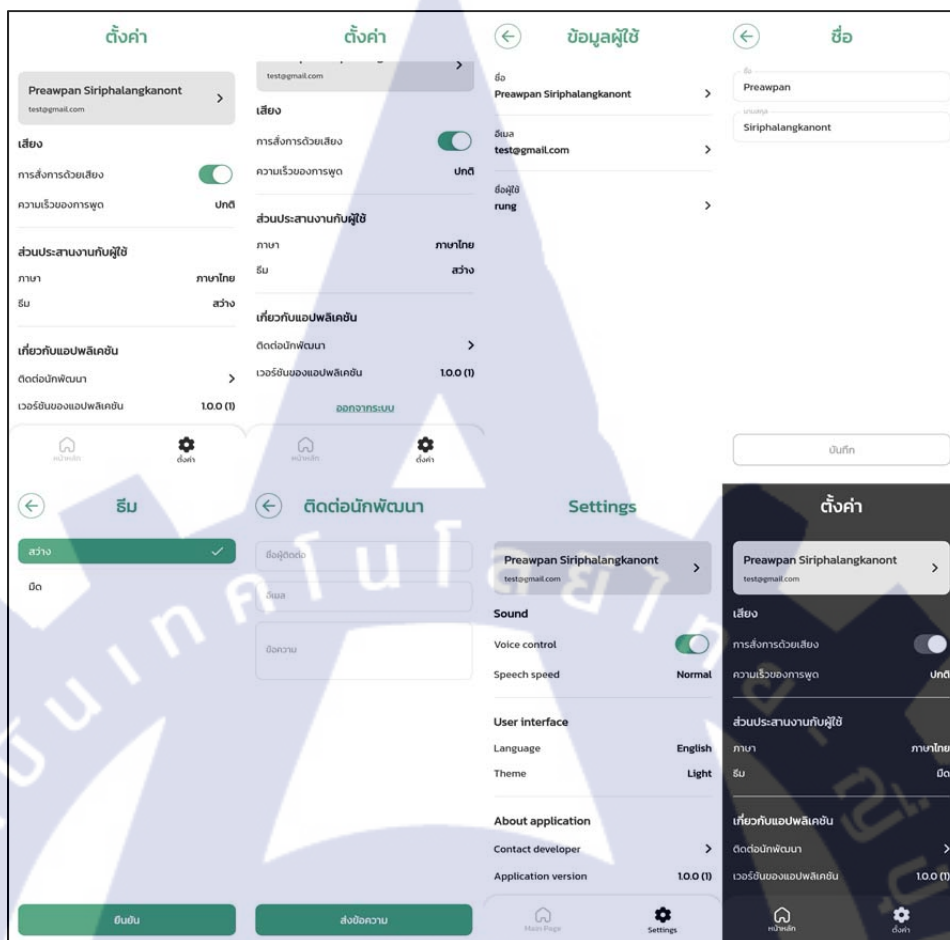
ฟังก์ชันการตรวจจับอักขระด้วยแสงทำงานโดยการรับภาพจากกล้องหรือคลังรูปภาพของสมาร์ทโฟนไปยังโมดูลการตรวจจับอักขระด้วยแสง และรับข้อความที่ตรวจพบกลับมาแสดงที่หน้าจอ และแปลงเป็นเสียงผ่านระบบ Text-to-Speech ดังตัวอย่างในรูปที่ 4.4



รูปที่ 4.4 ตัวอย่างหน้าจอการสแกนข้อความ

4.1.5 หน้าจอการตั้งค่า

แอปพลิเคชันมีหน้าจอสำหรับการตั้งค่าเพื่อให้ผู้ใช้งานสามารถปรับแต่งรูปแบบการใช้งานได้ เช่น การเลือกภาษา ความเร็วของเสียงในการแจ้งเตือน ธีม การจัดการบัญชีผู้ใช้งาน และการติดต่อนักพัฒนา ดังตัวอย่างในรูปที่ 4.5



รูปที่ 4.5 ตัวอย่างหน้าจอการตั้งค่า และการปรับแต่งแอปพลิเคชัน

4.2 ผลการทดสอบระบบฝั่งเซิร์ฟเวอร์ (Express.js)

4.2.1 ผลการรับ-ส่งข้อมูล และการประสานงานของระบบ

จากการทดสอบการทำงานของระบบฝั่งเซิร์ฟเวอร์พบว่าสามารถรองรับการสื่อสารผ่านโปรโตคอล HTTP ได้อย่างถูกต้อง โดยใช้ทั้ง Post, Get, Put และ Delete ตามประเภทของงาน และลักษณะของข้อมูลที่ร้องขอ (Request) เพื่อช่วยให้โครงสร้าง API มีความชัดเจน เป็นมาตรฐาน และง่ายต่อการบำรุงรักษาในอนาคต โดยแต่ละคำสั่งมีหน้าที่แตกต่างกันอย่างชัดเจน เช่น

Post ใช้สำหรับการส่งข้อมูลที่ต้องมีการประมวลผลหรือสร้างข้อมูลใหม่ เช่น การส่งภาพเพื่อไปทำการตรวจจับวัตถุ การตรวจจับอักขระ หรือการเริ่มต้น Session การนำทาง

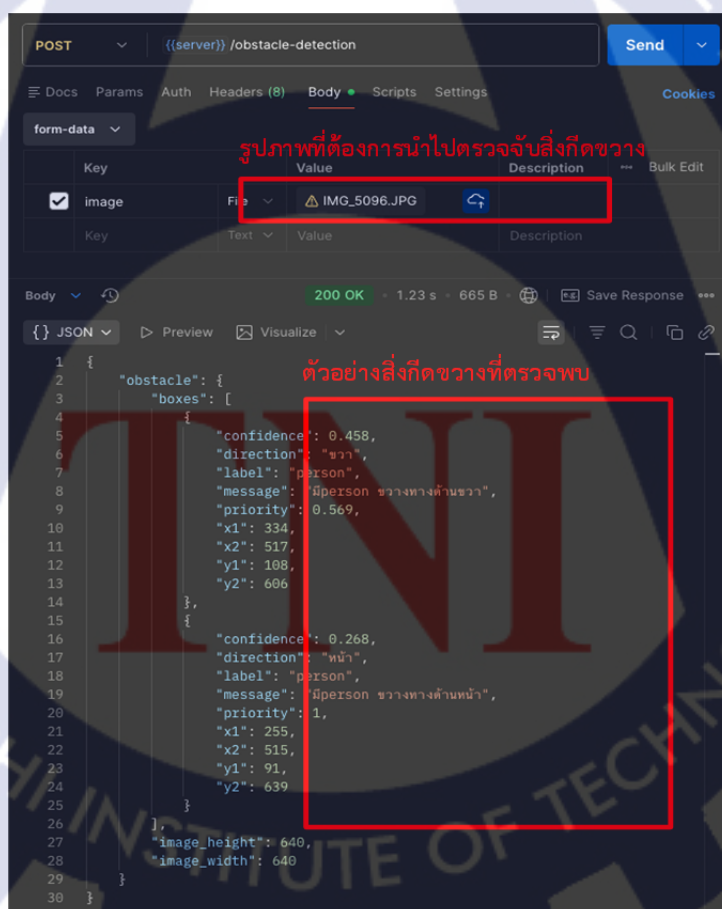
Get ใช้สำหรับการดึงข้อมูลจากระบบฐานข้อมูล เช่น การดึงข้อมูลสถานที่จากระบบ

Put ใช้สำหรับการปรับปรุงหรืออัปเดตข้อมูลในฐานข้อมูล เช่น การอัปเดตผู้ใช้

Delete ใช้สำหรับการลบข้อมูลภายในฐานข้อมูล เช่น การลบข้อมูลผู้ใช้

เพื่อแสดงให้เห็นถึงการทำงานของระบบฝั่งเซิร์ฟเวอร์อย่างเป็นรูปธรรม จึงได้นำเสนอตัวอย่างการทดสอบการรับส่งข้อมูลบางส่วนผ่านโปรแกรม Postman ซึ่งใช้ในการจำลองการส่งคำร้องขอ (Request) ไปยังแต่ละ Endpoint ของระบบ และตรวจสอบผลลัพธ์ที่ตอบกลับ (Response) จากเซิร์ฟเวอร์ ทั้งในรูปแบบ JSON และรหัสสถานะ (HTTP Status Code) ตัวอย่างการทดสอบมีดังนี้

1. การทดสอบการตรวจจับสิ่งกีดขวาง ซึ่งเป็นการส่งรูปภาพไปยังระบบเพื่อทำการตรวจสอบว่ามีสิ่งกีดขวางอยู่ในระยะที่ควรจะต้องแจ้งเตือนแก่ผู้ใช้งาน โดยการทดสอบผ่าน โปรแกรม Postman ด้วยการส่งคำร้องขอแบบ HTTP Post ไปยัง Endpoint/Obstacle-Detection ดังภาพที่ 4.6 เมื่อระบบได้รับภาพแล้ว จะทำการประมวลผลด้วยโมเดลการตรวจจับวัตถุ โมเดลการประเมินค่าความลึก พร้อมทั้งการคำนวณค่าลำดับความสำคัญ และตอบกลับผลลัพธ์มาในรูปแบบ JSON ระบบจะแสดงชื่อสิ่งกีดขวาง ค่าความเชื่อมั่น ตำแหน่งของสิ่งกีดขวางภายในภาพ ค่าลำดับความสำคัญ และตำแหน่งของวัตถุในภาพ (Bounding Box Coordinates) เพื่อนำมาแสดงผลในแอปพลิเคชันในรูปแบบข้อความ และเสียงแจ้งเตือน



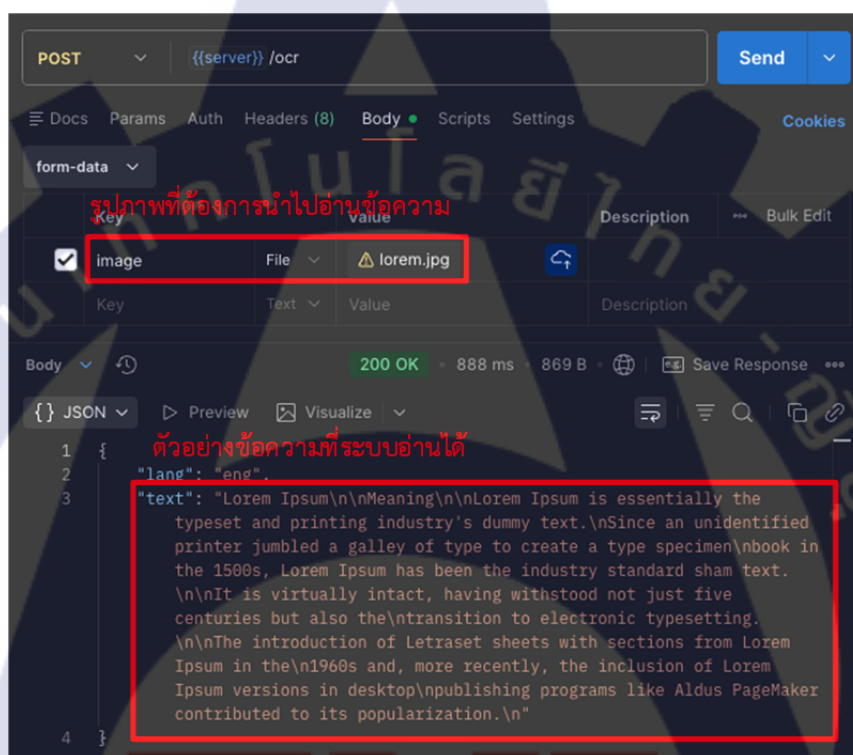
รูปที่ 4.6 ตัวอย่างการส่งรูปภาพเพื่อการตรวจจับสิ่งกีดขวางผ่านโปรแกรม Postman

2. การทดสอบการตรวจจับวัตถุ ซึ่งเป็นการส่งรูปภาพไปยังระบบเพื่อทำการตรวจสอบว่ามีวัตถุใดปรากฏอยู่ในภาพ โดยการทดสอบผ่านโปรแกรม Postman ด้วยการส่งคำร้องขอแบบ HTTP Post ไปยัง Endpoint/Object-Detection ดังรูปที่ 4.7 เมื่อระบบได้รับภาพแล้ว จะทำการประมวลผลด้วยโมเดลการตรวจจับวัตถุ และตอบกลับผลลัพธ์ในรูปแบบ JSON หากพบวัตถุที่ตรงกับวัตถุที่ใช้ในการฝึกโมเดล ระบบจะแสดงชื่อวัตถุ ค่าความเชื่อมั่น และตำแหน่งของวัตถุในภาพ เพื่อนำมาแสดงผลในแอปพลิเคชันในรูปแบบข้อความ และเสียงแจ้งเตือน



รูปที่ 4.7 ตัวอย่างการส่งรูปภาพเพื่อการตรวจจับวัตถุผ่านโปรแกรม Postman

3. การทดสอบการตรวจจับอักขระด้วยแสง ซึ่งเป็นการส่งรูปภาพไปยังระบบเพื่อทำการอ่านข้อความที่ปรากฏอยู่ภายในภาพ โดยการทดสอบผ่านโปรแกรม Postman ด้วยการส่งคำร้องขอแบบ HTTP Post ไปยัง Endpoint/ocr ดังรูปที่ 4.8 เมื่อระบบได้รับภาพแล้ว จะทำการประมวลผลด้วย Pytesseract เพื่อแยกข้อความออกจากภาพ และตอบกลับผลลัพธ์ในรูปแบบ JSON หากพบข้อความระบบจะทำการส่งข้อความที่อ่านได้มา เพื่อนำมาแสดงผลในแอปพลิเคชันในรูปแบบตัวอักษร และเสียงแจ้งเตือน

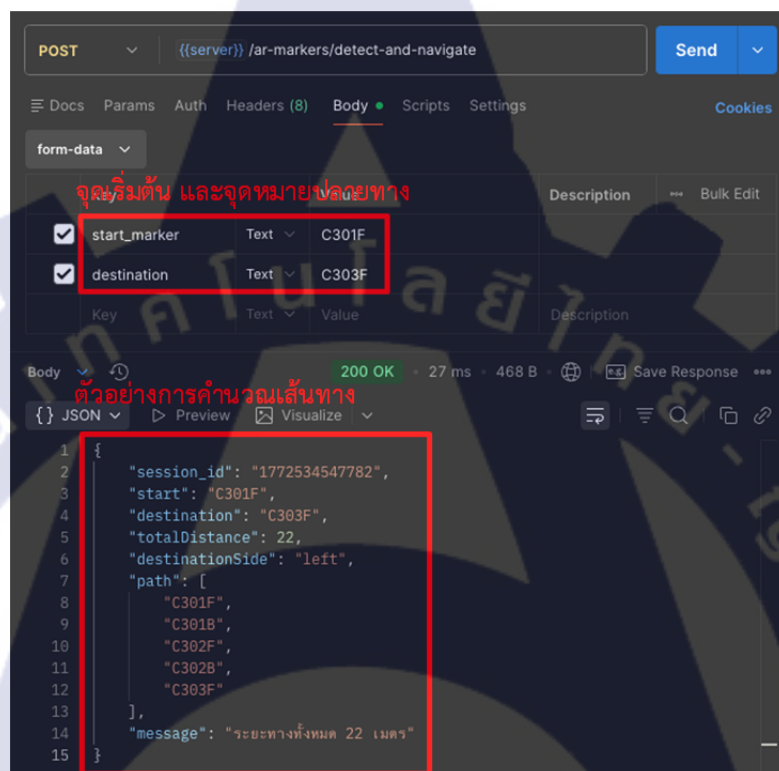


รูปที่ 4.8 ตัวอย่างการส่งรูปภาพเพื่อการตรวจจับอักขระด้วยแสงผ่านโปรแกรม Postman

4.2.2 ผลการจัดการเซสชันการนำทาง

ระบบฝั่งเซิร์ฟเวอร์มีการจัดการเซสชันการนำทางผ่านการกำหนด `session_id` ให้กับผู้ใช้งานแต่ละรายเมื่อเริ่มต้นการนำทาง เพื่อใช้เป็นรหัสอ้างอิงเฉพาะ (Unique Identifier) สำหรับติดตามสถานการณ์เดินทางของผู้ใช้งานแต่ละคนอย่างเป็นอิสระ `session_id` ถูกออกแบบมาให้ไม่ซ้ำกัน จึงช่วยป้องกันความสับสนของข้อมูล และลดความเสี่ยงของการปะปนกันของสถานะการนำทางระหว่างผู้ใช้งานหลายรายที่ใช้งานระบบพร้อมกัน

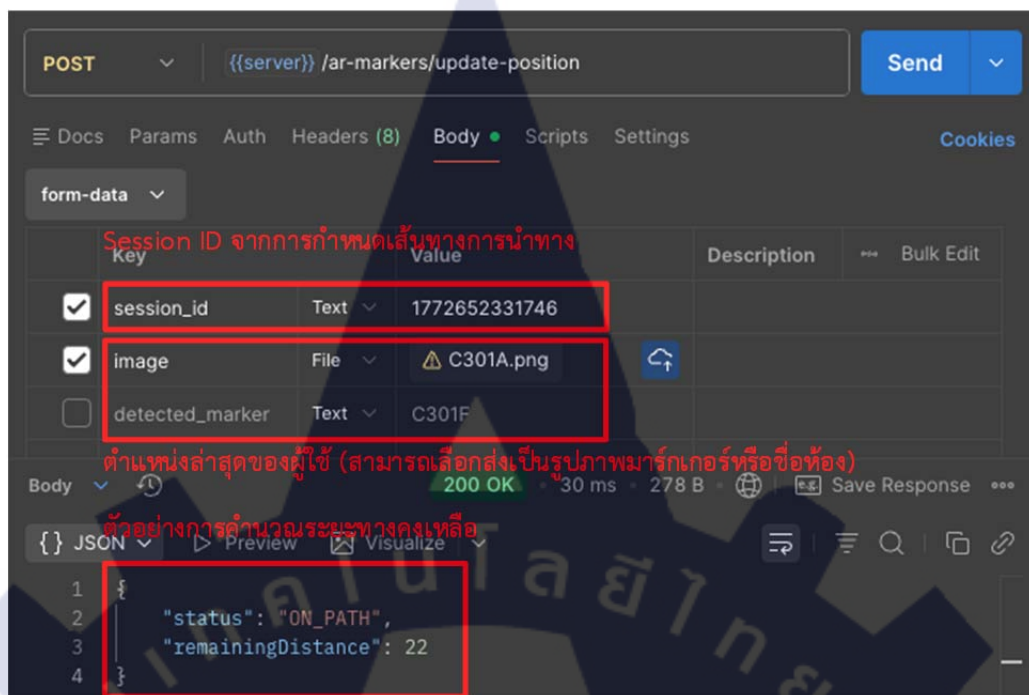
เมื่อผู้ใช้งานเริ่มต้นการนำทาง ระบบจะทำการเรียก API โดยการส่งคำร้องขอแบบ HTTP Post ไปยัง Endpoint/ar-markers/detect-and-navigate เพื่อทำการกำหนด session_id และเส้นทางการเดิน โดยจะส่งข้อมูลในรูปแบบ Multipart/form-data ประกอบด้วยจุดเริ่มต้น (สามารถส่งได้ทั้งแบบรูปภาพหรือชื่อมาร์กเกอร์) และจุดหมายปลายทาง



รูปที่ 4.9 ตัวอย่างผลลัพธ์การกำหนด session_id และเส้นทางผ่านโปรแกรม Postman

จากรูปที่ 4.9 แสดงตัวอย่างผลลัพธ์ที่ได้รับกลับมาในรูปแบบ JSON ซึ่งประกอบด้วยข้อมูลสำคัญ ได้แก่ session_id จุดเริ่มต้น จุดหมายปลายทาง ลำดับเส้นทาง ระยะทางทั้งหมด ทิศทางของจุดหมายปลายทาง และคำแจ้งเตือนผู้ใช้งาน ซึ่ง session_id จะเป็นข้อมูลสำคัญในการนำไปใช้ในการอัปเดตตำแหน่งของผู้ใช้งานต่อไป

หลังจากเริ่มต้นการนำทางแล้ว ระบบจะทำการเรียก API โดยการส่งคำร้องขอแบบ HTTP Post ไปยัง Endpoint/ar-markers/update-position เพื่อทำการอัปเดตตำแหน่งปัจจุบันของผู้ใช้งาน และเป็นการคำนวณระยะทางคงเหลือ รวมถึงการแจ้งเตือนในกรณีที่ผู้ใช้งานเดินออกนอกเส้นทางหรือเดินเลยเป้าหมายอีกด้วย ซึ่งข้อมูลสำคัญที่จะทำการส่งไปจะถูกส่งในรูปแบบ Multipart/form-data ประกอบด้วย session_id จากการนำทาง และรูปมาร์กเกอร์ที่ตรวจพบ



รูปที่ 4.10 ตัวอย่างผลลัพธ์การอัปเดตตำแหน่งของผู้ใช้งานผ่านโปรแกรม Postman

```
1 {
2   "status": "ON_PATH",
3   "remainingDistance": 14
4 }
```

```
1 {
2   "status": "ARRIVED",
3   "remainingDistance": 0
4 }
```

```
1 {
2   "status": "OFF_PATH",
3   "message": "คุณออกนอกเส้นทาง"
4 }
```

```
1 {
2   "status": "OVERSHOOT",
3   "message": "คุณเดินเลยจุดหมาย กรุณากลับหลัง"
4 }
```

รูปที่ 4.11 ตัวอย่างผลลัพธ์การอัปเดตตำแหน่งของผู้ใช้งาน

จากรูปที่ 4.10 แสดงตัวอย่างผลลัพธ์ที่ได้รับกลับมาในรูปแบบ JSON ซึ่งประกอบด้วยข้อมูลสำคัญ ได้แก่ ระยะทางคงเหลือ สถานะปัจจุบันของการนำทาง ข้อความแจ้งเตือนผู้ใช้งาน ซึ่งผลลัพธ์จะถูกแบ่งออกมาเป็น 4 แบบด้วยกัน คือ ถึงเป้าหมาย กำลังเดินทางตามเส้นทางที่กำหนดไว้ (on_path) เดินทางนอกเส้นทาง (off_path) และการเดินเลยเป้าหมาย (Overshoot) ดังรูปที่ 4.11

จากการทดลองเดินจริงภายในอาคาร C ชั้น 3 สถาบันเทคโนโลยีไทย-ญี่ปุ่น พบว่าระบบสามารถสร้างเส้นทาง และอัปเดตตำแหน่งของผู้ใช้งานได้ทันทีที่มีการตรวจพบมาร์กเกอร์ใหม่ ระยะทางคงเหลือถูกคำนวณใหม่อย่างถูกต้องตามลำดับโหนด ระบบสามารถตรวจจับกรณีที่ผู้ใช้งานเดินทางนอกเส้นทางหรือเดินเลยเป้าหมายได้อย่างถูกต้อง เมื่อถึงเป้าหมายก็สามารถส่งข้อความยืนยัน และสิ้นสุดเซสชันได้

4.2.3 ความเสถียร และความต่อเนื่องในการทำงาน

เพื่อทำการประเมินความเสถียรของระบบฝั่งเซิร์ฟเวอร์ ได้ดำเนินการทดสอบแบบ ต่อเนื่อง (Continuous Testing) โดยจำลองสถานการณ์การใช้งานจริงในรูปแบบที่มีการส่งคำร้องขอเข้าสู่ระบบเป็นจำนวนมากติดต่อกัน ทั้งในลักษณะการส่งภาพต่อเนื่อง และการสลับใช้งานหลายพีเจียร์ร่วมกัน

ผลการทดสอบพบว่า ระบบสามารถทำงานได้อย่างต่อเนื่องโดยไม่เกิดอาการค้างของกระบวนการ หรือการหยุดทำงานของเซิร์ฟเวอร์ การจัดการไฟล์ภาพชั่วคราวในโฟลเดอร์ uploads เป็นไปอย่างเหมาะสม โดยมีการลบไฟล์หลังการประมวลผลเสร็จสิ้น ไม่พบการสะสมของไฟล์ตกค้าง

ในด้านการใช้ทรัพยากร ระบบมีการใช้หน่วยความจำ และหน่วยประมวลผลในระดับคงที่ ไม่พบปัญหา Memory Leak และสามารถรองรับคำร้องขอพร้อมกันได้

จากการประเมินเวลาในการประมวลผลต่อ 1 คำร้องขอในส่วนของ การประมวลผล การตรวจจับวัตถุ ร่วมกับการประเมินค่าความลึกลับเครื่อง Macbook Air M3 พบว่าในกรณีที่โมเดลถูกโหลดไว้ในหน่วยความจำล่วงหน้า (Warm Start) ระบบใช้เวลาในการประมวลผลเฉลี่ยประมาณ 80-120 มิลลิวินาทีต่อภาพ ทำให้สามารถรองรับคำร้องขอได้ประมาณ 5-8 คำร้องขอต่อวินาที ภายใต้สภาวะการใช้งานปกติ

อย่างไรก็ตามในกรณีที่โมเดลไม่ได้ถูกโหลดไว้ในหน่วยความจำล่วงหน้า (Cold Start) พบว่าเวลาตอบสนองอาจเพิ่มขึ้นได้สูงสุดประมาณ 8 วินาทีต่อคำร้องขอ ทั้งนี้ระยะเวลาดังกล่าวรวมขั้นตอนการจัดการไฟล์ การส่งต่อข้อมูลระหว่าง Express และ Flask ตลอดจนกระบวนการโหลดโมเดลและประมวลผลภาพทั้งหมด เมื่อเพิ่มอัตราการส่งคำร้องขอสูงกว่า 10 คำร้องขอต่อวินาที พบว่าเวลาตอบสนอง (Response Time) จะเพิ่มขึ้นตามลำดับ เนื่องจากข้อจำกัดด้านทรัพยากรของหน่วยประมวลผล แต่ไม่พบการล่มของระบบ (System Crash) หรือการหยุดทำงานของเซิร์ฟเวอร์ ระบบยังสามารถตอบสนองคำร้องขอได้ครบถ้วน แม้จะมีความหน่วงเพิ่มขึ้น ในบริบทของการใช้งานจริงของแอปพลิเคชัน Seeable

ซึ่งมีอัตราการส่งภาพเฉลี่ยประมาณ 1-3 คำร้องขอต่อวินาทีต่อผู้ใช้งาน 1 ราย พบว่าระบบสามารถรองรับได้อย่างเหมาะสม และให้ประสบการณ์ใช้งานในระดับกึ่งเรียลไทม์ (Near Real-Time) โดยไม่เกิดปัญหาคอขวด (Bottleneck) อย่างมีนัยสำคัญ

จากผลการทดลองสามารถสรุปได้ว่า ระบบฝังเซิร์ฟเวอร์มีความเสถียรในการทำงานต่อเนื่อง สามารถประสานงานกับโมดูล Flask ได้อย่างมีประสิทธิภาพ และมีสมรรถนะเพียงพอต่อการใช้งานจริงภายในอาคารตามวัตถุประสงค์ของงานวิจัย

4.3 ผลการทดสอบระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Flask)

4.3.1 ผลการฝึกโมเดลการตรวจจับสิ่งกีดขวางด้วย YOLOv8n

โมเดลตรวจจับสิ่งกีดขวางใช้โมเดล YOLOv8n ซึ่งเป็นโมเดลแบบ Lightweight ที่เหมาะสมกับงานตรวจจับวัตถุแบบเรียลไทม์ โดยทำฝึกด้วยชุดข้อมูลสิ่งกีดขวางจากอาคาร C สถาบันเทคโนโลยีไทย-ญี่ปุ่น ประกอบด้วยวัตถุ 14 ประเภท ได้แก่ ตู้ (Cabinet), เก้าอี้ (Chair), สิ่งของที่วางอยู่บริเวณทางเดิน (Clutter), ประตู (Door), ลิฟต์ (Elevator), ป้ายทางออกฉุกเฉิน (Exit Sign), ถังดับเพลิง (Fire Extinguisher), คน (Person), บันได (Stairs), โต๊ะ (Table), ป้ายสัญลักษณ์ห้องน้ำ (Toilet Sign), ถังขยะ (Trash Bin), ไวท์บอร์ด (Whiteboard) และหน้าต่าง (Window)

การฝึกโมเดลดำเนินการบนเครื่องที่ใช้หน่วยประมวลผลกราฟิก (GPU) NVIDIA GeForce RTX 2060 โดยกำหนดขนาดภาพที่ 640x640 พิกเซล ใช้ Batch Size เท่ากับ 16 และฝึกเป็นจำนวน

Epoch (Best)	Box Loss	CLS Loss	DFL Loss	GPU Memory	mAP50	mAP50-95
115 / 200	0.59	0.3491	0.9269	2.65	0.944	0.724

รูปที่ 4.12 ผลการฝึกโมเดลการตรวจจับสิ่งกีดขวางใน Epoch ที่ดีที่สุด

120 Epochs พร้อมทั้งกำหนด Early Stopping เพื่อลดความเสี่ยงของการเกิด Overfitting โดยเลือกใช้ Optimizer แบบ AdamW และปรับค่า Learning Rate ให้เหมาะสมกับลักษณะของข้อมูล รวมถึงใช้เทคนิค Data Augmentation และการปรับค่าแสงสี เพื่อเพิ่มความหลากหลายของข้อมูลฝึก

จากรูปที่ 4.12 แสดงผลการฝึกโมเดลการตรวจจับสิ่งกีดขวาง ซึ่งประกอบด้วยค่า Loss และค่าประเมินผลตามจำนวน Epoch พบว่า

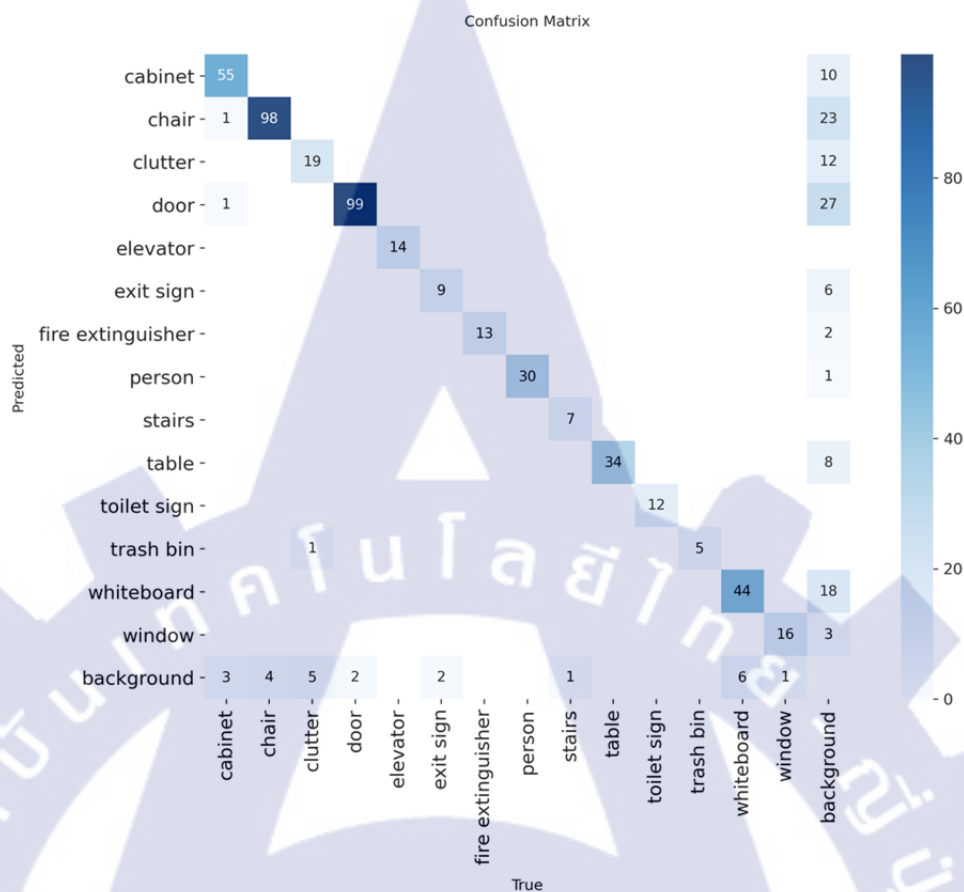
1. ค่า box_loss และ cls_loss ลดลงอย่างต่อเนื่องตลอดการฝึก
2. ค่า mAP50 เพิ่มขึ้นอย่างสม่ำเสมอ และเข้าสู่ภาวะคงที่ในช่วงท้าย

3. ระบบมีการหยุดฝึกอัตโนมัติ (Early Stopping) เมื่อผลลัพธ์ไม่ดีขึ้นในช่วง Epoch ที่กำหนด

จากข้อมูลค่า mAP50 เท่ากับ 0.944 และค่า mAP50-95 เท่ากับ 0.724 แสดงให้เห็นว่าโมเดลมีความสามารถในการตรวจจับ และระบุตำแหน่งวัตถุได้แม่นยำในระดับที่เหมาะสมต่อการนำไปใช้งานจริง โดยเฉพาะในสภาพแวดล้อมที่มีวัตถุหลากหลายประเภท และมีการจัดวางที่แตกต่างกัน เมื่อพิจารณาค่า Box Loss (0.59) และ Classification Loss (0.3491) ซึ่งมีแนวโน้มลดลงอย่างต่อเนื่อง สะท้อนให้เห็นถึงความสามารถของโมเดลในการเรียนรู้ตำแหน่ง และจำแนกประเภทของวัตถุได้อย่างมีประสิทธิภาพ

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
all	120	482	0.929	0.888	0.948	0.729
cabinet	40	60	0.912	0.868	0.946	0.793
chair	60	102	0.919	0.895	0.959	0.719
clutter	18	25	0.848	0.671	0.813	0.563
door	73	101	0.891	0.911	0.956	0.747
elevator	8	14	0.982	1.000	0.995	0.846
exit sign	11	11	0.800	0.727	0.904	0.658
fire extinguisher	13	13	0.972	1.000	0.995	0.875
person	30	30	1.000	0.998	0.995	0.770
stairs	8	8	0.980	0.875	0.926	0.743
table	32	34	0.916	0.967	0.974	0.743
toilet sign	8	12	0.989	1.000	0.995	0.714
trash bin	5	5	0.979	1.000	0.995	0.686
whiteboard	40	50	0.817	0.713	0.848	0.662
window	17	17	1.000	0.803	0.979	0.689

รูปที่ 4.13 ประสิทธิภาพของโมเดลการตรวจจับสิ่งกีดขวางรายการคลาส



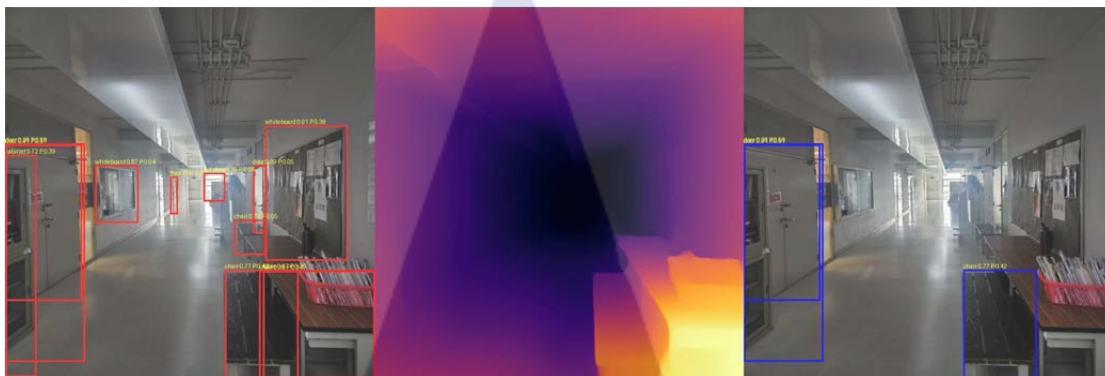
รูปที่ 4.14 Confusion Matrix ของโมเดลการตรวจจับสิ่งกีดขวาง

จากรูปที่ 4.13 และ 4.14 ที่แสดงประสิทธิภาพของโมเดลการตรวจจับสิ่งกีดขวาง ซึ่งใช้วิเคราะห์ความแม่นยำรายคลาส พบว่า

1. คลาสที่มีจำนวนข้อมูลมาก เช่น เก้าอี้ และประตู มีค่า True Positive สูง
2. คลาสที่มีลักษณะคล้ายกัน เช่น ตู้ และไวท์บอร์ด มีค่าความสับสนกันในบางกรณี
3. คลาสที่มีจำนวนตัวอย่างน้อย เช่น สิ่งของที่วางอยู่บริเวณทางเดิน มีค่า Recall น้อยกว่า

คลาสอื่น

จากการวิเคราะห์สะท้อนให้เห็นว่าปริมาณข้อมูลในแต่ละคลาสมีผลโดยตรงต่อประสิทธิภาพของโมเดล และเป็นแนวทางในการปรับปรุงชุดข้อมูลในอนาคต



รูปที่ 4.15 ตัวอย่างผลการผสานการตรวจจับวัตถุ การประเมินค่าความลึก และค่าลำดับความสำคัญ พร้อมแสดง Bounding Box และ Priority Score

4.3.2 ผลการผสานการตรวจจับวัตถุ การประเมินค่าความลึก และค่าลำดับความสำคัญ

การประเมินสิ่งกีดขวางไม่ได้อาศัยเพียงผลการตรวจจับวัตถุเพียงอย่างเดียว แต่เป็นการผสานผลลัพธ์จากโมเดล YOLOv8n ร่วมกับโมเดลประมาณค่าความลึก (MiDaS Depth Estimation) และค่าลำดับความสำคัญ (Priority Score) ของวัตถุแต่ละรายการ ก่อนส่งผลกลับไปยังแอปพลิเคชัน ซึ่งประกอบด้วย 3 ขั้นตอนหลัก คือ

1. การตรวจจับวัตถุ และระบุตำแหน่ง Bounding Box
2. คำนวณค่าความลึกเฉลี่ยภายใน Bounding Box
3. คำนวณค่าลำดับความสำคัญโดยพิจารณาจากระยะทาง ความเสี่ยงของวัตถุแต่ละประเภท และตำแหน่งเชิงทิศทางเมื่อเทียบกับแนวทางการเคลื่อนที่ของผู้ใช้งาน

จากรูปที่ 4.15 ตัวอย่างผลการผสานการตรวจจับวัตถุ การประเมินค่าความลึก และค่าลำดับความสำคัญ พบว่า

1. วัตถุที่อยู่ในแนวการเดินด้านหน้า และมีค่าความลึกต่ำ (อยู่ใกล้ผู้ใช้งาน) จะมีค่าลำดับความสำคัญสูง
2. วัตถุที่อยู่ด้านข้างหรือมีระยะห่างมาก จะมีค่าลำดับความสำคัญต่ำกว่า
3. ระบบสามารถเลือกแจ้งเตือนเฉพาะวัตถุที่มีลำดับความสำคัญสูงสุด ช่วยลดการแจ้งเตือนที่ซ้ำซ้อน

ผลจากการทดลองในพื้นที่จริงภายในอาคาร C ชั้น 3 สถาบันเทคโนโลยีไทย-ญี่ปุ่นแสดงให้เห็นว่า ระบบสามารถจัดลำดับวัตถุที่มีความเสี่ยงสูงได้อย่างเหมาะสม ลดการแจ้งเตือนวัตถุที่ไม่อยู่ในแนวการเดิน และลดการแจ้งเตือนที่ซ้ำซ้อนหรือไม่จำเป็น ซึ่งช่วยลดความสับสนแก่ผู้ใช้งาน

เมื่อเปรียบเทียบการแจ้งเตือนแบบใช้การตรวจจับวัตถุเพียงอย่างเดียวกับการผสมผสานการประเมินค่าความลึก และค่าลำดับความสำคัญ พบว่าวิธีผสมผสานดังกล่าวช่วยลดการแจ้งเตือนที่ไม่จำเป็นลงได้อย่างชัดเจน โดยเฉพาะกรณีที่มีวัตถุหลายรายการในเฟรมภาพเดียวกัน

ดังนั้นการผสมผสานวิธีดังกล่าวนี้จึงเป็นส่วนสำคัญในการที่ช่วยเพิ่มความแม่นยำ ความเหมาะสม และประสิทธิภาพของการแจ้งเตือนในแอปพลิเคชัน Seeable ทำให้การใช้งานมีประสิทธิภาพ และสอดคล้องกับบริบทของผู้ใช้การทางสายตามากยิ่งขึ้น

4.4 ผลการประเมินเชิงเปรียบเทียบ (Comparative Evaluation)

ในส่วนนี้เป็นการประเมินประสิทธิภาพของระบบโดยเปรียบเทียบผลลัพธ์ของระบบกับค่ามาตรฐานอ้างอิง (Ground Truth) ที่ได้จากผู้ประเมิน เพื่อสะท้อนให้เห็นถึงความแม่นยำ และความเหมาะสมของระบบในการใช้งานจริง โดยมุ่งเน้นการประเมินใน 2 ประเด็นหลัก ได้แก่ การแจ้งเตือนสิ่งกีดขวาง และการนำทางภายในอาคาร

4.4.1 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินกับระบบ

ในการประเมินการแจ้งเตือนสิ่งกีดขวาง ผู้วิจัยเลือกใช้วิธีการประเมินเชิงคุณภาพ (Qualitative Evaluation) โดยทำการเปรียบเทียบผลลัพธ์ของระบบกับการตัดสินใจของผู้ประเมิน ซึ่งทำหน้าที่เป็นค่ามาตรฐานอ้างอิง (Ground Truth)

กระบวนการประเมินดำเนินการโดยการนำภาพจากสภาพแวดล้อมจริงภายในอาคารมาใช้ทดสอบ จากนั้นให้ผู้ประเมินพิจารณาวัตถุที่ปรากฏในภาพ และเลือกวัตถุที่ควรได้รับการแจ้งเตือน โดยอ้างอิงจากตำแหน่งของวัตถุในแนวการเดิน ระยะห่าง และระดับความเสี่ยงต่อผู้ใช้งาน

ในขณะเดียวกัน ระบบจะทำการประมวลผลภาพด้วยโมเดลตรวจจับวัตถุ ร่วมกับการประมาณค่าความลึก และการคำนวณค่าลำดับความสำคัญ เพื่อเลือกวัตถุที่ควรแจ้งเตือน จากนั้นนำผลลัพธ์ของระบบมาเปรียบเทียบกับผลการตัดสินใจของผู้ประเมิน



รูปที่ 4.16 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินและระบบในกรณีที่มีวัตถุหลายรายการในแนวทางเดิน



รูปที่ 4.17 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินและระบบในสภาพแวดล้อมทางเดินที่มีระยะลึกและวัตถุซ้อนกัน



รูปที่ 4.18 การเปรียบเทียบการแจ้งเตือนสิ่งกีดขวางระหว่างผู้ประเมินและระบบในกรณีที่มีวัตถุใกล้และวัตถุไกลในภาพเดียวกัน

จากผลการประเมินพบว่า ระบบสามารถเลือกวัตถุที่ควรแจ้งเตือนได้สอดคล้องกับการตัดสินใจของผู้ประเมินในกรณีส่วนใหญ่ โดยเฉพาะวัตถุที่อยู่ในแนวการเดินทางด้านหน้า และมีระยะใกล้ ซึ่งเป็นวัตถุที่มีความเสี่ยงสูงต่อการเคลื่อนที่ของผู้ใช้งาน ดังแสดงในรูปที่ 4.16 ซึ่งเป็นกรณีที่มิววัตถุหลายรายการในแนวทางเดิน ระบบสามารถเลือกแจ้งเตือนวัตถุหลักที่อยู่ด้านหน้าได้สอดคล้องกับผู้ประเมิน

ในรูปที่ 4.17 ซึ่งเป็นสภาพแวดล้อมที่มีระยะใกล้และมีวัตถุซ้อนกันหลายระดับ พบว่าระบบยังสามารถระบุวัตถุที่อยู่ในแนวการเดินทางได้อย่างเหมาะสม แม้ว่าจะมีวัตถุในระยะไกลปรากฏอยู่ในภาพร่วมด้วย โดยระบบให้ความสำคัญกับวัตถุที่อยู่ใกล้และมีผลต่อการเคลื่อนที่มากกว่า

ส่วนในรูปที่ 4.18 ซึ่งมีทั้งวัตถุใกล้และวัตถุไกลในภาพเดียวกัน ระบบสามารถเลือกแจ้งเตือนวัตถุที่อยู่ในระยะใกล้ได้สอดคล้องกับผู้ประเมิน และลดการแจ้งเตือนวัตถุที่อยู่ไกลหรืออยู่นอกแนวการเดิน ซึ่งช่วยลดความสับสนในการรับรู้ข้อมูลของผู้ใช้งาน และทำให้การแจ้งเตือนมีความเหมาะสมมากยิ่งขึ้น

อย่างไรก็ตาม ในบางกรณีที่มีวัตถุหลายรายการในภาพเดียวกัน หรือมีวัตถุที่มีลักษณะใกล้เคียงกัน ระบบอาจมีการเลือกวัตถุที่แตกต่างจากผู้ประเมินเล็กน้อย โดยเฉพาะวัตถุที่ไม่ได้อยู่ในแนวการเดินโดยตรง แต่ยังอยู่ในขอบเขตของภาพ

4.4.2 การเปรียบเทียบเส้นทางการนำทางระหว่างการเดินจริงกับระบบ

ในการประเมินส่วนนี้ ผู้วิจัยได้ทำการเปรียบเทียบเส้นทางการนำทางที่ได้จากระบบกับเส้นทางที่ผู้ทดลองเดินจริงภายในอาคาร เพื่อใช้เป็นข้อมูลอ้างอิงสำหรับประเมินความถูกต้องของอัลกอริทึมการนำทาง

กระบวนการทดสอบเริ่มจากกำหนดจุดเริ่มต้นและจุดหมายปลายทาง จากนั้นให้ผู้ทดลองเดินไปยังจุดหมายด้วยตนเอง โดยบันทึกระยะทางจริงที่ใช้ในการเดิน แล้วนำมาเปรียบเทียบกับระยะทางที่ระบบคำนวณได้ ซึ่งได้จากโครงสร้างการนำทางแบบโหนดของอาคาร การประเมินผลพิจารณาจาก

1. ความสอดคล้องของเส้นทางที่ระบบคำนวณกับเส้นทางที่ผู้ทดลองใช้จริง
2. ค่าความคลาดเคลื่อนของระยะทาง
3. ความสามารถของระบบในการนำผู้ใช้งานไปยังจุดหมายได้อย่างถูกต้อง

ตารางที่ 4.1 ผลการเปรียบเทียบเส้นทางการนำทางระหว่างการเดินจริงกับระบบ

จุดเริ่มต้น	จุดหมายปลายทาง	ระยะทางที่ระบบคำนวณ (เมตร)	ระยะทางจริง (เมตร)	ค่าคลาดเคลื่อน (เมตร)	ผลการนำทาง
C301F	C303B	30	30	0	สำเร็จ
C302F	C305F	3	3	0	สำเร็จ
C303B	C307B	0	0	0	สำเร็จ
C301F	C314F	8	8	0	สำเร็จ

จากผลการทดลองดังตารางที่ 4.1 พบว่าระยะทางที่ระบบคำนวณได้มีค่าเท่ากับระยะทางที่ได้จากการเดินจริงในทุกกรณีทดสอบ ส่งผลให้ค่าความคลาดเคลื่อนของระยะทางมีค่าเท่ากับ 0 เมตร ผลลัพธ์ดังกล่าวแสดงให้เห็นว่าอัลกอริทึมการคำนวณเส้นทางที่พัฒนาขึ้นมีความสอดคล้องกับโครงสร้างของผังอาคาร และสามารถคำนวณระยะทางได้อย่างถูกต้องภายใต้เงื่อนไขของการทดลอง

อย่างไรก็ตาม ควรพิจารณาว่าการทดลองดำเนินการภายใต้สภาพแวดล้อมที่มีการควบคุม และเส้นทางที่กำหนดไว้อย่างชัดเจน ซึ่งอาจแตกต่างจากการใช้งานจริงที่มีปัจจัยอื่นเข้ามาเกี่ยวข้อง เช่น พฤติกรรมการเดินของผู้ใช้งาน หรือสิ่งกีดขวางที่เกิดขึ้นแบบไม่คาดคิด



บทที่ 5

สรุปและข้อเสนอแนะ

จากการพัฒนาแอปพลิเคชัน Seeable ซึ่งเป็นแอปพลิเคชันสำหรับการนำทางในที่ร่มและการแจ้งเตือนสิ่งกีดขวางสำหรับผู้พิการทางสายตา งานวิจัยนี้มุ่งเน้นการช่วยให้ผู้ใช้งานสามารถเคลื่อนที่ภายในอาคารได้อย่างปลอดภัยและมีประสิทธิภาพมากยิ่งขึ้น โดยอาศัยการนำทางด้วยมาร์กเกอร์ความเป็นจริงเสริม (Augmented Reality Marker: AR Marker) เป็นจุดอ้างอิงเชิงตำแหน่ง ร่วมกับการตรวจจับวัตถุ (Object Detection) เพื่อแจ้งเตือนสิ่งกีดขวาง และการตรวจจับอักขระด้วยแสง (Optical Character Recognition) เพื่ออ่านตัวหนังสือจากสภาพแวดล้อมจริงเพื่อให้ข้อมูลแก่ผู้ใช้งานในรูปแบบเสียง ระบบได้รับการออกแบบและทดสอบภายในอาคาร C ชั้น 3 ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น บทนี้นำเสนอข้อสรุปของการวิจัย ปัญหา และข้อจำกัดที่พบ รวมถึงข้อเสนอแนะสำหรับการพัฒนางานวิจัยในอนาคต

5.1 ข้อสรุปของงานวิจัย

จากการดำเนินงานวิจัยสามารถสรุปได้ว่า วัตถุประสงค์ของงานวิจัยที่กำหนดไว้สามารถบรรลุผลได้ตามเป้าหมายที่ตั้งไว้ แอปพลิเคชัน Seeable สามารถช่วยให้ผู้พิการทางสายตารับรู้ และโต้ตอบกับสิ่งแวดล้อมภายในอาคารได้ผ่านการนำทางด้วยเสียง การแจ้งเตือนสิ่งกีดขวาง และอ่านตัวอักษรจากสภาพแวดล้อมจริง

ระบบการนำทางในที่ร่มที่พัฒนาขึ้นสามารถระบุตำแหน่งผู้ใช้งานโดยใช้มาร์กเกอร์ความเป็นจริงเสริมเป็นจุดอ้างอิงเชิงตำแหน่ง และสามารถกำหนดเส้นทางการเดินภายในอาคารได้อย่างเหมาะสม นอกจากนี้ ระบบตรวจจับวัตถุ และระบบตรวจจับอักขระด้วยแสงสามารถทำงานร่วมกันเพื่อแจ้งเตือนสิ่งกีดขวาง และแปลงข้อมูลตัวอักษรเป็นเสียง ช่วยเพิ่มความสะดวก และความปลอดภัยในการใช้งานภายในอาคาร

ผลการทดสอบภายในอาคาร C ชั้น 3 ของสถาบันเทคโนโลยีไทย-ญี่ปุ่นแสดงให้เห็นว่าแอปพลิเคชันสามารถทำงานได้ตามขอบเขตที่กำหนด และสามารถใช้งานได้อย่างมีประสิทธิภาพบนสมาร์ตโฟนทั้งระบบปฏิบัติการ iOS และ Android ซึ่งแสดงให้เห็นถึงศักยภาพในการนำไปประยุกต์ใช้งานจริงสำหรับผู้พิการทางสายตาในสภาพแวดล้อมภายในอาคาร

5.2 ปัญหา และข้อจำกัดของงานวิจัย

จากการพัฒนา และทดสอบแอปพลิเคชัน Seeable พบปัญหา และข้อจำกัดที่สำคัญหลายประการ ประการแรก คือ ปัญหาด้านการตรวจจับมาร์กเกอร์ความเป็นจริงเสริม ซึ่งความแม่นยำของการตรวจจับอาจได้รับผลกระทบจากสภาพแสง มุมกล้อง และระยะห่างระหว่างกล้องกับมาร์กเกอร์ ส่งผลให้บางกรณีระบบไม่สามารถระบุตำแหน่งของผู้ใช้งานได้อย่างต่อเนื่อง

ประการที่สอง คือ ปัญหาด้านประสิทธิภาพในการใช้งานระบบ ซึ่งแม้ว่าการประมวลผลหลัก เช่น การตรวจจับวัตถุจะดำเนินการบนเซิร์ฟเวอร์เป็นหลัก แต่ประสิทธิภาพของระบบยังคงขึ้นอยู่กับความเร็วของการเชื่อมต่อเครือข่าย ในกรณีที่สัญญาณอินเทอร์เน็ตไม่เสถียรอาจส่งผลให้การส่งข้อมูลภาพ และการรับผลการประมวลผลเกิดความล่าช้าได้ นอกจากนี้ การเปิดใช้งานกล้องของสมาร์ทโฟนเป็นระยะเวลานานอาจส่งผลให้แอปพลิเคชันเกิดอาการหน่วงหรือค้างโดยเฉพาะบนอุปกรณ์ที่มีทรัพยากรจำกัด

5.3 สำหรับการพัฒนางานวิจัยในอนาคต

เพื่อเพิ่มประสิทธิภาพ และขยายขอบเขตการใช้งานของแอปพลิเคชัน Seeable ในอนาคต ควรมีการพัฒนาเพื่อให้สามารถรองรับพื้นที่ที่หลากหลายมากขึ้น เช่น อาคารหลายชั้น หรืออาคารที่มีโครงสร้างซับซ้อน รวมถึงการเพิ่มจำนวนมาร์กเกอร์ และปรับปรุงแบบการติดตั้งให้เหมาะสมกับลักษณะของแต่ละพื้นที่

ควรพัฒนาระบบการตรวจจับวัตถุ และการตรวจจับอักขระด้วยแสงให้มีความแม่นยำ และความเร็วสูงขึ้น โดยการเพิ่ม และปรับปรุงชุดข้อมูล (Dataset) สำหรับการฝึกให้ครอบคลุมสภาพแวดล้อมที่หลากหลาย เช่น ความแตกต่างของแสง มุมมอง และลักษณะของวัตถุภายในอาคาร เพื่อให้ระบบสามารถตรวจจับ และแจ้งเตือนได้อย่างถูกต้องมากขึ้นในสถานการณ์จริง นอกจากนี้ อาจพิจารณาการนำข้อมูลจากเซนเซอร์ของสมาร์ทโฟน เช่น ตัววัดการเคลื่อนไหวหรือทิศทางมาประมวลผลร่วมกับมาร์กเกอร์ความเป็นจริงเสริม เพื่อช่วยประเมินตำแหน่ง และทิศทางการเคลื่อนที่ของผู้ใช้งาน และลดการพึ่งพาการสแกนมาร์กเกอร์เพียงอย่างเดียว

เนื่องจากระบบในปัจจุบันอาศัยการประมวลผลบนเซิร์ฟเวอร์เป็นหลัก การใช้งานจึงยังมีข้อจำกัดในกรณีที่ไม่สามารถเชื่อมต่ออินเทอร์เน็ตได้ ดังนั้นงานวิจัยในอนาคตควรพิจารณาการพัฒนาให้สามารถประมวลผลบนสมาร์ทโฟนที่มีทรัพยากรจำกัดได้อย่างมีประสิทธิภาพ และสามารถทำงานได้ทั้งขณะเชื่อมต่ออินเทอร์เน็ต และไม่ได้เชื่อมต่ออินเทอร์เน็ต เพื่อลดความล่าช้าในการประมวลผล และเพิ่มความต่อเนื่องในการใช้งาน นอกจากนี้ โมเดลบางประเภท เช่น การตรวจจับวัตถุ และการประเมินค่าความลึก ยังมีข้อจำกัดในการนำมาประมวลผลโดยตรงสำหรับการพัฒนาด้วยเฟรมเวิร์กข้ามแพลตฟอร์ม (Cross-Platform Framework) หากในอนาคตสามารถพัฒนาให้สามารถรองรับการประมวลผลโมเดลเหล่านี้บนสมาร์ทโฟน

ได้โดยตรง จะช่วยเพิ่มประสิทธิภาพของระบบ ลดการพึ่งพาเซิร์ฟเวอร์ และทำให้แอปพลิเคชันสามารถทำงานได้อย่างราบรื่นยิ่งขึ้น

นอกจากนี้ในอนาคตควรมีการนำแอปพลิเคชันไปทดสอบกับผู้พิการทางสายตาในสภาพแวดล้อมการใช้งานจริง เพื่อประเมินประสบการณ์การใช้งาน ความเข้าใจในการนำทาง และความเหมาะสมของการแจ้งเตือนในสถานการณ์ต่างๆ อย่างเป็นระบบ โดยผลการทดสอบดังกล่าวจะช่วยให้สามารถปรับปรุงระบบให้สอดคล้องกับความต้องการของผู้ใช้งานจริง และเพิ่มความน่าเชื่อถือในการนำไปใช้งานจริงในชีวิตประจำวัน



บรรณานุกรม

TNI

บรรณานุกรม

- [1] กรมส่งเสริมและพัฒนาคุณภาพชีวิตคนพิการ, “สถิติคนพิการในประเทศไทย,” [Online]. Available : <https://dep.go.th/images/uploads/files/Dashboard012569.pdf>. [Accessed : 10 มีนาคม 2569].
- [2] M. Yang and X. Fan, “YOLOv8-lite : A lightweight object detection model for real-time autonomous driving systems,” *IECE Transactions on Emerging Topics in Artificial Intelligence*, vol. 1, no. 1, pp. 1-16, April 2024.
- [3] R. A. Welikala et al., “Automated detection and classification of oral lesions using deep learning for early detection of oral cancer,” *IEEE Access*, vol. 8, no. 4, pp. 132,677-132,693, July 2020.
- [4] Z. Xue et al., “Enhancement and fusion of multi-scale feature maps for small object detection,” *39th Chinese Control Conference, NEDSI 2020*, Shenyang, China, July 27-29, 2020, pp. 7,212-7,217.
- [5] C. Gao et al., “YOLOv4 object detection algorithm with efficient channel attention mechanism,” *5th International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2020*, New Jersey, USA, December 25-27, 2020, pp. 1,764-1,770.
- [6] Z. Pan et al., “Lightweight spatial sliced-concatenate-multireceptive-field enhance and joint channel attention mechanism for infrared object detection,” *Journal of Physics*, vol. 10, no. 3, pp. 55,508-55,521, May 2022.
- [7] S. Li et al., “YOLO-FIRI : Improved YOLOv5 for infrared image object detection,” *European Journal of Science and Technology*, vol. 9, no. 5, pp. 141,861-141,873, October 2021.
- [8] M. P. Groover, “How to Implement Object Detection Using Deep Learning : A Step-by-Step Guide,” [Online]. Available : <https://www.augmentedstartups.com/blog/how-to-implement-object-detection-using-deep-learning-a-step-by-step-guide>. [Accessed : March 15, 2025].
- [9] A. Chinapas et al., “Personal verification system using thai ID card and face photo for cross-age face,” *International Computer Science and Engineering Conference, ICSEC 2019*, Phuket, Thailand, October 28-30, 2019, pp. 325-330.

- [10] A. Chuttergoon and L. Nagowah, "Eye guide : An assistive mobile technology for visually impaired Persons," *International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering, ELECOM 2020*, Balaclava, Mauritius, November 25-27, 2020, pp. 198-203.
- [11] Q. Nguyen et al., "An efficient unsupervised approach for OCR error correction of vietnamese OCR text," *Neural Computing and Applications*, vol. 11, no. 7, pp. 58,406-58,421, June 2023.
- [12] K. Elferjani et al., "Assisting blind people using object detection with vocal reedback," *International Conference on Communications, Computing, Cybersecurity, and Informatics, CCCI 2020*, Sharjah, United Arab Emirates, November 3-5, 2020, pp. 1-5.
- [13] A. M. Norkhalid et al., "Mobile application : Mobile assistance for visually impaired people-speech interface system," *International Conference on Information Technology and Multimedia, ICIMU 2020*, Selangor, Malaysia, August 24-25, 2020, pp. 329-333.
- [14] H. L. Warnars et al., "Mobile application for the blind and their family," *TEM Journal*, vol. 10, no. 3, pp. 1,039-1,044, August 2021.
- [15] M. Artikova and M. Artikov, "Augmented reality technologies in mobile applications to optimize user experience in museum tours," *International Conference on Advanced Information Scientific Development, ICAISD 2024*, Jawa Barat, Indonesia, November 25-26, 2024, pp. 31-34.
- [16] W. Li et al., "Animated virtual therapist in virtual reality environment for people with parkinson's disease," *Advances in Neural Information Processing Systems, NIPS 2017, Long Beach, CA, USA*, January 27-28, 2025, pp. 286-289.
- [17] T. Kitsukawa et al., "A recognition system of AR markers attached to carts in a factory," *International Conference on Information and Communication Technology Convergence, ICIT 2022*, Ghent, Belgium, January 9-12, 2022, pp. 608-613.
- [18] T. Okaya et al., "A study on indoor positioning of an open-source drone using AR markers," *VTS Asia Pacific Wireless Communications Symposium, APWCS 2023*, Tainan City, Taiwan, August 23-25, 2023, pp. 1-5.
- [19] J. Zhu et al., "Multi-stage local path planning algorithm based on obstacles as anchor points," *International Conference on Advanced Algorithms and Control Engineering, ICAACE 2024*, Shanghai, China, March 1-3, 2024, pp. 1,440-1,445.

- [20] Y. Tao and J. Du, "Great circle path planning for USV voyages across oceans with irregular static obstacles," *Asian Control Conference, ASCC 2024*, Dalian, China, July 5-8, 2024, pp. 202-207.
- [21] X. Yu and J. Saniie, "Visual impairment spatial awareness system for indoor navigation and daily activities," *Journal of Imaging*, vol. 11, no. 1, pp. 9, January 2025.
- [22] B. S. C. Putra et al., "Adaptive AR navigation : Real-time mapping for indoor environment using node placement and marker localization," *Journal of Physics : Conference Series*, vol. 16, no. 6, pp. 478, June 2025.
- [23] F. Vitale and C. Roncoli, "Reference tracking optimization with obstacle avoidance via task prioritization for automated driving," *Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 1,200-1,214, February 2024.
- [24] H. Wang et al. "Risk-aware reinforcement learning with dynamic safety filter for collision risk mitigation in mobile robot navigation," *Proceeding of IEEE*, vol. 25, no. 17, pp. 5,488, September 2025.
- [25] Z. Long et al., "Human-risk-aware safe path planning based on reinforcement learning for autonomous mobile robots," *International Journal of Computer & Organization Trends*, vol. 25, no. 23, pp. 7,211, November 2025.
- [26] K. Moller et al., "Pedestrian-aware motion planning for autonomous driving in complex urban scenarios," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 7, no. 3, pp. 365-378, January 2026.
- [27] G. Sun et al., "An image object detection model based on mixed attention mechanism optimized YOLOv5," *KKU Engineering Journal*, vol. 12, no. 7, pp. 1,515, March 2023.
- [28] H. Yi et al., "Small object detection algorithm based on improved YOLOv8 for remote sensing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, no. 7, pp. 1,734-1,747, December 2023.
- [29] Z. Luo and Y. Tian, "Improved infrared road object detection algorithm based on attention mechanism in YOLOv8," *International Journal of Computer Science*, vol. 51, no. 6, pp. 673-680, June 2024.
- [30] P. Kiatphaisansophon et al., "Efficient text bounding box identification using mask R-CNN : Case of thai documents," *Signal & Image Processing : An International Journal (SIPIJ)*, vol. 12, no. 6, pp. 49,306-49,326, April 2024.

- [31] T. Chumwatana and W. Rattana-umnaychai, "Using OCR framework and information extraction for thai documents digitization," *International Electrical Engineering Congress, IEECON 2021*, Pattaya, Thailand, March 10-12, 2021, pp. 440-444.
- [32] M. Awad et al., "Intelligent eye : A mobile application for assisting blind people," *Proceedings of the 2018 Northeast Decision Science Institute Conference, NEDSI 2018*, New Jersey, USA, April 18-20, 2004, pp. 1-6.
- [33] S. Vaidya et al., "Real-time object detection for visually challenged people," *International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, San Francisco, California, May 13-15, 2020, pp. 311-316.



ภาคผนวก

TNI

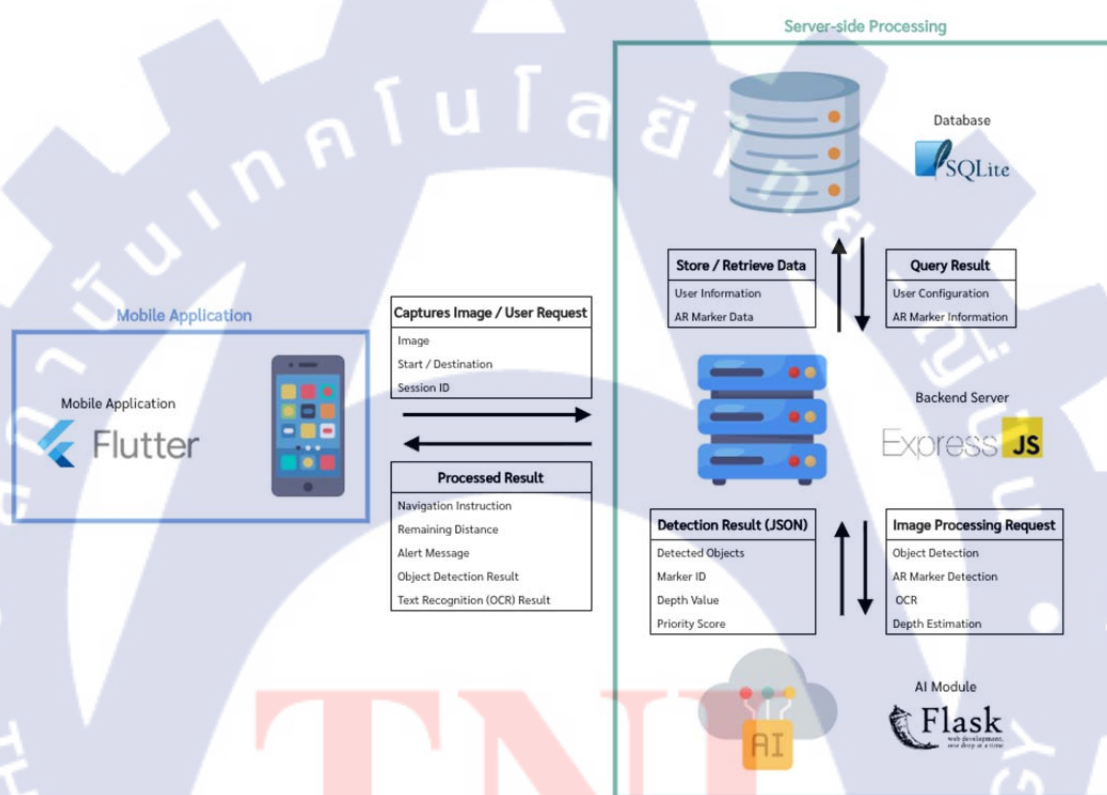
ภาคผนวก ก.

สถาปัตยกรรมระบบ (System Architecture)

TNI

สถาปัตยกรรมระบบ (System Architecture)

แอปพลิเคชัน Seeable ถูกออกแบบโดยใช้สถาปัตยกรรมแบบแยกส่วน (Module Architecture) เพื่อเพิ่มความยืดหยุ่น และความสามารถในการขยายระบบในอนาคต โดยสถาปัตยกรรมของระบบประกอบด้วย 4 ส่วนหลัก ได้แก่ แอปพลิเคชันบนสมาร์ทโฟน (Mobile Application) ระบบฝั่งเซิร์ฟเวอร์ (Backend System) ระบบประมวลผลภาพและปัญญาประดิษฐ์ (Image Processing and Ai Module) และฐานข้อมูล (Database) ซึ่งส่วนประกอบทั้งหมดเชื่อมต่อ และสื่อสารกันผ่านทางเครือข่ายอินเทอร์เน็ต



รูปที่ ก.1 ระบบ System Architecture

ภาคผนวก ข.

แอปพลิเคชันบนอุปกรณ์เคลื่อนที่ (Mobile Application)

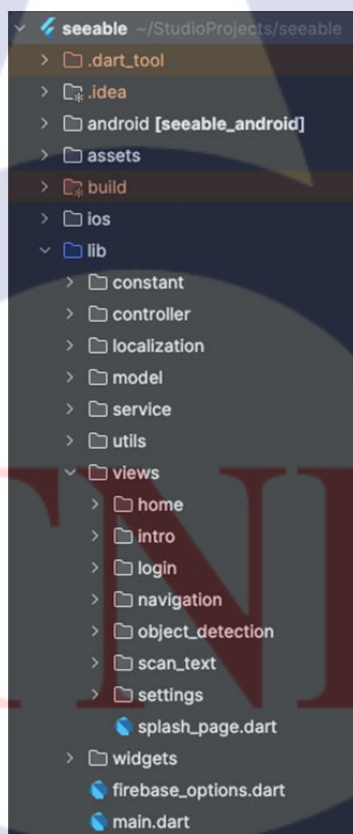
TNI

แอปพลิเคชันบนอุปกรณ์เคลื่อนที่ (Mobile Application)

แอปพลิเคชัน Seeable ถูกพัฒนาด้วยเฟรมเวิร์ค Flutter ซึ่งเป็นเฟรมเวิร์คแบบข้ามแพลตฟอร์ม (Cross-Platform Framework) ที่สามารถรองรับการทำงานได้ทั้งบนระบบปฏิบัติการ iOS และ Android ได้จากโค้ดเบสเดียวกัน แอปพลิเคชันทำหน้าที่เป็นส่วนติดต่อหลักระหว่างผู้ใช้งานกับระบบ โดยมีหน้าที่รับข้อมูลจากอุปกรณ์ของผู้ใช้ ส่งข้อมูลดังกล่าวไปยังระบบฝั่งเซิร์ฟเวอร์เพื่อทำการประมวลผล และนำผลลัพธ์ที่ได้รับกลับมาแสดงในรูปแบบเสียงแจ้งเตือน และคำแนะนำการนำทาง

โครงสร้างของแอปพลิเคชัน

โครงสร้างหลักของแอปพลิเคชันถูกจัดเก็บไว้ภายใต้โฟลเดอร์ lib โดยสามารถอธิบายองค์ประกอบสำคัญได้ดังนี้



1. lib/views

เป็นส่วนที่เกี่ยวข้องกับหน้าจอการทำงาน (UI Screen) ของแอปพลิเคชัน โดยแต่ละโฟลเดอร์จะแทนฟีเจอร์หลักของระบบ เช่น

1. home หน้าหลักของแอปพลิเคชัน
2. login หน้าลงชื่อเข้าใช้ และการจัดการบัญชีผู้ใช้
3. navigation หน้าการนำทางภายในอาคาร
4. object_detection หน้าการตรวจจับวัตถุ
5. scan_text หน้าการอ่านตัวหนังสือ
6. settings หน้าตั้งค่าการใช้งานแอปพลิเคชัน เช่น ภาษา ธีม และความเร็วในแง่เดือน
7. intro หน้าต้อนรับเข้าสู่แอปพลิเคชัน และสิทธิ์การขอเข้าถึงอุปกรณ์

แต่ละฟีเจอร์ภายใต้โฟลเดอร์ Views จะมีการแยกโครงสร้างภายในออกเป็นส่วนย่อยอย่างชัดเจน เพื่อให้การพัฒนาแอปพลิเคชันมีความเป็นระบบ และง่ายต่อการดูแลรักษาในระยะยาว โดยสามารถแบ่งออกเป็น 4 ส่วนหลัก ได้แก่

ส่วนหน้าจอการทำงาน (Page) ทำหน้าที่เป็นส่วนติดต่อกับผู้ใช้ (User Interface) ของแต่ละฟีเจอร์ โดยหน้าจอนี้จะทำหน้าที่ในการสั่งการ และนำเสนอข้อมูล หรือสถานการณ์ทำงานต่อของระบบในรูปแบบที่เหมาะสมกับการใช้งาน

ส่วนควบคุมการทำงาน (Controller) ทำหน้าที่จัดการการทำงานเฉพาะของฟีเจอร์นั้นๆ เช่น การเรียกใช้งาน API การรับ-ส่งข้อมูลกับระบบเซิร์ฟเวอร์

ส่วนโครงสร้างข้อมูล (Model) ใช้ในการแปลงรูปแบบของข้อมูลที่ได้รับจากเซิร์ฟเวอร์ในรูปแบบ JSON ให้อยู่ในรูปแบบที่สามารถนำไปใช้ในภายในแอปพลิเคชันได้

ส่วนประกอบย่อยของหน้าจอ (Widgets) เป็นส่วนองค์ประกอบย่อยที่ใช้ในฟีเจอร์นั้นๆ เช่น ปุ่มควบคุม ป้ายแสดงข้อความ หรือส่วนแสดงสถานะต่างๆ ซึ่งสามารถนำกลับมาใช้ซ้ำได้ ช่วยลดความซ้ำซ้อนของโค้ด และทำให้โครงสร้างของหน้าจอการทำงานมีความเป็นระเบียบมากขึ้น

2. lib/controller

เป็นส่วนดึงข้อมูลสำคัญของแอปพลิเคชัน เช่น เวอร์ชันของแอปพลิเคชันที่ติดตั้งภายในสมาร์ทโฟน รุ่นของสมาร์ทโฟนที่ใช้รันระบบ

3. lib/service

ใช้สำหรับการติดต่อสื่อสารกับระบบฝั่งเซิร์ฟเวอร์ โดยเฉพาะการเรียกใช้งาน REST API เช่น การส่งภาพที่ได้จากกล้องเพื่อการตรวจจับวัตถุ การส่งคำสั่งในการเริ่มต้น และสิ้นสุดการนำทาง

4. lib/utls

เป็นส่วนของฟังก์ชันช่วยเหลือ (Utility Functions) ที่ถูกใช้งานาร่วมกันในหลายส่วนของระบบ เช่น การแสดงข้อความแจ้งเตือน การเข้าถึงกล้องถ่ายรูป และคลังรูปภาพ

5. lib/constant

ใช้สำหรับเก็บค่าคงที่ของระบบ เช่น url ของเซิร์ฟเวอร์ ค่าพารามิเตอร์การตั้งค่า และข้อความที่ใช้ซ้ำภายในแอปพลิเคชัน

6. lib/localization

เป็นส่วนที่เกี่ยวข้องกับการจัดการภาษาภายในแอปพลิเคชัน โดยแอปพลิเคชันนี้รองรับการใช้งานทั้งภาษาไทย และภาษาอังกฤษ

7. lib/widgets

เป็นส่วนที่เก็บคอมโพเนนต์ส่วนติดต่อกับผู้ใช้ที่ใช้ในหลายฟีเจอร์ของแอปพลิเคชัน เช่น ปุ่มตกลง หน้าแสดงข้อมูล ส่วนแสดงข้อความแจ้งเตือน เป็นต้น

ไลบรารี และเครื่องมือที่ใช้ในการพัฒนา

เพื่อให้แอปพลิเคชันสามารถทำงานได้ครบถ้วนตามวัตถุประสงค์ที่กำหนดไว้ จึงมีการเลือกใช้ไลบรารี และเครื่องมือภายนอกที่เหมาะสม เพื่อสนับสนุนการทำงานของแอปพลิเคชัน ดังนี้

camera ใช้สำหรับการเข้าถึงกล้องของสมาร์ทโฟน และบันทึกภาพแบบเรียลไทม์ เพื่อนำภาพไปใช้ในการตรวจจับวัตถุ การตรวจจับมาร์กเกอร์ความเป็นจริงเสริม

image_picker ใช้สำหรับการเลือกภาพจากคลังรูปภาพภายในอุปกรณ์สมาร์ทโฟน เพื่อใช้ในการตรวจจับวัตถุ และการตรวจจับอักขระด้วยแสง

dio ใช้สำหรับการสื่อสารระหว่างแอปพลิเคชันบนสมาร์ทโฟน และระบบฝั่งเซิร์ฟเวอร์ผ่าน การรับส่งข้อมูลในรูปแบบ HTTP และ REST API โดยรองรับการส่งข้อมูลภาพ ข้อมูลฟอร์ม และการรองรับผลลัพธ์การประมวลผลกลับมาในรูปแบบ JSON ไลบรารีตัวนี้ช่วยในการจัดการคำขอ (Request) การควบคุมเวลาการเชื่อมต่อ และการจัดการข้อผิดพลาดเป็นไปอย่างมีประสิทธิภาพ

permission_handler ใช้สำหรับจัดการการขอลิขสิทธิ์การเข้าถึงอุปกรณ์ของสมาร์ทโฟน เช่น กล้อง ไมโครโฟน และพื้นที่จัดเก็บข้อมูล

speech_to_text / flutter_tts ใช้สำหรับการสั่งการแอปพลิเคชันด้วยเสียง โดยที่ speech_to_text ใช้สำหรับแปลงเสียงพูดของผู้ใช้งานเป็นข้อความ เพื่อนำมาเป็นคำสั่งในการควบคุมการทำงาน

ของแอปพลิเคชัน หลังจากนั้นก็จะใช้ flutter_tts ในการแปลงข้อความเป็นเสียงเพื่อแจ้งเตือนผลลัพธ์การทำงานของระบบแก่ผู้ใช้งาน

firebase_analytics ใช้สำหรับเก็บข้อมูลเชิงสถิติ และพฤติกรรมของการใช้งานของผู้ใช้ในแต่ละฟีเจอร์ของแอปพลิเคชัน ข้อมูลที่ได้สามารถนำไปใช้ในการวิเคราะห์พฤติกรรมผู้ใช้งาน และใช้เป็นข้อมูลสนับสนุนในการปรับปรุงประสิทธิภาพของแอปพลิเคชันในอนาคต

firebase_crashlytics ใช้สำหรับตรวจสอบ บั๊ก และรายงานความผิดพลาดที่เกิดขึ้นระหว่างการทำงานของแอปพลิเคชันบนอุปกรณ์จริง โดยช่วยให้ผู้พัฒนาสามารถติดตามสาเหตุของปัญหา วิเคราะห์ข้อผิดพลาด และดำเนินการแก้ไขได้อย่างรวดเร็ว



ภาคผนวก ค.
ระบบฝั่งเซิร์ฟเวอร์ (Backend System)

TNI

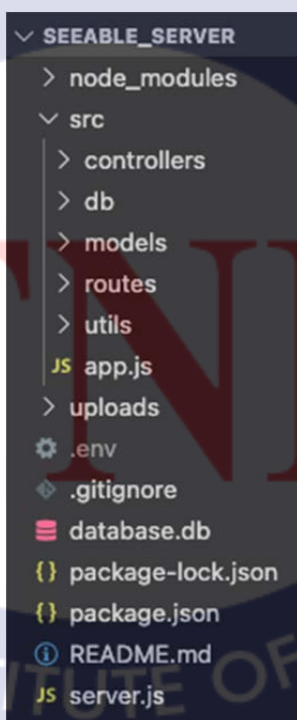
ระบบฝั่งเซิร์ฟเวอร์ (Backend System)

ระบบฝั่งเซิร์ฟเวอร์ของแอปพลิเคชัน Seeable ถูกพัฒนาด้วยภาษา JavaScript โดยใช้สภาพแวดล้อมการทำงานของ Node.js และเฟรมเวิร์ค Express.js ทำหน้าที่เป็นศูนย์กลางในการรับคำร้องขอจากแอปพลิเคชันบนสมาร์ตโฟน ประมวลผลข้อมูล และประสานงานกับโมดูลประมวลผลภาพ และปัญญาประดิษฐ์ รวมถึงการจัดการข้อมูลผู้ใช้งาน และมาร์กเกอร์ความเป็นจริงเสริมภายในระบบ

โครงสร้างของระบบฝั่งเซิร์ฟเวอร์ถูกออกแบบในลักษณะเชิงโมดูล เพื่อแยกความรับผิดชอบของแต่ละส่วนอย่างชัดเจน โดยสามารถแบ่งองค์ประกอบหลักตามโครงสร้าง ดังนี้

โครงสร้างของระบบฝั่งเซิร์ฟเวอร์

ระบบฝั่งเซิร์ฟเวอร์มีการจัดโครงสร้างโดยแบ่งออกเป็นส่วนควบคุมการทำงาน (Controllers) ส่วนจัดการข้อมูล (Models) และส่วนกำหนดเส้นทางการเรียกใช้งาน (Routes) ซึ่งเป็นโครงสร้างหลักในการควบคุมการทำงานของระบบ นอกจากนี้ยังมีส่วนประกอบย่อย เช่น db ใช้สำหรับการจัดการการเชื่อมต่อ และการเข้าถึงฐานข้อมูล Utils ใช้สำหรับฟังก์ชันช่วยเหลือที่ใช้ร่วมกันภายในระบบ เพื่อให้โครงสร้างของระบบมีความชัดเจน เป็นระเบียบ และเอื้อต่อการดูแลรักษา และขยายระบบในอนาคต



Controllers

โพลเดอร์ Controllers ทำหน้าที่เป็นส่วนควบคุมการทำงานของระบบฝั่งเซิร์ฟเวอร์ โดยรับคำร้องขอ (Request) จากฝั่งผู้ใช้งานผ่าน API และดำเนินการประมวลผลตามฟังก์ชันที่ร้องขอ ก่อนส่งผลลัพธ์กลับไปยังผู้ใช้งาน ไฟล์ในส่วนนี้ครอบคลุมการนำทาง การตรวจจับสิ่งกีดขวาง การตรวจจับวัตถุ การตรวจจับอักขระด้วยแสง และการจัดการข้อมูลผู้ใช้งาน โดยประกอบด้วย

1. arMarker.controller.js ใช้จัดการข้อมูลของมาร์กเกอร์ความเป็นจริงเสริม เช่น การจัดการข้อมูลมาร์กเกอร์ การดึงข้อมูล และค้นหามาร์กเกอร์จากฐานข้อมูล รวมถึงการกำหนดเส้นทางในการนำทาง (Path Planning) การอัปเดตตำแหน่งเพื่อคำนวณหาระยะทางที่เหลือ
2. objectDetect.controller.js/obstacleWarning.controller.js/ocr.controller.js ทำหน้าที่รับภาพจากแอปพลิเคชัน และส่งต่อไปยังโมดูลปัญญาประดิษฐ์เพื่อให้ประมวลผล แล้วนำผลลัพธ์ที่ได้ส่งกลับไปยังแอปพลิเคชันเพื่อแจ้งเตือนผู้ใช้งาน
3. login.controller.js/register.controller.js/forget.controller.js ใช้จัดการกระบวนการยืนยันตัวตนของผู้ใช้งาน ได้แก่ การลงทะเบียน การลงชื่อเข้าใช้ และการเปลี่ยนรหัสผ่าน
4. contactDev.controller.js ใช้สำหรับรับข้อมูลการติดต่อหรือข้อเสนอแนะจากผู้ใช้งาน ส่งถึงผู้พัฒนาแอปพลิเคชัน

Models

โพลเดอร์ Models ทำหน้าที่จัดการโครงสร้างข้อมูล และการติดต่อกับฐานข้อมูล SQLite โดยแยกข้อมูลออกจากส่วนควบคุมการทำงาน เพื่อให้ระบบมีความเป็นระเบียบเรียบร้อย และดูแลรักษาได้ง่าย ประกอบด้วย

1. user.model.js ใช้จัดเก็บ และจัดการข้อมูลผู้ใช้งาน ที่เกี่ยวข้องกับการการเข้าสู่ระบบ
2. arMarkers.model.js ใช้จัดเก็บ และจัดการข้อมูลมาร์กเกอร์ความเป็นจริงเสริม ซึ่งถูกนำมาใช้เป็นจุดอ้างอิงเชิงตำแหน่งภายในอาคาร

Routes

โพลเดอร์ Routes ทำหน้าที่กำหนดเส้นทางการเรียกใช้งาน API (API Endpoints) ของระบบ โดยเชื่อมโยงคำร้องขอจากแอปพลิเคชันไปยัง Controller ที่เกี่ยวข้อง ตัวอย่างไฟล์ในส่วนนี้ ได้แก่

1. arMarkers.route.js
2. objectDetect.route.js
3. obstacleWarning.route.js
4. ocr.route.js

5. login.route.js/register.route.js

6. contactDev.route.js

การแยก Routes ออกจาก Controllers ช่วยให้โครงสร้างของระบบมีความชัดเจน อ่านง่าย และสามารถขยายระบบในอนาคตได้สะดวก

ส่วนประกอบอื่นๆ ของระบบ

1. app.js เป็นไฟล์หลักสำหรับกำหนดค่าพื้นฐาน Express.js เช่น การตั้งค่า Middleware การจัดการคำร้องขอ (Request) และการเชื่อมโยงเส้นทางการเรียกใช้งาน (routes) ต่างๆ เข้ากับระบบฝั่งเซิร์ฟเวอร์

2. server.js ใช้สำหรับเริ่มต้นการทำงานของเซิร์ฟเวอร์ โดยทำหน้าที่กำหนดพอร์ตที่ใช้ในการให้บริการ และเรียกใช้งาน app.js เพื่อเปิดให้ระบบสามารถรับคำร้องขอจากแอปพลิเคชันได้

3. โฟลเดอร์ Uploads ใช้สำหรับจัดเก็บไฟล์ชั่วคราวที่ได้รับจากแอปพลิเคชัน เช่น ภาพจากกล้องสมาร์ทโฟน ก่อนส่งต่อไปยังโมดูลประมวลผลภาพ และปัญญาประดิษฐ์ เมื่อการประมวลผลเสร็จสิ้นระบบจะทำการลบไฟล์ดังกล่าวเพื่อลดการใช้พื้นที่จัดเก็บข้อมูลบนเซิร์ฟเวอร์

4. database.db เป็นฐานข้อมูลแบบ SQLite ที่ใช้จัดเก็บข้อมูลที่จำเป็นต่อการทำงานของระบบ เช่น ข้อมูลผู้ใช้งาน และข้อมูลมาร์กเกอร์ความเป็นจริงเสริม

ภาคผนวก ง.

ระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Image Processing and Artificial Intelligence (AI) Module)

TNI

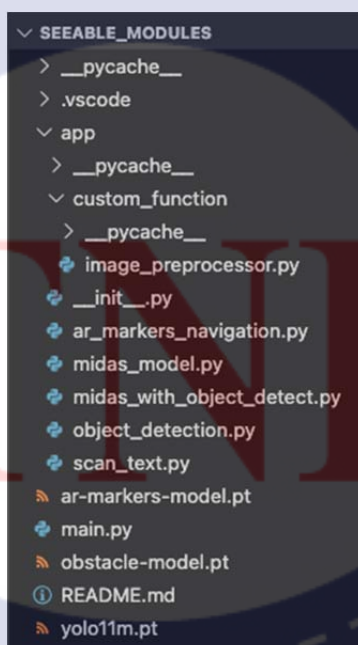
ระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Image Processing and Artificial Intelligence (AI) Module)

ระบบประมวลผลภาพ และปัญญาประดิษฐ์ของแอปพลิเคชัน Seeable ถูกพัฒนาด้วยภาษา Python และทำงานในลักษณะโมดูลแยกออกจากเซิร์ฟเวอร์หลัก โดยใช้เฟรมเวิร์ค Flask สำหรับการประมวลผลผ่าน API โดยที่โมดูลนี้มีหน้าที่รับภาพจากระบบฝังเซิร์ฟเวอร์ ทำการประมวลผล แล้วส่งผลลัพธ์กลับในรูปแบบข้อมูล JSON เพื่อใช้ในการนำทาง และการแจ้งเตือนผู้ใช้งานแบบเรียลไทม์

โครงสร้างของระบบถูกออกแบบในลักษณะเชิงโมดูล เพื่อแยกหน้าที่ของการประมวลผลแต่ละประเภทออกจากกันอย่างชัดเจน และเอื้อต่อการพัฒนา ปรับปรุง หรือเปลี่ยนแปลงในอนาคต

โครงสร้างของระบบประมวลผลภาพ และปัญญาประดิษฐ์

ระบบประมวลผลภาพ และปัญญาประดิษฐ์ถูกจัดอยู่ภายใต้โฟลเดอร์หลัก และโฟลเดอร์ app ภายในประกอบด้วยไฟล์ และโมดูลที่ใช้ในการประมวลผลภาพแต่ละประเภท โดยมีไฟล์ `_init_.py` ทำหน้าที่กำหนดให้โฟลเดอร์นี้ถูกมองเป็นหนึ่งใน Python Package เพื่อให้สามารถเรียกใช้งานโมดูลนี้ภายในระบบได้อย่างถูกต้อง และช่วยในการจัดการไฟล์อย่างเป็นระบบ



main.py

ไฟล์นี้ทำหน้าที่เป็นตัวสร้าง Flask Application ขึ้นมาและกำหนด API Endpoint สำหรับรับภาพจากระบบเซิร์ฟเวอร์เมื่อได้รับคำร้องขอ ระบบจะทำการเรียกใช้งานโมดูลการประมวลผลที่เกี่ยวข้องตามประเภทของงาน จากนั้นส่งผลลัพธ์กลับไปยังระบบฝั่งเซิร์ฟเวอร์ในรูปแบบข้อมูล JSON

object_detection.py

ไฟล์นี้ใช้สำหรับการตรวจจับวัตถุทั่วไปจากภาพ โดยอาศัย YOLOv11m ที่ผ่านการฝึกด้วยชุดข้อมูล COCO Dataset ซึ่งสามารถตรวจจับวัตถุพื้นฐานได้หลายประเภท เช่น คน โต๊ะ เก้าอี้ ประตู และวัตถุทั่วไปที่พบได้ในชีวิตประจำวัน

midas_with_object_detect.py

ไฟล์นี้ใช้สำหรับการตรวจจับสิ่งกีดขวาง ซึ่งเป็นการผสานผลลัพธ์จากการตรวจจับวัตถุ และการประมาณค่าระยะลึกเข้าด้วยกัน โดยอาศัย YOLOv8n ที่ผ่านการฝึกด้วยชุดข้อมูลที่จัดเก็บจากสภาพแวดล้อมจริงภายในอาคาร C ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

ar_markers_navigation.py

ไฟล์นี้ใช้สำหรับการตรวจจับมาร์กเกอร์ความเป็นจริงเสริมจากภาพที่ได้รับ โดยทำหน้าที่ระบุรหัสของมาร์กเกอร์ที่ตรวจพบ โดยอาศัย YOLOv8n ที่ผ่านการฝึกด้วยชุดข้อมูลมาร์กเกอร์ความเป็นจริงเสริมในมุมมองต่างๆ

scan_text.py

ไฟล์นี้ใช้สำหรับการตรวจจับอักขระด้วยแสง โดยอาศัยไลบรารี pytesseact เพื่อแปลงข้อความที่ปรากฏในภาพให้เป็นข้อความดิจิทัล

ชุดข้อมูลสำหรับการตรวจจับสิ่งกีดขวาง และการตรวจจับมาร์กเกอร์ความเป็นจริงเสริม

การพัฒนาระบบตรวจจับสิ่งกีดขวาง และการระบุตำแหน่งด้วยมาร์กเกอร์ความเป็นจริงเสริมในงานวิจัยนี้ จำเป็นต้องอาศัยข้อมูลที่สอดคล้องกับสภาพแวดล้อมจริงของพื้นที่ใช้งาน เพื่อให้แบบจำลองปัญญาประดิษฐ์สามารถเรียนรู้ได้อย่างเหมาะสม งานวิจัยนี้จึงได้จัดเตรียม และสร้างชุดข้อมูลนี้ขึ้นจากสภาพแวดล้อมจริงภายในอาคาร C ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น และบริหารจัดการชุดข้อมูลผ่านแพลตฟอร์ม Roboflow

ชุดข้อมูลสำหรับการตรวจจับสิ่งกีดขวาง

ชุดข้อมูลสำหรับการตรวจจับสิ่งกีดขวางถูกจัดเก็บจากสภาพแวดล้อมจริงภายในอาคาร C ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น โดยที่ถ่ายในสภาพแวดล้อมที่หลากหลาย เพื่อให้ครอบคลุมลักษณะการใช้งานจริงของผู้พิการทางสายตา ภาพที่ได้ถูกนำมาทำการกำหนดกรอบวัตถุ (Bounding Box Annotation) และจำแนกประเภทของสิ่งกีดขวาง ชุดข้อมูลดังกล่าวประกอบด้วยวัตถุที่พบได้บ่อยภายในอาคาร ซึ่งอาจส่งผลกระทบต่อความปลอดภัยในการเคลื่อนที่ของผู้ใช้งาน ได้แก่ คน เก้าอี้ โต๊ะ ตู้ ถังขยะ ประตู บันได หน้าต่าง ถังดับเพลิง ไม้เท้าขาว ลิฟท์ ป้ายสัญลักษณ์ห้องน้ำ ป้ายทางหนีไฟ และสิ่งกีดขวางทั่วไปสามารถเข้าถึงชุดข้อมูลได้ผ่านแพลตฟอร์ม Roboflow ที่ลิงก์: <https://app.roboflow.com/seeable/tni-indoor/>

ชุดข้อมูลสำหรับการตรวจจับมาร์กเกอร์ความเป็นจริงเสริม

ชุดข้อมูลสำหรับการตรวจจับมาร์กเกอร์ความเป็นจริงเสริมถูกจัดเตรียมขึ้นเพื่อใช้ในการระบุตำแหน่งผู้ใช้งานภายในอาคาร โดยมาร์กเกอร์ถูกติดตั้งในตำแหน่งห้องต่างๆ ภายในอาคาร C ภาพของมาร์กเกอร์ถูกถ่ายในมุมมอง ระยะห่าง และสภาพแสงที่แตกต่างกัน เพื่อเพิ่มความแม่นยำในการตรวจจับมาร์กเกอร์ในสถานการณ์ต่างๆ สามารถเข้าถึงชุดข้อมูลนี้ได้ผ่านแพลตฟอร์ม Roboflow ที่ลิงก์: <https://app.roboflow.com/seeable/tni-ar-markers/>

ภาคผนวก จ.
ฐานข้อมูล (Database)

TNI

ฐานข้อมูล (Database)

ระบบ Seeable ใช้ฐานข้อมูลแบบ SQLite ฐานข้อมูลประกอบด้วยตารางสำคัญ 2 ตาราง ได้แก่ ตารางผู้ใช้งาน (users) ใช้จัดเก็บข้อมูลบัญชีผู้ใช้งาน เพื่อรองรับการลงทะเบียน การเข้าสู่ระบบ และการจัดการบัญชี โดยประกอบด้วยข้อมูลสำคัญ เช่น รหัสผู้ใช้งาน (id) ชื่อ (firstname) นามสกุล (lastname) ชื่อบัญชีผู้ใช้ (username) รหัสผ่าน (password) และอีเมล (email)

	id	firstname	lastname	username	password	email
...	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Preawpan	Siriphalangkanont	rung	password	test@gmail.com

ตารางมาร์กเกอร์ความเป็นจริงเสริม (markers) ใช้จัดเก็บข้อมูลมาร์กเกอร์ความเป็นจริงเสริม เช่น รหัสมาร์กเกอร์ (marker_id) และชื่อมาร์กเกอร์ (marker_name) ซึ่งถูกนำมาใช้เป็นจุดอ้างอิงเชิงตำแหน่งในการระบุตำแหน่งของผู้ใช้งาน และการนำทางภายในอาคาร

	id	marker_id	marker_name
F...	Filter	Filter	
1	1	1	C301-F
2	2	2	C301-B
3	3	3	C302-F
4	4	4	C302-B
5	5	5	C303-F
6	6	6	C303-B
7	7	7	C304-F
8	8	8	C304-B
9	9	9	C305-F
10	10	10	C305-B
11	11	11	C306-F
12	12	12	C306-B
13	13	13	C307-F
14	14	14	C307-B
15	15	15	C314-F

ภาคผนวก จ.

การเข้าถึงซอร์สโค้ด และการรันระบบ (Source Code Access and Execution)

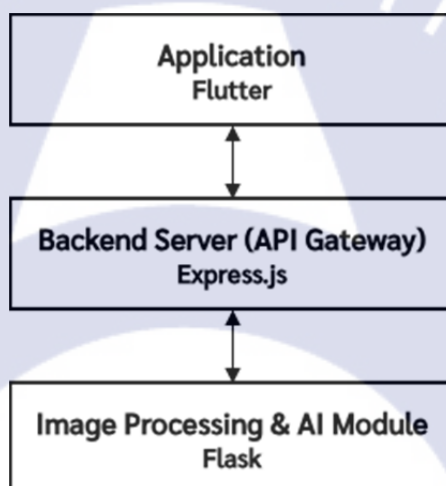
TNI

การเข้าถึงซอร์สโค้ด และการรันระบบ (Source Code Access and Execution)

แอปพลิเคชัน Seeable ถูกพัฒนาในรูปแบบสถาปัตยกรรมแยกส่วน โดยแบ่งการทำงานออกเป็น ฝั่งแอปพลิเคชันบนสมาร์ตโฟน ฝั่งระบบประมวลผลภาพ และปัญญาประดิษฐ์ และฝั่งระบบเซิร์ฟเวอร์ โดยซอร์สโค้ดทั้งหมดถูกจัดเก็บไว้บนระบบจัดการเวอร์ชัน Git เพื่อให้สามารถติดตามการเปลี่ยนแปลง และทำงานร่วมกันได้อย่างเป็นระบบ

ภาพรวมสถาปัตยกรรมของระบบ

ระบบ Seeable ใช้สถาปัตยกรรม Client-Server ร่วมกับโมดูลปัญญาประดิษฐ์ โดยมีการเชื่อมต่อการทำงานระหว่างส่วนต่างๆ ดังนี้



แอปพลิเคชันบนสมาร์ตโฟนทำหน้าที่รับภาพจากกล้อง และส่งข้อมูลไปยังระบบฝั่งเซิร์ฟเวอร์ผ่าน API Gateway จากนั้นระบบ Flask จะประมวลผลภาพด้วยโมเดลปัญญาประดิษฐ์ และส่งผลลัพธ์กลับมายังแอปพลิเคชันเพื่อแสดงผล และแจ้งเตือนผู้ใช้งานในรูปแบบเสียง

ซอร์สโค้ดของระบบ

ซอร์สโค้ดของระบบถูกแยกออกเป็น 3 ส่วนหลัก ดังนี้

1. แอปพลิเคชันบนสมาร์ตโฟน (Flutter)

<https://github.com/rungpreawpan/seeable-application-for-blind>

2. ระบบเซิร์ฟเวอร์ (Express.js)

https://github.com/rungpreawpan/seeable_server

3. ระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Flask)

<https://github.com/rungpreawpan/seeable-script>

ขั้นตอนรันระบบ

เพื่อให้ระบบ Seeable สามารถทำงานได้อย่างสมบูรณ์ จำเป็นต้องเริ่มต้นการทำงานของทั้ง 3 ส่วน ตามลำดับ ดังนี้

1. การรันระบบประมวลผลภาพ และปัญญาประดิษฐ์ (Flask)

ทำการโคลนซอร์สโค้ดจาก Github

```
git clone https://github.com/rungpreawpan/seeable-script
```

ติดตั้งไลบรารีที่จำเป็น และเริ่มต้นระบบด้วยคำสั่ง

```
python main.py
```

ระบบ Flask จะทำการทำงานที่พอร์ต 5001

2. การรันระบบเซิร์ฟเวอร์ (Express.js)

ทำการโคลนซอร์สโค้ดจาก Github

```
git clone https://github.com/rungpreawpan/seeable-server
```

ติดตั้งไลบรารีที่จำเป็น และเริ่มต้นระบบด้วยคำสั่ง

```
npm install
```

```
npm run dev
```

ระบบ Express จะทำงานที่พอร์ต 3000

3. การรันแอปพลิเคชัน (Flutter)

ทำการโคลนซอร์สโค้ดจาก Github

```
https://github.com/rungpreawpan/seeable-application-for-blind
```

ติดตั้งไลบรารีที่จำเป็น และเริ่มต้นระบบด้วยคำสั่ง

ทำการเข้าโฟลเดอร์ lib/constant แล้วเข้าไปที่ไฟล์ environment.dart เช็คว่าแปร baseURL ว่ามี url ตรงกับเซิร์ฟเวอร์ Express.js หรือไม่ถ้าตรงกันก็สามารถใช้งานเชื่อมต่อกับระบบเซิร์ฟเวอร์ได้หลังจากนั้นให้รันคำสั่งในการติดตั้งไลบรารี และรันแอปพลิเคชัน

```
flutter pub get
```

```
flutter run
```