

ระบบต้นแบบค้นหาสินค้าและระบุเส้นทางด้วยภาพ

VISUAL SEARCH



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VISUAL SEARCH



NARATCHA SATHIENPAKIRANAKORN

NIRATTISAI SATHU

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL

FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเอกสารทุกครั้งที่มีการนำไปใช้

ACADEMIC YEAR 2022



หัวข้อสหกิจศึกษา	ระบบต้นแบบค้นหาสินค้าและระบุเส้นทางด้วยภาพ
ชื่อนักศึกษา	นาย นรัชฌา เสถียรปภิรณกร รหัสนักศึกษา 62050181 นาย นิรัตติย สาธู รหัสนักศึกษา 62050187
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ดร.ประพจน์ ศรีนัฐติวงศ์

## บทคัดย่อ

การปฏิบัติสหกิจในครั้งนี้ คณะผู้จัดทำได้ทำการปฏิบัติสหกิจศึกษา ณ บริษัท โทคมนาคม แห่งชาติ จำกัด (มหาชน) ในตำแหน่ง Research Assistant ซึ่งโครงการวิจัยนี้มีวัตถุประสงค์เพื่อสร้างระบบต้นแบบ Client-Server ที่มีการใช้งานเทคโนโลยี DPU ของ Xilinx ที่มีการปรับแต่งสำหรับการเร่งความเร็วการประมวลผลโมเดลปัญญาประดิษฐ์สำหรับการค้นหาสินค้า และระบุตำแหน่งในห้างสรรพสินค้าด้วยภาพ ในการจัดทำโครงการวิจัยนี้ แบ่งออกเป็น 3 ส่วนได้แก่ การพัฒนาระบบ Client ในรูปแบบ Mobile Application ด้วย React Native การพัฒนาระบบ Web Server ในรูปแบบ REST API ด้วย Node.JS และการพัฒนาโมเดลในการตรวจจับวัตถุด้วยโครงสร้าง CNN ที่มีชื่อว่า YOLOv4 ซึ่งฝึกสอนโดยใช้ Darknet Framework เริ่มตั้งแต่ขั้นตอนการเก็บรวบรวมข้อมูลการฝึกสอนบน GPU และปรับแต่งโมเดลเพื่อใช้งานกับเทคโนโลยี DPU ของ Xilinx โดยสร้างแอปพลิเคชัน Python สำหรับการประมวลผลการตรวจจับวัตถุและยังสามารถเรียกใช้งาน Library high-level API ใน Vitis-AI Development Platform เพื่อการใช้งานเทคโนโลยี DPU ของ Xilinx

**คำสำคัญ :** DPU, CNN, Object Detection, YOLOv4, Darknet Framework, Vitis-AI;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Title</b>	VISUAL SEARCH
<b>Students</b>	Mister Naratcha Sathienpakiranakorn Student ID 62050181 Mister Nirattisai Sathu Student ID 62050187
<b>Degree</b>	Bachelor of Science (Computer Science)
<b>Department</b>	Computer Science
<b>School</b>	Science
<b>University</b>	King Mongkut's Institute of Technology Ladkrabang (KMITL)
<b>Academic Year</b>	2022
<b>Advisor</b>	Dr. Prapoj Srinuwattiwong

### Abstract

This cooperative education experience involved the researcher serving as a Research Assistant at National Telecom Public Company Limited, with the objective of designing and developing a prototype Client-Server model that utilizes Xilinx's DPU technology for optimized acceleration of CNN deep learning workloads. The purpose of the prototype was to facilitate product searching and identifying the location of users within a shopping mall through Object Detection. The preparation of this research project was divided into three parts: developing a client system in the form of a mobile application using React Native, developing a web server in the form of a REST API with Node.JS, and developing an object detection model with a YOLOv4 CNN architecture trained using the Darknet Framework. The model was trained by custom training data and optimizing it for use with Xilinx's DPU technology. Python applications were created for object detection processing and used a high-level API in Vitis-AI Development Platform for implementing Xilinx DPU technology.

**Keywords:** DPU, CNN, Object Detection, YOLOv4, Darknet Framework, Vitis-AI;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการสหกิจศึกษานี้ได้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากความกรุณาและความร่วมมือของทุก ๆ ท่านที่มีส่วนร่วมในการให้คำแนะนำ ให้ความช่วยเหลือ และการสนับสนุนตลอดมา ขอขอบพระคุณ ดร.ประพจน์ ศรีนิวัตติวงศ์ ที่คอยให้คำปรึกษา และแนะนำเพื่อปรับปรุงในส่วนที่บกพร่องเพื่อให้การดำเนินงานในโครงการสหกิจศึกษานี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณ ดร.สมรักษ์ เพชรชาตรี ที่คอยให้คำปรึกษาตลอดทั้งการช่วยเหลือ ดูแลในระหว่างปฏิบัติงาน ให้คำแนะนำในการใช้เทคโนโลยีและซอฟต์แวร์ต่าง ๆ อีกทั้งให้แนวคิดใน การศึกษา พัฒนาระบบใช้งานเทคโนโลยีใหม่ที่ยังไม่ค่อยเป็นที่รู้จัก และชี้แนะ แนวทางในการทำสหกิจศึกษาใน ครั้งนี้จนสำเร็จลุล่วง

ขอขอบพระคุณบริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) ที่ให้โอกาสในการศึกษาปฏิบัติงานวิจัยในศูนย์วิจัยและพัฒนา Brownien Lab ของบริษัท คอยสนับสนุนอุปกรณ์เทคโนโลยีและซอฟต์แวร์ต่างๆ สำหรับการปฏิบัติงานในโครงการสหกิจศึกษานี้

ขอขอบพระคุณ คณาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ ทุกท่านที่ได้ถ่ายทอดความรู้และอบรมสั่งสอนทั้งในภาคทฤษฎีและภาคปฏิบัติมาตลอดระยะเวลา 4 ปี

ขอขอบพระคุณ เพื่อน ๆ พี่ ๆ ภาควิชาวิทยาการคอมพิวเตอร์ที่คอยแนะนำการทำงานสหกิจศึกษา ชี้แนะข้อบกพร่องที่ควรพัฒนา และคอยให้กำลังใจอยู่เสมอ รวมถึงบุคคลอื่นๆ ที่มีส่วนในงานสหกิจครั้งนี้ ที่ไม่ได้กล่าวมา ผู้จัดทำโครงการขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

นรัชมา เสถียรปัทมกร

นิรัตน์ ศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

บทคัดย่อ .....	ก
Abstract .....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ .....	ง
สารบัญรูป .....	ฉ
สารบัญตาราง .....	ณ
บทที่ 1 บทนำ.....	1
1.1. ความเป็นมาและความสำคัญของปัญหา.....	1
1.2. วัตถุประสงค์ของโครงการ .....	2
1.3. เป้าหมายของโครงการ .....	2
1.4. ขอบเขตของโครงการ.....	2
1.5. ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6. อุปกรณ์และเครื่องมือที่ใช้.....	3
1.7. เงื่อนไข/ข้อจำกัด.....	6
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	7
2.1. การพัฒนา Mobile Application.....	7
2.2. การพัฒนา API.....	9
2.3. Object Detection.....	13
2.4. การพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการทำ Object Detection.....	14
2.5. เทคโนโลยี Data Center accelerator ของ Xilinx.....	16
2.6. การพัฒนาและใช้งาน Model AI บนการ์ด Data Center accelerator Cards.....	17
บทที่ 3 วิธีการดำเนินงานวิจัย.....	25
3.1. แผนการดำเนินการ .....	25

เอกสารนี้เป็นเอกสารลิขสิทธิ์ในชื่อโครงการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ท่านไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.	ขั้นตอนการศึกษาและรวบรวมข้อมูลของระบบ .....	28
3.3.	ขั้นตอนการวิเคราะห์ระบบ .....	30
3.4.	ขั้นตอนการออกแบบระบบและขั้นตอนการดำเนินงาน.....	34
3.5.	การออกแบบจำลองหน้าจอระบบ.....	60
3.6.	การฝึกสอนโมเดลปัญญาประดิษฐ์และปรับแต่งเป็น Fix-point โมเดล.....	71
บทที่ 4	ผลการวิจัยและการอภิปรายผล .....	78
4.1.	Mobile Application .....	78
4.2.	Server-side.....	90
4.3.	Python Application.....	105
4.4.	ผลการดำเนินการทดสอบ Test Case.....	125
4.5.	Traceability matrix.....	132
บทที่ 5	สรุปผลการวิจัยและข้อเสนอ .....	133
5.1.	สรุปผลการดำเนินงาน.....	133
5.2.	ปัญหาที่พบเจอ.....	134
5.3.	ข้อเสนอแนะ.....	134
เอกสารอ้างอิง.....		136

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่ 2.1	โครงสร้าง React Native .....	8
รูปที่ 2.2	ตัวอย่างโครงสร้างสถาปัตยกรรม RESTful API.....	10
รูปที่ 2.3	ภาพอธิบายประวัติศาสตร์การ Deployment .....	11
รูปที่ 2.4	ภาพ Classification และ Objection detection .....	13
รูปที่ 2.5	ภาพตัวอย่างโครงสร้าง Convolutional Neural Network ของ VGG16 .....	15
รูปที่ 2.6	Xilinx Alveo U50 Data Center accelerator cards .....	17
รูปที่ 2.7	Vitis-AI Development Platform .....	18
รูปที่ 2.8	Development Flow.....	24
รูปที่ 3.1	กราฟแท่งแสดงแผนดำเนินงานวิจัยแบบย่อ .....	26
รูปที่ 3.2	สถาปัตยกรรมของระบบ Visual Search.....	34
รูปที่ 3.3	Use Case Diagram .....	35
รูปที่ 3.4	แผนภาพแสดงกระบวนการค้นหาสินค้าด้วยรูปภาพ .....	40
รูปที่ 3.5	แผนภาพแสดงกระบวนการระบุตำแหน่งร้านค้าด้วยรูปภาพ .....	40
รูปที่ 3.6	แผนภาพแสดงกระบวนการค้นหาสินค้า.....	41
รูปที่ 3.7	แผนภาพแสดงกระบวนการค้นหาร้านค้า.....	41
รูปที่ 3.8	แผนภาพแสดงกระบวนการจัดการข้อมูลในฐานข้อมูล.....	41
รูปที่ 3.9	แผนภาพ Context Diagram.....	42
รูปที่ 3.10	แผนภาพ Data Flow Diagram Level 1 .....	43
รูปที่ 3.11	แผนภาพ Data Flow Diagram Level 2-1.....	44
รูปที่ 3.12	แผนภาพ Data Flow Diagram Level 2-3.....	44
รูปที่ 3.13	แผนภาพ Data Flow Diagram Level 2-4.....	45
รูปที่ 3.14	แผนภาพ Data Flow Diagram Level 2-5.....	45
รูปที่ 3.15	Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยรูปภาพ .....	49
รูปที่ 3.16	Sequence Diagram แสดงรายละเอียดการแสดงผลเส้นทางไปยังร้านค้าปลายทาง .....	51
รูปที่ 3.17	Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยชื่อ.....	53
รูปที่ 3.18	Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยชื่อ .....	54
รูปที่ 3.19	Sequence Diagram แสดงรายละเอียดจัดการข้อมูลร้านค้า.....	55
รูปที่ 3.20	Sequence Diagram แสดงรายละเอียดจัดการข้อมูลสินค้า .....	56
รูปที่ 3.21	Entity Relations Diagram แสดงโครงสร้างของฐานข้อมูล.....	57
รูปที่ 3.22	Mockup: Search Product with Image Flow.....	60
รูปที่ 3.23	Mockup: Search Store Flow .....	65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
 3.23 Mockup: Search Store Flow .....

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.24 Process Flow Python Application .....	70
รูปที่ 3.25 ตัวอย่างข้อมูลหน้าร้านค้า.....	71
รูปที่ 3.26 ตัวอย่างการ Annotation รูปภาพ.....	72
รูปที่ 3.27 ตัวอย่างการปรับแต่งรูปภาพ .....	73
รูปที่ 3.28 ตัวอย่างไฟล์รูปแบบ Label ของ Darkent .....	73
รูปที่ 3.29 ตัวอย่างการตั้งค่าไฟล์ data ของ Darkent.....	75
รูปที่ 3.30 ตัวอย่างการตั้งค่า option การ Quantization .....	76
รูปที่ 3.31 ตัวอย่างการตั้งค่า option การ Compile .....	77
รูปที่ 4.1 Apps icon .....	78
รูปที่ 4.2 Search with Image Flow .....	79
รูปที่ 4.3 รูปหน้าจอหลักเมื่อเข้าสู่แอปพลิเคชัน .....	80
รูปที่ 4.4 รูปหน้าจอหลักเมื่อกด Floating button.....	80
รูปที่ 4.5 Camera Screen .....	81
รูปที่ 4.6 Gallery Screen.....	81
รูปที่ 4.7 Loading Screen .....	81
รูปที่ 4.8 List Screen .....	81
รูปที่ 4.9 Store Detail Flow .....	82
รูปที่ 4.10 ตัวอย่างรายละเอียดหน้าร้าน Hor IT.....	83
รูปที่ 4.11 รายละเอียดหน้าร้าน NSP Furniture.....	83
รูปที่ 4.12 รายละเอียดหน้าร้าน BaNANA.....	84
รูปที่ 4.13 รายละเอียดหน้าร้าน True.....	84
รูปที่ 4.14 รายละเอียดหน้าร้าน 24 INK.....	84
รูปที่ 4.15 รายละเอียดหน้าร้าน .....	84
รูปที่ 4.16 แผนผังแสดงตำแหน่งร้าน True.....	85
รูปที่ 4.17 แผนผังแสดงตำแหน่งร้าน True แบบขยาย.....	85
รูปที่ 4.18 Navigation System Flow.....	86
รูปที่ 4.19 Search Store Flow .....	87
รูปที่ 4.20 Search Product Flow.....	88
รูปที่ 4.21 Server Error.....	89
รูปที่ 4.22 Fail to detect .....	89
รูปที่ 4.23 โครงสร้างไฟล์ระบบเบื้องหลัง .....	90
รูปที่ 4.24 Plot Graph ที่ได้จากการฝึกสอน Model สินค้า โดย Plot จากค่า AVG loss.....	105
รูปที่ 4.25 รูปการประเมินผลค่าความแม่นยำค่าน้ำหนัก Model สินค้าที่ 1.....	107
รูปที่ 4.26 รูปการประเมินผลค่าความแม่นยำค่าน้ำหนัก Model สินค้าที่ 2.....	109
รูปที่ 4.27 รูปการประเมินผลค่าความแม่นยำค่าน้ำหนัก Model สินค้าที่ 3.....	111

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย

รูปที่ 4.28 Plot Graph ที่ได้จากการฝึกสอน Model ร้านค้า โดย Plot จากค่า AVG loss .....	113
รูปที่ 4.29 รูปการประเมินผลค่าความแม่นยำค่าน้ำหนัก Model ร้านค้า .....	114
รูปที่ 4.30 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า MR.D.I.Y.....	115
รูปที่ 4.31 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า Mezzo .....	115
รูปที่ 4.32 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า VIDA.....	116
รูปที่ 4.33 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า BaNANA .....	116
รูปที่ 4.34 ไฟล์ xmodel ของ Model สินค้า .....	117
รูปที่ 4.35 ไฟล์ xmodel ของ Model ร้านค้า.....	117
รูปที่ 4.36 ผลลัพธ์การทดสอบความเร็วการประมวลผลด้วย DPU.....	118
รูปที่ 4.37 Function การเรียงรูปภาพด้วยเวลา.....	119
รูปที่ 4.38 Function เรียกใช้งาน dpu_runner .....	120
รูปที่ 4.39 Function Post Process .....	120
รูปที่ 4.40 Function predict.....	121
รูปที่ 4.41 Function Pre-Process Image.....	122
รูปที่ 4.42 Function Run Inference (1).....	122
รูปที่ 4.43 Function Run Inference (2).....	123
รูปที่ 4.44 ตัวอย่างผลลัพธ์การประมวลผลด้วย Model ตรวจจับ Product.....	123
รูปที่ 4.45 ตัวอย่างผลลัพธ์การประมวลผลด้วย Model ตรวจจับ Store.....	124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่ 2.1	ตารางแสดงตระกูล DPU ต่างๆ	19
ตารางที่ 2.2	ตารางแสดงข้อมูลของ Xilinx Alveo U50LV	19
ตารางที่ 2.3	ตัวดำเนินการที่รองรับ	20
ตารางที่ 3.1	ตารางแสดงแผนการดำเนินงานวิจัยแบบละเอียด	26
ตารางที่ 3.2	ตารางแสดงคำบรรยาย Use Case ค้นหาข้อมูลสินค้าด้วยรูปภาพ	35
ตารางที่ 3.3	ตารางแสดงคำบรรยาย Use Case ค้นหาข้อมูลสินค้าหรือร้านค้าด้วยชื่อ	36
ตารางที่ 3.4	ตารางแสดงคำบรรยาย Use Case ระบุตำแหน่งที่อยู่ผู้ใช้งาน	36
ตารางที่ 3.5	ตารางแสดงคำบรรยาย Use Case ประมวลผลด้วย FPGA	37
ตารางที่ 3.6	ตารางแสดงคำบรรยาย Use Case ดูแผนที่ร้านค้า	38
ตารางที่ 3.7	ตารางแสดงคำบรรยาย Use Case ดูข้อมูลสินค้าและร้านค้า	38
ตารางที่ 3.8	ตารางแสดงคำบรรยาย Use Case ระบุตำแหน่งที่อยู่ผู้ใช้งาน	39
ตารางที่ 3.9	พจนานุกรมตาราง products	58
ตารางที่ 3.10	พจนานุกรมตาราง stores	58
ตารางที่ 3.11	พจนานุกรมตาราง promoProducts	59
ตารางที่ 3.12	พจนานุกรมตาราง promoStores	59
ตารางที่ 4.1	ตารางอธิบายโครงสร้างไฟล์ระบบเบื้องหลัง	90
ตารางที่ 4.2	ตารางอธิบาย Table ใน Database	91
ตารางที่ 4.3	ตารางอธิบาย API ประมวลผลรูปภาพ	91
ตารางที่ 4.4	ตารางอธิบาย API เพิ่มข้อมูลร้านค้า	92
ตารางที่ 4.5	ตารางอธิบาย API เพิ่มข้อมูลโปรโมชั่นของร้านค้า	93
ตารางที่ 4.6	ตารางอธิบาย API แสดงรายการร้านค้าทั้งหมด	94
ตารางที่ 4.7	ตารางอธิบาย API แสดงรายละเอียดร้านค้า	95
ตารางที่ 4.8	ตารางอธิบาย API ค้นหาร้านค้าด้วยชื่อ	96
ตารางที่ 4.9	ตารางอธิบาย API อัปเดตข้อมูลร้านค้า	97
ตารางที่ 4.10	ตารางอธิบาย API ลบข้อมูลร้านค้า	98
ตารางที่ 4.11	ตารางอธิบาย API เพิ่มข้อมูลสินค้า	98
ตารางที่ 4.12	ตารางอธิบาย API เพิ่มข้อมูลโปรโมชั่นของสินค้า	99
ตารางที่ 4.13	ตารางอธิบาย API แสดงรายการสินค้าทั้งหมด	100
ตารางที่ 4.14	ตารางอธิบาย API แสดงรายละเอียดสินค้า	101
ตารางที่ 4.15	ตารางอธิบาย API ค้นหาสินค้าด้วยชื่อ	102
ตารางที่ 4.16	ตารางอธิบาย API อัปเดตข้อมูลร้านค้า	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกิจกรรมเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.17 ตารางอธิบาย API ลบข้อมูลสินค้า	104
ตารางที่ 4.18 ตารางรูปผลลัพธ์การตรวจจับสินค้าด้วยค่าน้ำหนัก Model สินค้าที่ 1	108
ตารางที่ 4.19 ตารางรูปผลลัพธ์การตรวจจับสินค้าด้วยค่าน้ำหนัก Model สินค้าที่ 2	110
ตารางที่ 4.20 ตารางรูปผลลัพธ์การตรวจจับสินค้าด้วยค่าน้ำหนัก Model สินค้าที่ 3	112
ตารางที่ 4.21 Test Case Mobile Application	126
ตารางที่ 4.22 Test Case RESTful API	127
ตารางที่ 4.23 Test Case โมเดลตรวจจับสินค้า หน้าร้านค้า และ Python Application	130
ตารางที่ 4.24 ตาราง Traceability Matrix	132



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

#### 1.1. ความเป็นมาและความสำคัญของปัญหา

บริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) หรือ National Telecom (NT) โดยมีที่ตั้งของอาคารสำนักงานใหญ่ ณ เลขที่ 99 ถนนแจ้งวัฒนะ แขวงทุ่งสองห้อง เขตหลักสี่ กรุงเทพมหานคร 10210 เป็นรัฐวิสาหกิจที่ดูแลการสื่อสารโทรคมนาคมในประเทศไทย

เนื่องจาก บริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) ได้มีแผนงานดำเนินการพัฒนาระบบต้นแบบที่มีการประยุกต์ใช้การ์ด FPGA ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ ในการทำงานด้านการค้นหาวัตถุ โดยประยุกต์ใช้กับการค้นหาสินค้าในห้างสรรพสินค้า ซึ่งได้เลือกห้างสรรพสินค้า IT Square หลักสี่ โดยได้มีการพิจารณาและคัดเลือกซึ่งเป็นห้างสรรพสินค้าที่มีร้านที่วางจำหน่ายสินค้าคล้ายคลึงกันเป็นจำนวนมาก ทำให้ผู้ใช้บริการเกิดความสับสนในการค้นหาร้านค้าและสินค้าที่ตนเองต้องการ เช่น ใช้เวลานานในการหาสินค้าที่ต้องการ โดยที่ไม่ทราบว่าสินค้าที่ต้องการอยู่ร้านไหนและส่วนใดในห้างสรรพสินค้า เป็นต้น การที่ลูกค้าต้องใช้เวลาและระยะทางในการเดินแต่ละจุดของร้านเพื่อซื้อสินค้าที่ห่างกันมากเกินไบนั้นสามารถส่งผลต่อความต้องการซื้อสินค้าของลูกค้าได้ ทางห้างสรรพสินค้าจึงมีความต้องการต้องการ Mobile Application ที่มาจัดการและแก้ไขปัญหา ซึ่งเป็นแพลตฟอร์มที่เข้าถึงลูกค้าที่มาเข้าใช้ห้างสรรพสินค้าได้ดีที่สุด โดยใช้ชื่อว่าโครงการ “Visual Search” ซึ่งโครงการแบ่งเป็น 2 ส่วน ได้แก่

ส่วนที่ 1 ส่วน Visual Search เป็นส่วนที่ใช้ค้นหาร้านค้าที่ต้องการผ่านทางรูปสินค้าที่นำเข้ามายังตัวแอปพลิเคชัน ระบบจะแสดงรายการร้านค้าที่มีสินค้าที่ใกล้เคียงกับสินค้าที่ลูกค้าต้องการออกมา และแสดงโปรโมชันของร้านค้านั้นๆ

ส่วนที่ 2 ส่วน Navigation Paths เป็นส่วนที่เกี่ยวกับการนำทางลูกค้าจากจุดเริ่มต้นที่ลูกค้าอยู่ไปยังร้านค้าที่ลูกค้าต้องการไป

ซึ่งทางคณะผู้จัดทำได้รับมอบหมายจากทางบริษัทให้ดูแลรับผิดชอบในส่วนที่ 1 (Visual Search)

ดังนั้นจึงเป็นที่มาของการพัฒนาแอปพลิเคชัน Visual Search เพื่อให้ลูกค้าสามารถค้นหาสินค้าหรือร้านค้าที่ต้องการด้วยภาพถ่ายซึ่งเป็นการช่วยลดเวลาในการหาสินค้าของลูกค้าและเป็นการดึงดูดให้ลูกค้าใช้บริการห้างสรรพสินค้ามากขึ้น พร้อมทั้งแสดงรายการส่งเสริมการขาย (Promotion) ที่ร้านค้าแต่ละร้านจัดทำสำหรับสินค้านั้น เป็นการโฆษณาสินค้าที่มีรายการส่งเสริมการขายเพื่อเพิ่มโอกาสการซื้อสินค้าแบบมีรายการเอกสารเป็นเอกสารพิสูจน์ส่วนไว้สำหรับการเช็กงานเพื่อการศึกษาเท่านั้น ไม่นอญูญาติหนาไปไซประโยชน์ดานการคาส่งเสริมการขายมากขึ้น

เมื่อมีการค้นเตฯ ทั้งสน อักทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2. วัตถุประสงค์ของโครงการ

- 1.2.1. เพื่อศึกษากระบวนการปฏิบัติงานวิจัยระบบต้นแบบ และประยุกต์ความรู้ที่ได้รับสู่การพัฒนาระบบต้นแบบจริง
- 1.2.2. เพื่อศึกษาการใช้งานและพัฒนาโมเดลปัญญาประดิษฐ์บนเทคโนโลยี DPU ของบริษัท Xilinx และประยุกต์การใช้งานเทคโนโลยีเข้ากับระบบต้นแบบที่จำลองการใช้งานแบบ Client/Server
- 1.2.3. เพื่อพัฒนาระบบต้นแบบที่สามารถเพิ่มจำนวนลูกค้าที่เข้ามาใช้บริการห้างสรรพสินค้า และเพิ่มความสะดวกสบายในการค้นหาสินค้าให้กับลูกค้าในห้างสรรพสินค้า

## 1.3. เป้าหมายของโครงการ

- 1.3.1. สร้างโมเดลปัญญาประดิษฐ์ Object Detection ที่ฝึกสอนบนข้อมูลสินค้าในห้างสรรพสินค้าสำหรับการประมวลผลบน DPU และคู่มือวิธีการพัฒนาโมเดลปัญญาประดิษฐ์ Object Detection เพื่อใช้ในการต่อยอดในโครงการอื่น
- 1.3.2. พัฒนาระบบต้นแบบ Mobile Application ของระบบ Visual Search สำหรับเป็นระบบหน้าบ้านของระบบ และสร้างคู่มือสำหรับการใช้งานระบบต้นแบบ Mobile Application เพื่อใช้ในการต่อยอดพัฒนาระบบต้นแบบให้สมบูรณ์
- 1.3.3. พัฒนาระบบต้นแบบ Web Server สำหรับการประมวลผล Object Detection กับเทคโนโลยี DPU และสร้างคู่มือสำหรับการใช้งานระบบต้นแบบ Web Server ที่เพื่อใช้ในการต่อยอดพัฒนาระบบต้นแบบให้สมบูรณ์

## 1.4. ขอบเขตของโครงการ

- 1.4.1. พัฒนาโมเดลปัญญาประดิษฐ์ Object Detection ที่ทำงานบนการ์ด GPU โดยใช้ข้อมูลจากภาพสินค้าในห้างสรรพสินค้า และระบุชื่อสินค้าในห้างสรรพสินค้า ได้โดยประมาณ 50 ชิ้น
- 1.4.2. พัฒนาโมเดลปัญญาประดิษฐ์ Object Detection ที่ทำงานบนการ์ด GPU โดยใช้ข้อมูลจากภาพร้านค้าในห้างสรรพสินค้า และระบุชื่อร้านค้าในห้างสรรพสินค้า ได้โดยประมาณ 20 ร้าน
- 1.4.3. ตั้งค่าสภาพแวดล้อมจำลองบนเครื่อง Server สำหรับการพัฒนาและทำงานกับเทคโนโลยี DPU บนการ์ด Xilinx Alveo U50LV
- 1.4.4. ปรับแต่งโมเดลปัญญาประดิษฐ์ Object Detection ให้สามารถทำงานกับเทคโนโลยี DPU บนการ์ด Xilinx Alveo U50LV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.4.5. พัฒนาระบบต้นแบบ Mobile Application บนระบบปฏิบัติการ iOS15 หรือระบบปฏิบัติการ Android 12 ขึ้นไป โดยถ่ายรูปรูปหรือเลือกรูปภาพจากคลังภาพในโทรศัพท์ และส่งข้อมูลรูปภาพขึ้นไปบนเครื่อง Server เพื่อประมวล เป็น Mobile Application ที่ต้องเชื่อมต่ออินเทอร์เน็ตระหว่างการใช้งาน
- 1.4.6. เชื่อมต่อระบบต้นแบบ Visual Search ให้รองรับกับการใช้งานระบบต้นแบบ Navigation Path
- 1.4.7. พัฒนาระบบต้นแบบ API สำหรับการประมวลผลรูปภาพโมเดลปัญญาประดิษฐ์บนการ์ด Xilinx Alveo U50 LV
- 1.4.8. ไม่มีการทดสอบระบบต้นแบบจำลองการทำงานแบบ Client/Server กับผู้ใช้งานจริง เนื่องจากเป็นระยะเวลาที่จำกัด และระบบที่ยังไม่สามารถใช้งานจริงในภาคธุรกิจได้ และข้อมูลที่ใช้ในงานการพัฒนา ระบบต้นแบบบางส่วนเป็นข้อมูลที่คณะผู้วิจัยจัดทำขึ้นมาเอง

## 1.5. ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1. นักศึกษาได้รับความรู้เกี่ยวกับกระบวนการและขั้นตอนการปฏิบัติงานวิจัยเทคโนโลยี DPU ของ Xilinx ด้วยการประยุกต์ใช้เข้ากับระบบต้นแบบจำลองการทำงานแบบ Client/Server และการพัฒนาโมเดลปัญญาประดิษฐ์ประเภท Object Detection ด้วย Darknet framework
- 1.5.2. บริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) ได้รับนวัตกรรมต้นแบบที่สามารถใช้งานในการค้นหาสินค้าหรือร้านค้าด้วยรูปภาพโดยประมวลผลด้วยโมเดลปัญญาประดิษฐ์ประเภท Object Detection และสามารถนำความรู้ คู่มือ แนวทางการพัฒนาโมเดลปัญญาประดิษฐ์นี้ไปใช้งานในโครงการอื่น และนำไปสู่การต่อยอดในโครงการต่อ ๆ ไป

## 1.6. อุปกรณ์และเครื่องมือที่ใช้

- Hardware
  - เครื่องคอมพิวเตอร์ Apple MacBook Pro 13-inch
    - : สำหรับการพัฒนาระบบ Mobile Application ทั้ง iOS และ Android และพัฒนาระบบ Web Server
    - OS : MacOS
    - CPU : Apple M1
    - Storage : SSD 256 GB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RAM : 8 GB
- เครื่องคอมพิวเตอร์ Apple Macbook Pro 16-inch
  - : สำหรับการพัฒนาระบบ Mobile Application ทั้ง iOS และ Android และพัฒนาระบบ Web Server
  - OS : MacOS
  - CPU : Apple M1 Pro
  - Storage : SSD 512 GB
  - RAM : 16 GB
- เครื่องคอมพิวเตอร์ Apple iMac 24-inch
  - : สำหรับการ Deploy Mobile Application
  - OS : MacOS
  - CPU : Apple M1
  - Storage : SSD 512 GB
  - RAM : 16 GB
- โทรศัพท์เคลื่อนที่ Apple iPhone 14 Pro Max
  - : เครื่องสำหรับทดสอบ Application บน iOS
  - OS : iOS
  - CPU : Apple A16 Bionic
  - Storage : 256 GB
  - RAM : 6 GB
- โทรศัพท์เคลื่อนที่ Apple iPhone XS Max
  - : เครื่องสำหรับทดสอบ Application บน iOS
  - OS : iOS
  - CPU : Apple A12 Bionic
  - Storage : 64 GB
  - RAM : 4 GB
- โทรศัพท์เคลื่อนที่ Samsung Galaxy Z Fold 4
  - : เครื่องสำหรับทดสอบ Application บน Android

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่มีเห็นแต่เพียงอย่างเดียว และยังอยู่ใต้อาณัติของเอกสารทุกครั้งที่มีการนำไปใช้

- Storage : 256 GB
  - RAM : 12 GB
- เครื่องเซิร์ฟเวอร์ Dell PowerEdge R740
  - : เครื่อง Server สำหรับการฝึกสอนโมเดลปัญญาประดิษฐ์
    - การ์ด GPU Nvidia Tesla V100
- เครื่องเซิร์ฟเวอร์ Boston Supermicro
  - : เครื่อง Server สำหรับการทดลองใช้งานเทคโนโลยี DPU
    - การ์ด FPGA ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ Xilinx Alveo U50LV
- เครื่องเซิร์ฟเวอร์ Gigabyte R282-Z9
  - : เครื่อง Server สำหรับ Hosting API และประมวลผลโมเดลปัญญาประดิษฐ์ด้วยเทคโนโลยี DPU
    - การ์ด FPGA ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ Xilinx Alveo U50LV
- การ์ด FPGA ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ Xilinx Alveo U200
- การ์ด FPGA ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ Xilinx Alveo U280
- ระบบปฏิบัติการบนเครื่อง Server Ubuntu เวอร์ชัน 18.04
- Software
  - Vitis-AI Development Platform เวอร์ชัน 2.5.0
  - Darknet Framework เวอร์ชัน 2021.1
  - React Native Framework เวอร์ชัน 0.69.5
  - Docker เวอร์ชัน 19.03.8
  - GitHub Version control
  - Figma
  - PostgreSQL เวอร์ชัน 14.5
  - Roboflow
- Programming Language
  - JavaScript
  - SQL
  - Python
  - CSS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.7. เงื่อนไข/ข้อจำกัด

- 1.7.1. ผู้ดูแลโครงการฝั่งบริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) จะต้องเป็นผู้ประสานงานในการเก็บข้อมูลจากห้างสรรพสินค้าให้กับคณะผู้จัดทำ
- 1.7.2. ผู้ดูแลโครงการฝั่งบริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) จะต้องเป็นผู้จัดหาอุปกรณ์เครื่อง Server ที่มีการ์ด FPGA ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ ของ Xilinx
- 1.7.3. คู่มือการใช้งานทั้งหมดจัดทำขึ้นบน Environment จำเพาะตามข้อมูลในเอกสารคู่มืออื่นๆ
- 1.7.4. ไม่มีการอัปเดตข้อมูลของคู่มือการใช้งานทั้งหมดหลังจากสิ้นสุดระยะเวลาของโครงการ
- 1.7.5. โครงการนี้มีระยะเวลาดำเนินการรวมทั้งหมด 7 เดือน
- 1.7.6. โครงการนี้มีผู้ดูแลโครงการฝั่งบริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) จำนวน 1 คน และมีโครงการอื่นๆ ที่ผู้ดูแลโครงการการฝั่งบริษัทรับผิดชอบหลายโครงการจึงอาจทำให้มีข้อจำกัดเรื่องการดูแลและแนะนำแนวทางของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องของโครงการพิเศษ ประกอบด้วยแนวคิด ทฤษฎี ที่นำมาประยุกต์ใช้กับโครงการพิเศษ ตลอดจนการทบทวนวรรณกรรมและงานวิจัยที่มีความใกล้เคียงหรือเกี่ยวข้องกับโครงการพิเศษที่ศึกษา

### 2.1. การพัฒนา Mobile Application

(สารานุกรมเสรี, 2022) โทรศัพท์มือถือ(Mobile) เป็นเทคโนโลยีที่สำคัญอย่างมากในการติดต่อสื่อสารของมนุษย์ ซึ่งปัจจุบันเทคโนโลยี โทรศัพท์มือถือ ได้พัฒนาให้มีความสามารถใช้งานโปรแกรมประยุกต์หรือแอปพลิเคชัน (Application) ใน โทรศัพท์มือถือได้ และถูกเรียกกล่าวในชื่อ สมาร์ทโฟน (Smart Phone) ความสามารถในการใช้งานแอปพลิเคชันบนโทรศัพท์มือถือได้ ทำให้ สมาร์ทโฟน มีความสามารถในการทำงานอย่างอื่นเพิ่มเติมนอกเหนือจากการสื่อสาร เช่นการใช้อินเทอร์เน็ต การถ่ายภาพถ่าย และการใช้งาน GPS เพื่อนำทาง เป็นต้น

ในปัจจุบัน สมาร์ทโฟน เป็นที่นิยมและมีการใช้งานกันอย่างแพร่หลาย จึงเป็นเหตุผลการพัฒนาแอปพลิเคชันบนมือถือ (Mobile Application) เป็นที่นิยมเป็นอย่างมาก ซึ่งการพัฒนาแอปพลิเคชันบนมือถือสามารถพัฒนาได้หลายรูปแบบตามระบบปฏิบัติการสมาร์ตโฟนที่ต้องการใช้งาน ทั้งรูปแบบที่เป็นการพัฒนากับภาษาที่ระบบปฏิบัติการแนะนำโดย (Native) และการใช้เครื่องมือช่วยในการพัฒนาแบบข้ามระบบปฏิบัติการ (Cross-platform) ซึ่งเครื่องมือที่นิยมใช้งานในการพัฒนาแอปพลิเคชันมือถือคือ React Native

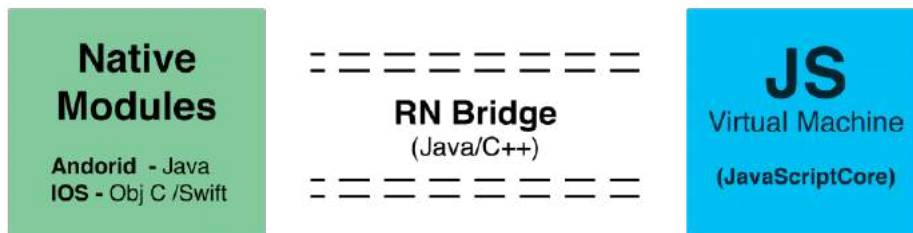
#### 2.1.1. React Native framework

React Native คือ Javascript Framework หรือ ชุดคำสั่งที่ถูกสร้างขึ้นสำหรับพัฒนา Mobile Application แบบ Cross-platform เราสามารถเขียนคำสั่งชุดเดียว หรือมีความแตกต่างกันแต่ไม่มาก แต่สามารถทำงานได้บนระบบปฏิบัติการที่แตกต่างกันทั้งระบบปฏิบัติการ iOS และระบบปฏิบัติการ Android โดยใช้ Javascript เป็นภาษาหลักในการเขียนโปรแกรม โดยมีความเสถียรที่เทียบเท่าหรือใกล้เคียงกับ Mobile Apps ที่พัฒนาแบบ Native สูง

- โครงสร้างหลักของ React Native ถูกพัฒนาตั้งต้นมาแบบ Native เรียกว่า Native Modules ความหมายคือ ถ้าเราจำเป็นจะต้องใช้งาน ScrollView เพื่อแสดงรายการสินค้าหรือร้านค้า ภายในแอป React Native จะมี ScrollView Component ที่ถูกพัฒนามาจาก Objective-C หรือ Swift สำหรับ iOS และมี ScrollView ที่ถูกพัฒนาจากภาษา Java สำหรับ Android

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนสิทธิ์ในเนื้อหาและข้อมูลที่มีอยู่ทั้งหมดโดยไม่มีการยินยอมใดๆ ทั้งสิ้น อีกทั้งยังมีเนื้อหาที่ละเอียดและต้องอยู่ภายใต้เงื่อนไขของเอกสารที่แนบมาไว้

ให้เราใช้งานสั่งการ เขียนโปรแกรมด้วยภาษา JavaScript ภาษาเดียว เรียกว่า React Native Bridge



รูปที่ 2.1 โครงสร้าง React Native

(codebee, 2022)

- Native Modules หรือ Package นักพัฒนาทั่วโลกสามารถพัฒนาฟังก์ชันที่ต้องการเองได้ เพื่อให้ตรงกับการใช้งานที่หลากหลายและแตกต่างกัน ด้วย Community ที่แข็งแกร่งของ React Native ทำให้ปัจจุบันมีการพัฒนา Native Modules จากนักพัฒนาทั่วโลก และเปิดให้นักพัฒนาสาย React Native ใช้งานได้ ซึ่งแทบจะครอบคลุมทุกส่วนการพัฒนาสำหรับ Mobile Application

#### 2.1.2. Package ที่เกี่ยวข้องและจำเป็นต่อการพัฒนาระบบ Visual Search ส่วนที่ 1

- react-native-axios เป็น Package ที่ทำงานบน React Native และมีฟังก์ชันการทำงานที่เหมือน JavaScript library Axios ซึ่ง Axios มีทำหน้าที่ในการทำ http request หรือเชื่อมต่อเพื่อร้องขอการรับส่งข้อมูลกับ API Service เพื่อการรับส่งข้อมูลแบบ RESTful API โดยที่ Axios จะทำหน้าที่เป็นตัวกลางในการจัดการทั้ง method, data, headers, security และอื่นๆ ที่เกี่ยวข้องกับการรับส่งข้อมูล และ Axios ยังรองรับการทำ Promises async/await
- react-native-vision-camera เป็น Package สำหรับการเรียกใช้งานกล้องบน Smart Phone โดยสามารถฟังก์ชันการทำงานเหมือน Application Camera ที่เป็นแอปพลิเคชันพื้นฐานบน Smart Phone และสามารถออกแบบหน้าจอกำหนดการใช้งานกล้องเองได้ตั้งแต่ขั้นพื้นฐานจนถึงขั้นสูง
- react-native-image-picker เป็น Package สำหรับการดึงข้อมูลรูปภาพจากคลังรูปภาพในโทรศัพท์ ใช้ในกรณีที่มีรูปสินค้าหรือรูปร้านค้าที่ต้องการจะค้นหาค้นหาบันทึกไว้ในเครื่องอยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2. การพัฒนา API

API หรือ ส่วนต่อประสานโปรแกรมประยุกต์ (Application Program Interface) เป็นตัวกลางที่ช่วยให้ส่วนประกอบหรือระบบซอฟต์แวร์สองส่วนสามารถสื่อสารกันได้โดยใช้ชุดคำเฉพาะและโปรโตคอลเดียวกัน ตัวอย่างเช่น ระบบซอฟต์แวร์ของสำนักพยากรณ์อากาศประกอบด้วยข้อมูลสภาพอากาศรายวัน แอปพลิเคชันสภาพอากาศในโทรศัพท์จะสื่อสาร กับระบบนี้ผ่าน API และแสดงการอัปเดตสภาพอากาศทุกวันบนแอปพลิเคชันนั้นๆ

สถาปัตยกรรม API มักจะถูกอธิบายในแง่ของไคลเอ็นต์ (Client) และเซิร์ฟเวอร์(Server) แอปพลิเคชันที่ส่งคำขอในการสื่อสารเรียกว่าไคลเอ็นต์ และแอปพลิเคชันที่ส่งการตอบกลับเรียกว่าเซิร์ฟเวอร์ ในตัวอย่างสภาพอากาศ ฐานข้อมูลสภาพอากาศของสำนักงานคือเซิร์ฟเวอร์ และแอปพลิเคชันมือถือคือไคลเอ็นต์ ซึ่งในโครงการนี้มีการสร้างสภาพแวดล้อมจำลองสำหรับการพัฒนาเซิร์ฟเวอร์ โดยใช้ Docker เป็นเครื่องมือในการสร้างสภาพแวดล้อมจำลองนั้น

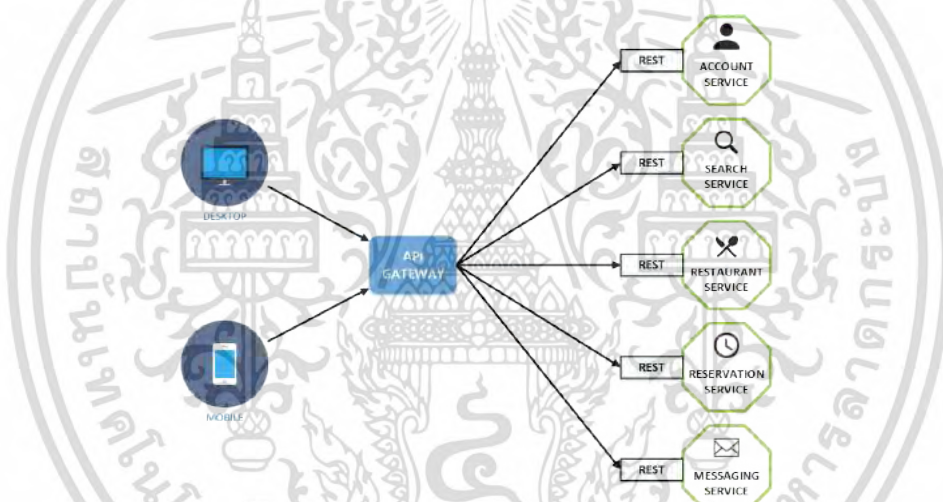
### 2.2.1. RESTful API

REST ย่อมาจาก Representational State Transfer (การโอนสถานะแบบตัวแทน) REST ช่วยกำหนดชุดฟังก์ชันต่างๆ เช่น GET, PUT, DELETE ฯลฯ ที่ไคลเอ็นต์สามารถใช้เพื่อเข้าถึงข้อมูลเซิร์ฟเวอร์ได้ ไคลเอ็นต์และเซิร์ฟเวอร์แลกเปลี่ยนข้อมูลโดยใช้ HTTP คุณสมบัตินี้ที่สำคัญของ RESTful API ประกอบไปด้วย (Amazon, 2022)

- โครงสร้างของระบบ Client-Server นั้นจะต้องประกอบไปด้วย ไคลเอ็นต์ เซิร์ฟเวอร์และทรัพยากร ที่แลกเปลี่ยนข้อมูลด้วยการส่งคำขอ (Request) ผ่าน HTTP Protocol
- Stateless การสื่อสารระหว่าง Client-Server จะต้องไม่มีการเก็บข้อมูลไคลเอ็นต์ที่ส่งคำขอ แต่ละคำขอจะต้อง แยกออกจากกัน และไม่มีความเชื่อมโยงกัน โดย Server จะต้องมีการดำเนินการกับคำขอจากไคลเอ็นต์ทั้งหมดโดยไม่ขึ้นกับคำขอก่อนหน้า
- มีความสามารถในการเก็บข้อมูลแคช ซึ่งเป็นกระบวนการจัดเก็บการตอบสนองบางส่วนบนไคลเอ็นต์หรือตัวกลางเพื่อปรับปรุงเวลาตอบสนองของเซิร์ฟเวอร์
- ต้องมีรูปแบบมาตรฐานส่วนติดต่อกับเซิร์ฟเวอร์ โดยจะมีมาตรฐานดังนี้:
  - คำขอควรใช้ตัวระบุทรัพยากรที่ต้องการร้องขอ และใช้ตัวระบุทรัพยากรรูปแบบเดียวกัน
  - ทรัพยากรที่เซิร์ฟเวอร์ส่งจะต้องเพียงพอที่จะทำให้ ไคลเอ็นต์จัดการและใช้งานกับทรัพยากรที่ได้รับได้
  - มีการอธิบายคำตอบกลับจากเซิร์ฟเวอร์ที่เพียงพอสำหรับ เพื่อให้ไคลเอ็นต์เข้าใจขั้นตอนดำเนินการกับข้อมูลนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่นิยามให้นำไปใช้โดยไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่แบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โคลเอ็นต์ได้รับข้อมูลที่เกี่ยวข้องกับทรัพยากรทั้งหมดที่จำเป็นต่อการทำงานให้เสร็จสมบูรณ์ มีไฮเปอร์ลิงก์ในการแทนข้อมูลส่วนที่เพิ่มเติม หากโคลเอ็นต์ต้องการเข้าถึงทรัพยากรเพิ่มเติมต้องมีไฮเปอร์ลิงก์เพื่อให้โคลเอ็นต์สามารถค้นพบทรัพยากรเพิ่มเติมทั้งหมดได้
- ระบบที่แบ่งออกเป็นชั้น โคลเอ็นต์สามารถเชื่อมต่อกับตัวกลางอื่นๆ ที่ได้รับอนุญาตระหว่างโคลเอ็นต์และเซิร์ฟเวอร์ได้ และจะยังคงได้รับการตอบสนองจากเซิร์ฟเวอร์ เซิร์ฟเวอร์ยังสามารถส่งต่อคำขอไปยังเซิร์ฟเวอร์อื่นได้อีกด้วย เพื่อความปลอดภัย การทำงาน และตรรกะทางธุรกิจ โดยทำงานร่วมกันเพื่อตอบสนองคำขอของโคลเอ็นต์ โดยโคลเอ็นต์ไม่สามารถมองเห็นชั้นต่างๆ เหล่านี้ได้
- การปรับแต่งโค้ดได้ตามความต้องการ เซิร์ฟเวอร์มีความสามารถในการขยายหรือปรับแต่งฟังก์ชันการทำงานของโคลเอ็นต์ได้ชั่วคราวโดยการถ่ายโอนโค้ดการเขียนโปรแกรมซอฟต์แวร์ไปยังโคลเอ็นต์



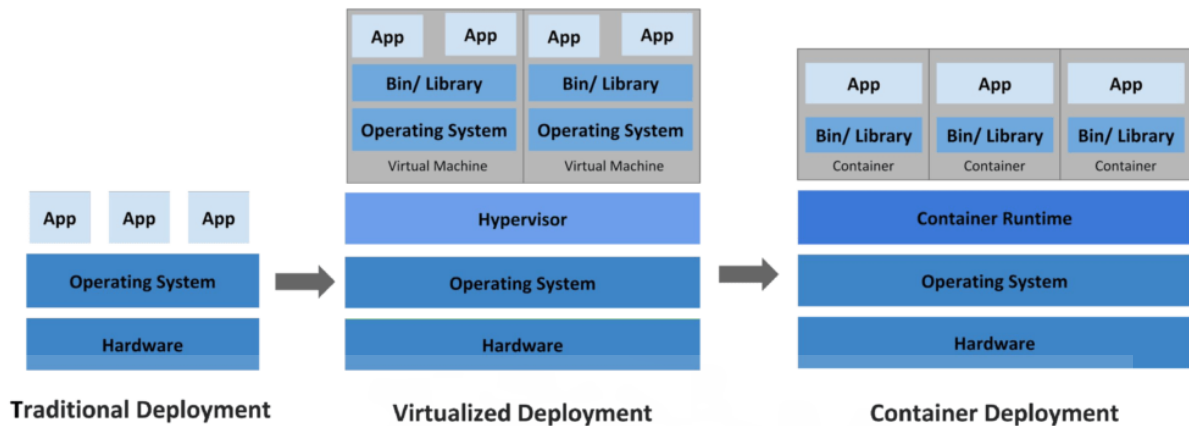
รูปที่ 2.2 ตัวอย่างโครงสร้างสถาปัตยกรรม RESTful API

(nich, 2023)

### 2.2.2. Docker

Docker คือ เครื่องมือแบบ open source ที่ช่วยจำลองสภาพแวดล้อม (environment) ในการเริ่มต้นการทำงาน service หรือ server ตามหลักการสร้าง container เพื่อจัดการกับ library ต่างๆ อีกทั้งยังช่วยจัดการในเรื่องของ version control เพื่อง่ายต่อการจัดการกับปัญหาต่างๆ ที่เกิดขึ้น วิวัฒนาการของการ Deploy โดยแบ่งการ Deploy แบ่งเป็น 3 ยุคง่าย ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Intro to Docker

### รูปที่ 2.3 ภาพอธิบายประวัติศาสตร์การ Deployment

(Boonmathanaruk, 2022)

1. Traditional Deployment เป็นยุคเริ่มแรกของการ Deploy โดยการ Deploy จะใช้เครื่อง Physical Server เครื่องเดียวในการเริ่มต้นการทำงาน Server หรือ Service และมีการใช้งาน Application หลาย ๆ ตัวพร้อมกันเพื่อประหยัดค่าใช้จ่าย จึงทำให้เกิดปัญหาเฉพาะของแต่ละ Application แต่ละตัวชนกัน เช่น เครื่อง 1 เครื่องมีหลาย Application ที่ใช้งาน JAVA เวอร์ชันที่แตกต่างกัน ทำให้เวลาเริ่มต้นการทำงานพร้อมกันจะมีปัญหาเกิดขึ้น ทั้งการ maintenance และปัญหาในการเลือกเวอร์ชัน
2. Virtualized Deployment เป็นยุคต่อมาที่มีการเกิดขึ้นของ Software Hypervisor ซึ่งเป็นการจำลองเครื่อง Physical Server ขึ้นมาโดยเรียกว่า Virtual Machine ทั้ง CPU, memory, hard disk, hardware ต่างๆ ขึ้นมาเสมือนคอมพิวเตอร์อีกเครื่องหนึ่ง โดยปรับสเปคปรับความเร็วต่างๆ ได้ตามงบ Hardware เครื่องที่เป็น Host สามารถรองรับได้ ซึ่งส่วนใหญ่ก็จะสร้าง VM ขึ้นมาหลายๆ เครื่อง ให้แต่ละเครื่องเพียงพอต่อการใช้งาน Application แต่ละตัว แต่ปัญหาคือจะเกิดปัญหา Overhead จากการจำลอง Hardware ของแต่ละ VM ทำให้ทำงานได้ช้าลง 2-5 เท่า รวมถึง Overhead ในกรณีที่เรามีการใช้งานใน environment ที่คล้ายคลึงกัน จะทำให้เปลืองทรัพยากรของเครื่อง
3. Container Deployment ซึ่งในยุคนี้จะเป็นการสร้าง Container ขึ้นมาเพื่อแบ่งการใช้งานทรัพยากร โดยสร้างกำแพงสำหรับแต่ละ Container ทำให้เราสามารถใส่ทรัพยากรได้อย่างมีประสิทธิภาพมากขึ้น สามารถเลือก Container ไปใช้กับแต่ละ Application ได้ดียิ่งขึ้น ทำให้ปัญหา overhead ลดลง และเครื่องมือในการสร้าง Container ที่นิยมก็คือ Docker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 2.2.3. PostgreSQL  
 ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PostgreSQL เป็นระบบจัดการฐานข้อมูล open source ประเภท Relational Database Management System (RDBMS) ที่มีความสามารถในการจัดการฐานข้อมูลเชิงวัตถุ (Object Relational Database Management หรือ ORDBMS) ซึ่งยึดหลักการและปฏิบัติตามข้อกำหนดของ SQL ที่เป็นมาตรฐานของตัวภาษาแบบที่มีโครงสร้าง โดย PostgreSQL ใช้สถาปัตยกรรมระบบรูปแบบ Client-Server ที่ Client และ Server สามารถอยู่บนโฮสต์ที่แตกต่างกันในสภาพแวดล้อมเครือข่ายเดียวกัน ส่วนของ Server จะจัดการไฟล์ฐานข้อมูลเมื่อยอมรับการเชื่อมต่อกับฐานข้อมูลจาก Client สามารถจัดการการเชื่อมต่อหลายรายการพร้อมกันจาก Client ได้ด้วย Multiversion Concurrency Control (MVCC) (linux-console.net, 2023)

#### 2.2.3.1. Multiversion Concurrency Control

MVCC เป็นวิธีการจัดการธุรกรรมอย่างหนึ่งที่เข้ามาแทนที่การ Lock เพื่อป้องกันการเขียนที่ผิดพลาด ซึ่งทำให้ในขณะที่ถูก Lock ไม่มีใครสามารถอ่านข้อมูลได้ โดยหลักการ ของ MVCC ของ PostgreSQL จะใช้การสร้าง snapshot ของข้อมูลในแต่ละ Version โดยไม่คำนึงถึงสถานะปัจจุบันของข้อมูลที่เป็นฐาน ซึ่งช่วยป้องกันธุรกรรมจากการดูข้อมูลที่ไม่สอดคล้องกัน ซึ่งอาจเกิดจากการอัปเดตธุรกรรมพร้อมกัน ทำให้เกิดการแยกธุรกรรมสำหรับแต่ละ Session

#### 2.2.3.2. ข้อดี

- รองรับ Data Type ที่เป็น NoSQL ด้วย เช่น พวก Geometric ต่าง ๆ เช่น Points, Circles และ Polygons รวมถึงรองรับ native xml สามารถใช้ key-value เทียบค่า กับ HSTORE แล้วยังสามารถทำ index ของ JSON ในการเข้าถึงข้อมูลได้
- มีความสามารถในการรองรับ concurrency สูง
- มีความเร็วทั้งในการอ่าน และ เขียน ที่ดีกว่า MySQL
- มี Security ที่เป็นค่าเริ่มต้นดีกว่า MySQL เพราะมีการใช้งาน SEPostgreSQL

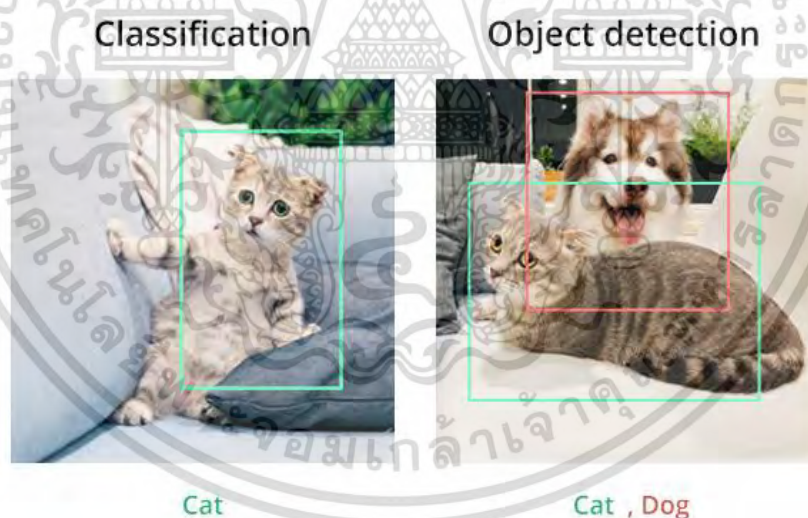
#### 2.2.3.3. ข้อเสีย

- ใช้ Memory มากเมื่อมี Concurrent สูง เพราะต้องมีการสร้าง snapshot ของข้อมูลหลายๆ Session
- มี Community ในการใช้งานน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3. Object Detection

- 2.3.1. Object Detection คือ ความสามารถของโปรแกรมในการตรวจจับวัตถุ เป็นเทคโนโลยีที่ใช้สำหรับการตรวจจับวัตถุ โดยเป็นหนึ่งในพีเจอรหลักของ AI (Artificial Intelligence) ที่ใช้กับรูปภาพจากกล้อง ใช้ในการค้นหาสิ่งของโดยใช้ AI มาวิเคราะห์ข้อมูลรูปภาพด้วยกระบวนการทำการมองเห็นของคอมพิวเตอร์ (Computer Vision) และการประมวลผลภาพ (Image Processing) เพื่อตรวจจับวัตถุที่อยู่ในรูปหรือวิดีโอ เช่น มนุษย์ สัตว์ สิ่งของ รถยนต์ อาคาร และวัตถุอื่น ๆ ที่อยู่ในรูปภาพ หรือวิดีโอ (Kanoktipsatharporn, 2022)
- 2.3.2. การทำงาน Object Detection เทคโนโลยีตรวจจับวัตถุ (Object detection) จะต้องผ่านการจัดหมวดหมู่ของวัตถุ (Object Classification) มาก่อน โดยที่การจัดหมวดหมู่ของวัตถุจะเป็นการจัดหมวดหมู่ของรูปภาพว่า รูปภาพนั้นคือภาพอะไร แต่เทคโนโลยีตรวจจับวัตถุจะเป็นการระบุเลยว่า ในรูปภาพนั้นมีวัตถุอะไรบ้าง ซึ่งจุดนี้จะต้องอาศัยการทำงานของ AI เข้ามาช่วยในการวิเคราะห์ข้อมูลด้วยกัน เช่น การจัดหมวดหมู่ของวัตถุจะสามารถระบุได้ว่าวัตถุที่อยู่ในภาพ คือ แมว ในขณะที่เทคโนโลยีตรวจจับวัตถุจะระบุได้ว่าวัตถุที่อยู่ในภาพ มีแมว และสุนัข



รูปที่ 2.4 ภาพ Classification และ Objection detection

(Warasanan, 2022)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4. การพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการทำ Object Detection

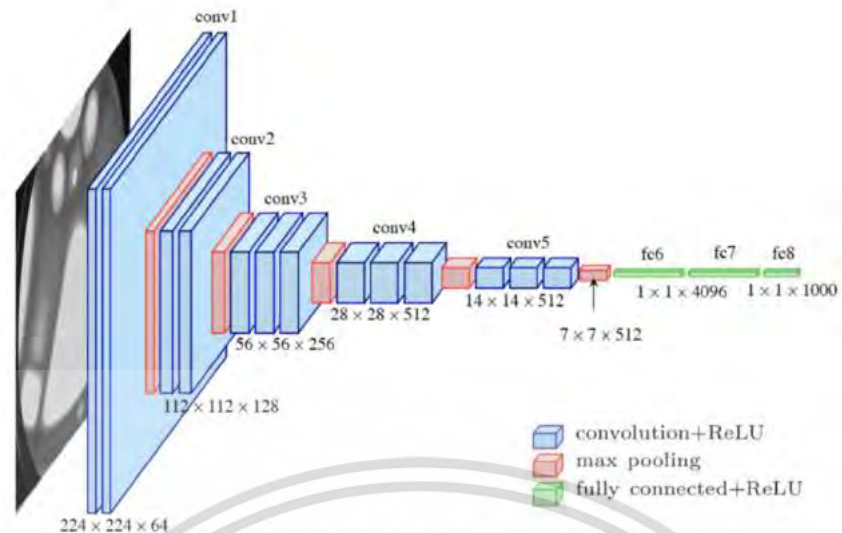
การพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการทำ Object Detection เป็นหนึ่งในกระบวนการทำให้คอมพิวเตอร์มองเห็นเช่นเดียวกับมนุษย์ (Computer Vision) ซึ่งมีเทคนิคสาขาที่แยกย่อยออกไป ไม่ว่าจะเป็น Object Detection, Image Classification, Object Segmentation, และอีกมากมายมีเทคนิคที่ใช้ในการพัฒนาที่หลากหลาย ในงานวิจัยนี้จะเน้นเทคนิคที่เกี่ยวข้องกับ Object Detection, Image Recognition ซึ่งเทคนิคในการพัฒนาโมเดล Object Detection ที่เป็นที่ยอมรับที่สุดคือเทคนิคการทำ Deep Learning หรือการเรียนรู้เชิงลึกโดยใช้ Convolutional Neural Network (CNN) หรือ โครงข่ายประสาทแบบคอนโวลูชัน เป็นโครงสร้างหลักในการทำงาน ซึ่งการฝึกสอน Model สำหรับการทำ Object Detection มีเครื่องมือที่ช่วยในการทำงานมากมาย ไม่ว่าจะเป็น TensorFlow, Keras, Pytorch, Darknet และ Caffe และยังมีโครงสร้าง Convolutional Neural Network ที่ผ่านการออกแบบโครงสร้างและฝึกสอนมาแล้ว (Pre-trained Model) หลากหลายรูปแบบที่ช่วยในการสร้างระบบ Object Detection โดยคณะผู้จัดทำได้เลือกโครงสร้าง Convolutional Neural Network ที่เป็นที่ยอมรับในการพัฒนาได้แก่ โครงสร้างที่มีชื่อว่า YOLO หรือ You Only Look Once

### 2.4.1. Deep Learning และ Convolutional Neural Network

Deep Learning คือวิธีการเรียนรู้ของโปรแกรมแบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) เพื่อสร้างแบบจำลองการทำงานของโครงข่ายประสาทของมนุษย์ (Neural Network) มีชื่อเรียกว่า โครงข่ายประสาทเทียม หรือ Artificial Neural Network (ANN) หลักการทำงานของระบบโครงข่ายประสาทเทียม คือการนำ node ที่เปรียบเสมือนการทำงานของเซลล์ประสาทมนุษย์มาซ้อนกันหลายชั้นเป็น Layer และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูลดังกล่าวจะถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจัดหมวดหมู่ข้อมูล (Classify the Data) ซึ่งโครงข่ายประสาทเทียมมีอยู่หลายประเภทขึ้นอยู่กับการจัดโครงสร้างของโครงข่ายประสาทเทียม โดยการทำงานเกี่ยวกับรูปภาพ (Computer Vision) จะใช้ Convolutional Neural Network เป็นหลัก

Convolutional Neural Network ซึ่งเป็นโครงสร้างโครงข่ายประสาทเทียมแบบพิเศษ ที่มีความสามารถในการจำแนกข้อมูลประเภทรูปภาพได้ดีกว่า Neural network ทั่วไป หลักการทำงานของ Convolutional Neural Network คือการใช้ Layer ชนิดพิเศษ ที่เรียกว่า Convolution layer ซึ่งทำหน้าที่สกัดเอาเฉพาะส่วนที่สำคัญต่างๆ ของภาพ เช่น เส้นขอบของวัตถุต่างๆ เพื่อให้โมเดลสามารถเรียนรู้ลักษณะเฉพาะ (Feature) ของภาพได้อย่างมีประสิทธิภาพและ

เอกสารนี้เป็นแม่แบบที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ภาพตัวอย่างโครงสร้าง Convolutional Neural Network ของ VGG16

(Le, 2022)

#### 2.4.2. YOLO

YOLO หรือ You Only Look Once เป็นโครงสร้าง Convolutional Neural Network ที่ผ่านการออกแบบและฝึกสอนมาแล้วเพื่อใช้งานสำหรับการทำ Object Detection มีความสามารถสูงทั้งด้านความเร็วในการคำนวณและความแม่นยำ และง่ายต่อการฝึกสอนต่อ (Transfer Learning) จึงทำให้โครงสร้างของ YOLO เป็นที่นิยมอย่างมาก ปัจจุบัน YOLO มีการพัฒนามาจนถึงเวอร์ชันที่ 7

#### 2.4.3. การพัฒนาระบบโมเดลปัญญาประดิษฐ์ YOLOv4 ด้วย Darknet framework

Darknet framework เป็น open source neural network framework ถูกสร้างด้วยภาษา C และ CUDA สำหรับการพัฒนาโมเดลปัญญาประดิษฐ์โดยเฉพาะ YOLO และมีโครงสร้าง Convolutional Neural Network หลายรูปแบบให้ฝึกสอนใหม่ด้วยชุดข้อมูลของตัวเองได้ โดย Basic terms ของ Darknet จะใช้ไฟล์สองตัวในการตั้งค่าการพัฒนา

- cfg-files ไฟล์สำหรับบอกโครงสร้างและค่าน้ำหนัก Convolutional Neural Network
- weights-files ไฟล์ค่าน้ำหนักที่สอดคล้องกับ cfg-files ไว้สำหรับกาตรวจจ็บบรูปแบบ

#### 2.4.4. TensorFlow คืออะไร

TensorFlow เป็น open source neural network framework เช่นเดียวกับ Darknet framework แต่ต่างที่ TensorFlow รองรับการใช้งานด้วยภาษา Python ซึ่งพัฒนาโดย Google

และถูกสร้างมาเพื่อให้ง่ายต่อการใช้งาน TensorFlow ได้รวมเอา API ที่แตกต่างกัน เพื่อสร้างสถาปัตยกรรม deep learning หลายแบบทั้งอย่าง CNN และ RNN นอกจากนี้ TensorFlow ยังมี

ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกทั้งห้ามมิให้ตีแบบสงวนเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มาใช้

graph เป็นตัวคำนวณหลัก ทำให้นักพัฒนาเห็นภาพโครงสร้าง neural network สามารถปรับสเกล ประมวลผลได้ ให้ทำงานได้ CPUs และ GPUs จึงทำให้ TensorFlow เป็นที่นิยมมากกว่า และ ได้รับการรองรับจากเครื่องมือการปรับแต่งโมเดลที่ Xilinx เตรียมมาให้ใน Vitis-AI

## 2.5. เทคโนโลยี Data Center accelerator ของ Xilinx

เทคโนโลยี Data Center accelerator เป็นการออกแบบชิป FPGA สำหรับงานประมวลผล Machine Learning, Computational Storage, การค้นหาข้อมูล และวิเคราะห์ข้อมูล โดย Xilinx ได้สร้างเทคโนโลยีนี้ ทั้งในแบบที่เป็น Embedded Board มีชื่อว่า UltraScale+ และเป็นการ์ด PCIe ในชื่อ Alveo การประมวล โมเดลปัญญาประดิษฐ์ ด้วย Data Center accelerator ของ Xilinx นั้นมีคุณสมบัติในการประมวลผล ปัญญาประดิษฐ์ที่ดีกว่า GPU ทั้งในด้านในการประมวลผลที่เร็วกว่า และประหยัดพลังงานในการคำนวณ

### 2.5.1. Field-Programmable Gate Array

(WellPCB, 2023) Field-Programmable Gate Array หรือ FPGA เป็นวงจรรวมประเภทหนึ่ง วิศวกรหรือนักออกแบบสามารถตั้งโปรแกรมได้หลังการผลิต เพื่อให้สามารถทำงานต่างๆได้ตามที่ ต้องการ FPGA มาพร้อมกับบล็อกลอจิกที่ตั้งโปรแกรมได้ และกรอบของการเชื่อมต่อระหว่างกันที่ สามารถกำหนดค่าใหม่เพื่อให้สามารถเดินสายไฟใหม่ ลอจิกเกต หรือ ประตูสัญญาณตรรกะ สามารถเชื่อมต่อระหว่างกันได้โดยใช้การกำหนดค่าที่แตกต่างกัน ซึ่งทำให้สามารถต่อเข้าด้วยกันได้ สามารถใช้ฟังก์ชันผสมผสานที่ซับซ้อนได้ โดยกำหนดค่าบล็อกลอจิกและลอจิกเกตที่ปลอดภัย เช่น XOR และ AND

### 2.5.2. The Xilinx® Deep Learning Processor Unit (DPU)

(Xilinx, DPU for Convolutional Neural Network, 2022) เทคโนโลยี Deep Learning Processor Unit ของ Xilinx เป็น programmable engine ที่ออกแบบมาเพื่อทำงานกับชิป FPGA สำหรับการใช้งานประมวลผล Convolutional Neural Network ซึ่งเป็นโครงข่ายประสาท เทียมหนึ่งประเภทหนึ่งของปัญญาประดิษฐ์ ซึ่งใช้พลังงานในการประมวลผลสูง โดยเทคโนโลยี Deep Learning Processor Unit ของ Xilinx ประกอบไปด้วย Register Configure Module, Data Controller Module และ Convolution Computing Module ออกแบบมาให้ใช้งานกับ แบบจำลอง Convolutional Neural Network ประสิทธิภาพสูงหลายประเภทไม่ว่าจะเป็น VGG, ResNet, GoogLeNet, YOLO, SSD, MobileNet, FPN และอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Xilinx Alveo U50 Data Center accelerator cards

(Xilinx, Alveo U50 Data Center Accelerator Card Data Sheet, 2022)

## 2.6. การพัฒนาและใช้งาน Model AI บนการ์ด Data Center accelerator Cards

โดยทั่วไปแล้วการ์ด Data Center accelerator ลักษณะภายนอกและ Slot สำหรับการเชื่อมต่อการ์ด Data Center accelerator กับ Mainboard จะเชื่อมต่อกันผ่าน Slot PCIe x16 ซึ่งจะมีความคล้ายคลึงกับ GPU แต่การตั้งค่า Driver และการใช้งานสำหรับการประมวลผลโมเดล AI จะมีความแตกต่างกัน โดยที่การ์ด Data Center accelerator ประเภทเร่งการประมวลผลปัญญาประดิษฐ์ของ Xilinx เมื่อติดตั้งใหม่ครั้งแรก จะต้องมีการลง Driver ที่เป็นตัวติดต่อระบบ Kernel ระหว่างระบบปฏิบัติการของเครื่อง Server กับการ์ด Data Center accelerator และทำการ Flash Card เพื่อลง Firmware ให้กับ Card ก่อนการใช้งาน และทำให้ Card เข้ากับสภาพแวดล้อมของระบบปฏิบัติการของเครื่อง

ในการพัฒนาและใช้งาน Model AI บนการ์ด Data Center accelerator บริษัท Xilinx ได้สร้าง Library เครื่องมือของที่จำเป็นสำหรับการใช้งานเทคโนโลยี DPU บน FPGA โดยมี Vitis-AI Development Platform ซึ่งเป็นสภาพแวดล้อมระบบปฏิบัติการจำลองบน Docker เพื่อให้ง่ายต่อติดตั้งและการใช้งานเทคโนโลยี DPU

### 2.6.1. Vitis-AI Development Platform

(Xilinx, Xilinx/Vitis-AI, 2022) Vitis-AI Development Platform เป็น Platform ที่บริษัท Xilinx สร้างขึ้นสำหรับการทำงานประมวลด้าน AI บนอุปกรณ์ของ Xilinx ไม่ว่าจะเป็น Board หรือ บน Alveo Data Center Acceleration Cards, Vitis-AI ถูกออกแบบมาให้มีประสิทธิภาพการใช้งานสูงและง่ายต่อการใช้งาน เพื่อเพิ่มศักยภาพการเร่งการประมวลผล AI ให้ถึงขั้นสูงสุดบน FPGA ของ Xilinx ภายใน Vitis-AI ประกอบไปด้วย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- AI Model Zoo - AI models จำนวนมากที่ผ่านการแปลงทำให้เหมาะสมกับการ์ด FPGA ต่างๆของ Xilinx
- AI Optimizer – เครื่องมือปรับแต่ง deep-learning processor unit (DPU)
- AI Quantizer – เครื่องมือสำหรับการลดขนาด Models ให้สามารถทำงานกับ FPGA ได้
- AI Compiler – เครื่องมือสำหรับการ Compile Models ที่ผ่านการลดขนาดมาแล้วให้เข้ากับโครงสร้างของการ์ด FPGA รุ่นต่างๆ ที่ต้องการ
- AI Profiler – เครื่องมือช่วยในการวิเคราะห์ประสิทธิภาพการ inference Model AI
- AI Library – Libraries เครื่องมือ high-level C++ APIs เพื่อช่วยในการทำงานกับการ์ด FPGA ทั้งบน Board และ Alveo Data Center Acceleration Cards
- DPU – เครื่องมือปรับแต่งประสิทธิภาพ DPU เพื่อให้เหมาะสมกับความต้องการที่หลากหลายของแอปพลิเคชัน



รูปที่ 2.7 Vitis-AI Development Platform

(Xilinx, Xilinx/Vitis-AI, 2022)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.2. DPU Naming

ตารางที่ 2.1 ตารางแสดงตระกูล DPU ต่างๆ

(Xilinx, DPU-Naming, 2023)

DPU Name	Hardware platform
DPUCZDX8G	Zynq® UltraScale+™ MPSoC
DPUCAHX8H	Alveo™ U50, U280 Data Center accelerator cards
DPUCAHX8L	Alveo U50, U280 Data Center accelerator cards
DPUCADF8H	Alveo U200, U250 Data Center accelerator cards
DPUCVDX8G	Versal® ACAP VCK190 evaluation board, Versal AI Core Series
DPUCVDX8H	Versal ACAP VCK5000 evaluation kit

## 2.6.3. Xilinx Alveo U50LV Data Sheet

ตารางที่ 2.2 ตารางแสดงข้อมูลของ Xilinx Alveo U50LV

(Xilinx, Alveo U50 Data Center Accelerator Card Data Sheet, 2022)

Specification	U50 LV Production
Product SKU	A-U50-P00G-LV-G
Total electrical card load	
Thermal cooling solution Weight	75W
Form factor	Passive
Network interface	300g – 325g
Network clock precision	Half height, half length 1x QSFP28 (100 GbE) IEEE 1588
PCIe interface	Gen3 x16
HBM2 total capacity	8 GB
HBM2 bandwidth	316 GB/s
Look-up tables (LUTs)	872K
Registers	1,743K
DSP slices	5,952

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Max. Dist. RAM	24.6 Mb
36 Kb block RAM	1344 (47.3 Mb)
288 Kb UltraRAM	640 (180.0 Mb)
GTY transceivers	20
VCCINT supported	VLOW (0.72V)
Vitis™ Development Environment	
Vitis platform	Yes
Vivado Design Suite	Gen3 x4 XDMA Yes
Target workloads	Machine learning (ML) inference
Qualified for deployment	Yes

#### 2.6.4. Supported Operators and DPU Limitations

ในการทำงานบน DPU นั้นยังมีข้อจำกัดบางประการเนื่องด้วยจากความเป็นเทคโนโลยีใหม่และมีตัวดำเนินการ (Operators) มากมายที่ใช้ในการทำเทคนิคปัญญาประดิษฐ์ Xilinx จึงต้องพัฒนาปรับปรุงตัวดำเนินการหลัก ๆ ที่จำเป็นต่อการใช้งานก่อน และจึงค่อยเพิ่มตัวดำเนินการใหม่ ๆ ให้รองรับมากขึ้น

ในส่วนของตัวดำเนินการที่รองรับสามารถสรุปได้ดังตารางด้านล่างนี้

#### Intrinsic Parameter

- channel\_parallel: 32
- bank\_depth: 4096

#### ตารางที่ 2.3 ตัวดำเนินการที่รองรับ

(Xilinx, Supported Operators and DPU Limitations, 2022)

Typical Operation Type in CNN	Parameters	DPUCAHX8L_ISA0 (U50, U50LV, U280)
conv2d	Kernel size	w, h: [1, 16]
	Strides	w, h: [1, 4]
	Dilation	dilation * input_channel <= 256 * channel_parallel

Typical Operation Type in CNN	Parameters	DPUCAHX8L_ISA0 (U50, U50LV, U280)
	Paddings	pad_left, pad_right: $[0, (\text{kernel\_w} - 1) * \text{dilation\_w}]$ pad_top, pad_bottom: $[0, (\text{kernel\_h} - 1) * \text{dilation\_h}]$
	In Size	$\text{kernel\_w} * \text{kernel\_h} * \text{ceil}(\text{input\_channel} / \text{channel\_parallel}) \leq \text{bank\_depth}$
	Out Size	$\text{output\_channel} \leq 256 * \text{channel\_parallel}$
	Activation	ReLU, ReLU6
	Group* (Caffe)	group==1
depthwise conv2d	Kernel size	w, h: [3]
	Strides	w, h: [1, 2]
	Dilation	$\text{dilation} * \text{input\_channel} \leq 256 * \text{channel\_parallel}$
	Paddings	pad_left, pad_right: $[0, (\text{kernel\_w} - 1) * \text{dilation\_w}]$ pad_top, pad_bottom: $[0, (\text{kernel\_h} - 1) * \text{dilation\_h}]$
	In Size	$\text{kernel\_w} * \text{kernel\_h} * \text{ceil}(\text{input\_channel} / \text{channel\_parallel}) \leq \text{bank\_depth}$
	Out Size	$\text{output\_channel} \leq 256 * \text{channel\_parallel}$
	Activation	ReLU, ReLU6
	Group* (Caffe)	group==input_channel
transposed conv2d	Kernel size	w, h: [1, 16]
	Strides	w, h: [1, 16]
	Paddings	pad_left, pad_right: [0, kernel_w-1] pad_top, pad_bottom: [0, kernel_h-1]
	Out Size	$\text{output\_channel} \leq 256 * \text{channel\_parallel}$
	Activation	ReLU, ReLU6
depthwise transposed conv2d	Kernel size	w, h: [3]
	Strides	w, h: [3]
	Paddings	pad_left, pad_right: [0, kernel_w-1] pad_top, pad_bottom: [0, kernel_h-1]
	Out Size	$\text{output\_channel} \leq 256 * \text{channel\_parallel}$
	Activation	ReLU, ReLU6
max pooling	Kernel size	w, h: {2, 3, 5, 7, 8}
	Strides	w, h: [1, 8]
	Paddings	pad_left, pad_right: [0, kernel_w-1] pad_top, pad_bottom: [0, kernel_h-1]
	Activation	not supported
average pooling	Kernel size	w, h: {2, 3, 5, 7, 8} w==h

Typical Operation Type in CNN	Parameters	DPUCAHX8L_ISA0 (U50, U50LV, U280)
	Strides	w, h: {2, 3, 5, 7, 8} w=h
	Paddings	pad_left, pad_right: [0, kernel_w-1] pad_top, pad_bottom: [0, kernel_h-1]
	Activation	not supported
eltwise	Type	sum
	Input Channel	input_channel <= 256 * channel_parallel
	Activation	ReLU
Concat		Network-specific limitation, which relates to the size of feature maps, quantization results and compiler optimizations.
reorg	Strides	reverse==false : stride ^ 2 * input_channel <= 256 * channel_parallel reverse==true : input_channel <= 256 * channel_parallel
Pad	In Size	input_channel <= 256 * channel_parallel
	mode	"SYMMETRIC" ("CONSTANT" pad(value=0) would be fused into adjacent operators during compiler optimization process)
global pooling		Global pooling will be processed as general pooling with kernel size equal to input tensor size.
InnerProduct, Fully Connected, Matmul		These ops will be transformed into conv2d op
Resize	Scale	NEAREST: ceil(scale/bank_num) * scale * ceil(input_channel/channel_parallel) <= bank_depth BILINEAR: only for 4-D feature maps. This would be transformed into pad and depthwise-transposed-conv2d. TRILINEAR: only for 5-D feature maps. This would be transformed into pad and transposed-conv3d.
	mode	NEAREST, BILINEAR
conv3d		Not supported
depthwise conv3d		Not supported
transposed conv3d		Not supported
depthwise transposed conv3d		Not supported

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

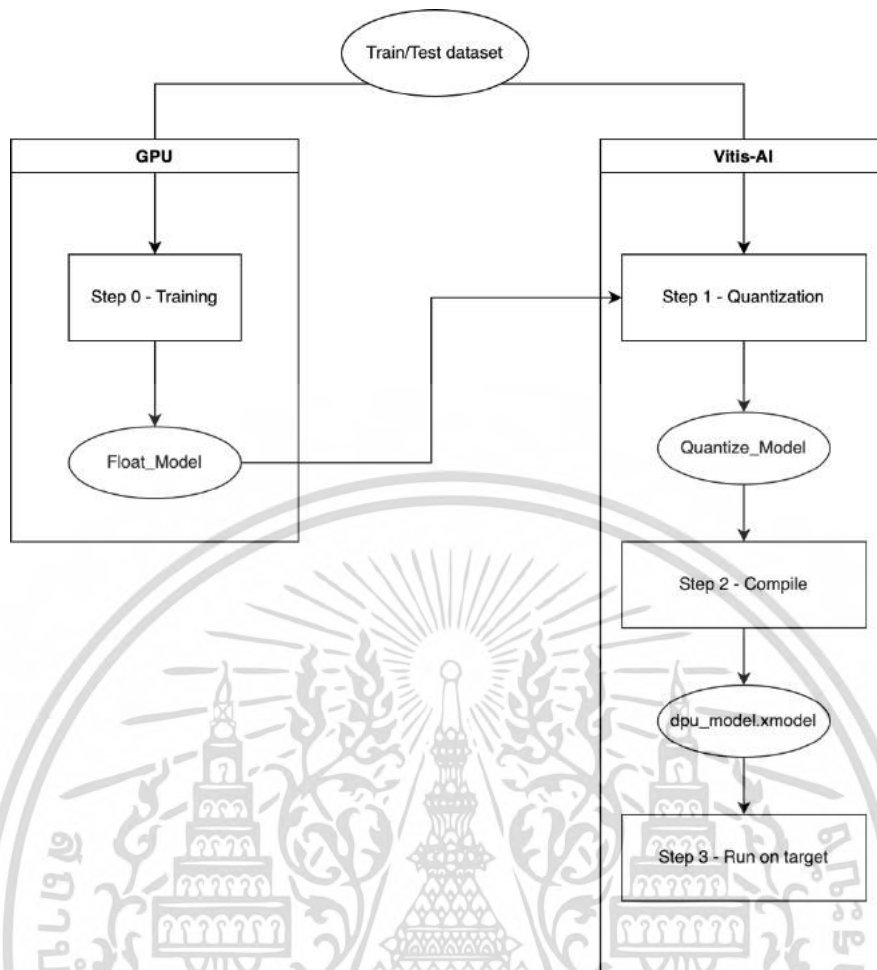
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.5. การพัฒนาและใช้งาน YOLOv4 บนการ์ด FPGA

จากการศึกษาและทดลองต่าง ๆ ผู้จัดทำงานวิจัย สามารถสรุปขั้นตอนในการพัฒนาโมเดล YOLOv4 เพื่อใช้งานบนการ์ด FPGA ได้โดยมีขั้นตอนดังต่อไปนี้

0. Training on GPU: ฝึกสอนโมเดลปัญญาประดิษฐ์บน GPU จากนั้นจะได้ Float Model
1. Quantization: ดำเนินกระบวนการแปลงโมเดลปัญญาประดิษฐ์ที่ทำงานบน GPU ให้เป็นโมเดล fixed-point เนื่องจากโมเดลปัญญาประดิษฐ์ที่ทำงานบน GPU มีค่าน้ำหนักเป็นทศนิยม 32 บิต (32-bit floating point) จึงต้องทำการแปลงโมเดลให้เป็นรูปแบบจำนวนเต็ม 8 บิต (INT8) โดย Vitis AI quantizer สามารถลดความซับซ้อนในการคำนวณโดยไม่ลดความแม่นยำในการทำนายผลได้ โมเดล fixed-point จะใช้ memory bandwidth น้อยจึงทำให้มีความเร็วและประสิทธิภาพสูงกว่าโมเดล floating-point เมื่อ Quantization เรียบร้อยแล้วจะได้โมเดลที่เป็น Fixed-point
2. Compile: เป็นขั้นตอนการ Compile โมเดล Fixed-point ให้เข้ากับการโครงสร้างการทำงานของ DPU แต่ละตระกูล (ตารางที่ 2.1 ตารางแสดงตระกูล DPU ต่างๆ) เมื่อ compile แล้วจะได้ xmodel ที่เป็นโมเดลสำหรับการทำงานบน DPU ตระกูลนั้นๆ
3. Run on Target: เมื่อได้ xmodel แล้วขั้นตอนต่อไปคือการใช้งาน xmodel ผ่านเครื่องมือ VART ที่มีใน Vitis-AI Development Platform

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 Development Flow

ทฤษฎีที่อ้างอิงมาทั้งหมดในข้างต้นนั้น ในงานวิจัยนี้ได้ทางคณะผู้จัดทำได้ใช้เครื่องมือทั้งหมดที่กล่าวมาในข้างต้น 2.1.1 React Native ใช้สำหรับการพัฒนาระบบต้นแบบ Mobile Application ที่เป็นส่วนของระบบ Client พัฒนาระบบต้นแบบ Web Server แบบ 2.2.1 REST API มีการ Deploy ระบบบนสภาพแวดล้อมจำลอง 2.2.2 Docker ซึ่งมีสภาพแวดล้อมจำลองของ Xilinx (2.6.1 Vitis-AI) ที่ใช้สำหรับการเชื่อมต่อและใช้งานกับการ์ด FPGA มีการใช้ฐานข้อมูล 2.2.3 PostgreSQL สำหรับเป็นฐานข้อมูลแบบพื้นฐานใช้ในการเก็บและจัดการกับข้อมูลร้านค้าและสินค้าต่าง ๆ ในห้างสรรพสินค้า

ในการพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการใช้งานบนการ์ด FPGA จะมีขั้นตอนและกระบวนการตามหัวข้อที่ 2.6.5 โดยการพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการทำ Object Detection เพื่อค้นหาสินค้าหรือร้านค้าใช้ 2.4.3 Darknet framework ในการพัฒนาโมเดล มีการปรับแต่งโมเดล และแปลงให้เป็นโมเดลของ 2.4.4 Tensorflow เพื่อให้เข้ากับตัวดำเนินการที่การ์ด Xilinx Alveo U50LV รองรับในตารางที่ 2.3 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### วิธีการดำเนินงานวิจัย

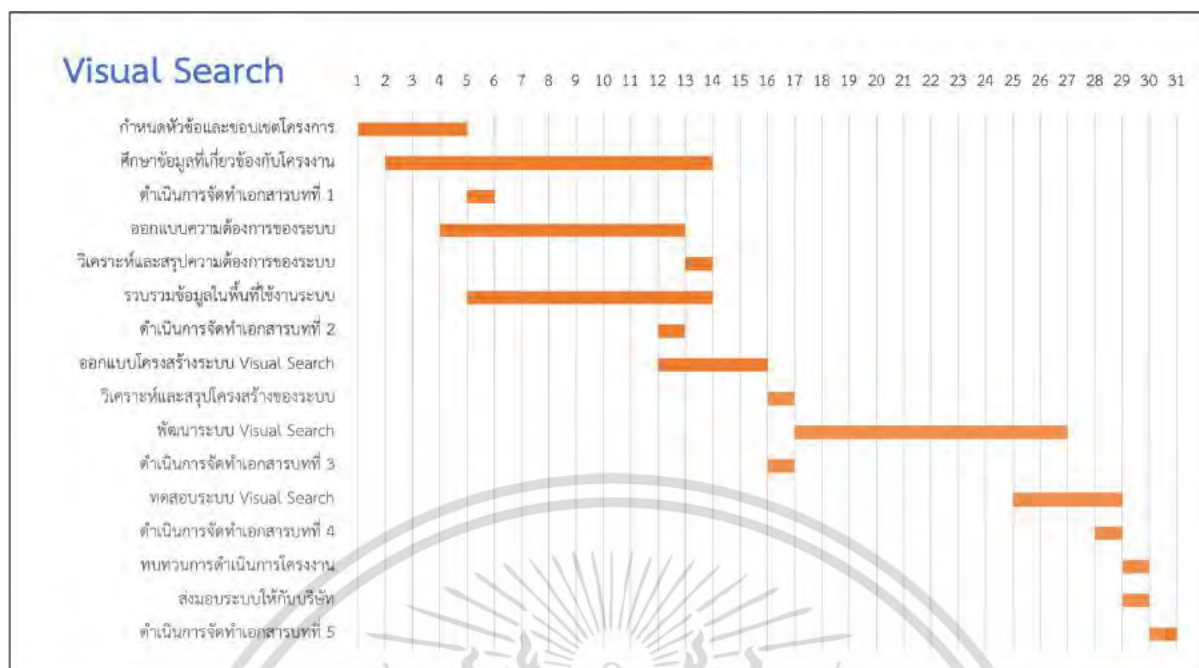
ในจัดทำโครงการวิจัยสหกิจศึกษา การศึกษา พัฒนา และทดสอบการทำงานของระบบ Visual Search ผู้จัดทำงานวิจัยได้วางแผนแบ่งรายละเอียดขั้นตอนวิธีดำเนินการของงานวิจัยออกเป็น 4 ขั้นตอน ได้แก่ 1) ขั้นตอนการศึกษาและรวบรวมข้อมูลของระบบ 2) ขั้นตอนการวิเคราะห์ระบบ 3) ขั้นตอนการออกแบบระบบ และ 4) ขั้นตอนการดำเนินงาน

#### 3.1. แผนการดำเนินการ

ระยะเวลาในการเริ่มเดินนำการสหกิจศึกษาตั้งแต่เดือนมิถุนายน พ.ศ.2565 และสิ้นสุดในเดือน ธันวาคม พ.ศ.2565 โดยมีรายละเอียด ดังนี้

- 1) Project Initiation: ทำการตกลงหัวข้อและขอบเขตการทำงานวิจัยร่วมกันกับบริษัท จากนั้นเรียนรู้ และศึกษาเครื่องมือที่จำเป็นต่อการดำเนินการพัฒนาระบบ
- 2) Requirement Gathering: ทำการสำรวจและรวบรวมความต้องการของระบบ และสกัดออกเป็น ความต้องการของระบบเพื่อดำเนินการออกแบบระบบในขั้นตอนต่อไป
- 3) Design and Analysis: ออกแบบ Mockup ของระบบต้นแบบ
- 4) Development: ดำเนินการพัฒนาระบบตามที่ได้ออกแบบและวางแผนไว้
- 5) Testing and Production: ดำเนินการทดสอบระบบและปรับปรุงแก้ไขข้อผิดพลาดต่างๆ ก่อนส่งมอบงานวิจัยให้กับบริษัทดำเนินการดูแลต่อ
- 6) Closure: ทบทวนความสามารถและความถูกต้องของระบบ และส่งมอบงานวิจัยให้บริษัทดำเนินการต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 กราฟแท่งแสดงแผนดำเนินงานวิจัยแบบย่อ

ตารางที่ 3.1 ตารางแสดงแผนการดำเนินงานวิจัยแบบละเอียด

ลำดับที่	รายละเอียดงาน	เริ่มต้น	สิ้นสุด	ระยะเวลา
Project Initiation		1/6/2565	4/7/2565	33
1	ประชุมหัวข้อและขอบเขตโครงการ	1/6/2565	2/6/2565	1
2	กำหนดหัวข้อและขอบเขตโครงการ	2/6/2565	24/6/2565	22
3	ศึกษาแอปพลิเคชันของห้างสรรพสินค้าต่างๆ	12/6/2565	10/7/2565	28
4	ดำเนินการจัดทำเอกสารที่ 1	29/6/2565	30/6/2565	1
5	เลือกสถานที่ใช้งานระบบ	1/7/2565	4/7/2565	3
Requirement Gathering		25/6/2565	30/8/2565	66
6	ประชุมกำหนดความต้องการของระบบ	25/6/2565	28/6/2565	3
7	ร่างแบบความต้องการของระบบ	28/6/2565	12/7/2565	14
8	ออกแบบความต้องการของระบบ	13/7/2565	22/8/2565	40
9	สำรวจสถานที่ใช้งานระบบ	18/7/2565	27/8/2565	40
10	ทบทวนความต้องการของระบบ	22/8/2565	29/8/2565	7

11	ดำเนินการจัดทำเอกสารบทที่ 2	22/8/2565	27/8/2565	5
12	วิเคราะห์และสรุปความต้องการทั้งหมด	29/8/2565	30/8/2565	1
Design and Analysis		30/8/2565	27/9/2565	28
13	ประชุมกำหนดโครงสร้างของระบบ	30/8/2565	2/9/2565	3
14	ออกแบบโครงสร้างของระบบ	2/9/2565	9/9/2565	7
15	ออกแบบฟังก์ชันการทำงานของ AI ในระบบ	9/9/2565	16/9/2565	7
16	ออกแบบโครงสร้าง API Server	16/9/2565	23/9/2565	7
17	ออกแบบ Mobile Application	16/9/2565	23/9/2565	7
18	ออกแบบโครงสร้างการใช้งานร่วมกับระบบ Navigation	23/9/2565	27/9/2565	4
19	ดำเนินการจัดทำเอกสารบทที่ 3	24/9/2565	30/9/2565	6
Development		27/9/2565	23/11/2565	57
20	ประชุมสรุปโครงสร้างของระบบ	27/9/2565	30/9/2565	3
21	พัฒนาระบบ Object detection และ Image Recognition	30/9/2565	31/10/2565	31
22	พัฒนาระบบ API Server	31/10/2565	15/11/2565	15
23	พัฒนา Mobile Application	5/11/2565	19/11/2565	14
24	พัฒนาส่วนเชื่อมต่อกับระบบ Navigation	17/11/2565	23/11/2565	6
Testing and Production		23/11/2565	24/12/2565	31
25	ออกแบบแผนกรณีทดสอบของระบบ	23/11/2565	27/11/2565	4
26	ทำการ Module Testing	27/11/2565	7/12/2565	10
27	ทดสอบองค์รวมของระบบ	3/12/2565	14/12/2565	11
28	ทำการ Functional Testing	14/12/2565	24/12/2565	10
29	ดำเนินการจัดทำเอกสารบทที่ 4	19/12/2565	24/12/2565	5
Closure		24/12/2565	30/12/2565	6
30	ทบทวนการดำเนินการของโครงการ	24/12/2565	29/12/2565	5

31	ดำเนินการจัดทำเอกสารบทที่ 5	24/12/2565	29/12/2565	5
32	ส่งมอบระบบให้กับบริษัท	28/12/2565	30/12/2565	2

### 3.2. ขั้นตอนการศึกษาและรวบรวมข้อมูลของระบบ

การจัดทำระบบต้นแบบ Visual Search จำเป็นต้องมีการศึกษาและรวบรวมข้อมูลของระบบ เพื่อเพิ่มความถูกต้องและตรงต่อความต้องการของระบบต้นแบบ และง่ายต่อการออกแบบระบบต้นแบบ ทางผู้จัดทำมีวิธีการดำเนินการเก็บรวบรวมข้อมูลของระบบ ดังนี้

#### 3.2.1. เก็บรวบรวมและศึกษาข้อมูลข้อมูลปฐมภูมิ

เนื่องจากระบบต้นแบบ Visual Search เป็นระบบต้นแบบที่คิดค้นขึ้นมาใหม่โดยประยุกต์การใช้งานเทคโนโลยี FPGA เข้ากับการทำ Object Detection จึงทำให้การพัฒนาของระบบจำเป็นต้องศึกษาและรวบรวมข้อมูลในการใช้งานการ์ด FPGA ซึ่งทางบริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) ส่วนปฏิบัติการงานวิจัยดิจิทัล มีบุคลากรที่มีประสบการณ์ในด้านการทำงานกับการ์ด FPGA ดังนั้นผู้วิจัยจึงทำการเก็บรวบรวมข้อมูลจากหน่วยงานที่เกี่ยวข้องโดยวิธีการสัมภาษณ์บุคลากรในหน่วยงานที่เคยทำงานหรือมีประสบการณ์เกี่ยวข้องกับเทคโนโลยี FPGA จากนั้นจึงเก็บรวบรวมข้อมูลเพิ่มเติมโดยการทดลองใช้งานเทคโนโลยี FPGA ด้วยตัวผู้จัดทำเองเพื่อนำความรู้ที่ได้ไปวิเคราะห์ และพัฒนาระบบต่อไป โดยมีขั้นตอนวิธีการดำเนินการดังต่อไปนี้

- 3.2.1.1. ออกแบบคำถามที่จำเป็นและเกี่ยวข้องกับวัตถุประสงค์ของการวิจัย เพื่อให้ได้ข้อมูลที่ครบถ้วนและตรงต่อความต้องการ
- 3.2.1.2. สัมภาษณ์บุคลากรที่มีประสบการณ์การใช้งานเทคโนโลยี FPGA โดยสัมภาษณ์จาก ดร.สมรักษ์ เพชรชาติ ตำแหน่งผู้จัดการส่วนปฏิบัติการงานวิจัยดิจิทัล ซึ่งเป็นผู้มีความชำนาญและประสบการณ์ในการดำเนินงานวิจัยระบบต้นแบบ และเป็นผู้ดูแลงานวิจัยนี้
- 3.2.1.3. วิเคราะห์ข้อมูลที่ได้รับจากการสัมภาษณ์ และสกัดความต้องการของระบบเบื้องต้น
- 3.2.1.4. ทดลองใช้งานเทคโนโลยี FPGA บนเครื่อง Server ที่ศูนย์วิจัยและพัฒนา Brownien Lab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.2. เก็บรวบรวมและศึกษาข้อมูลข้อมูลปฐมภูมิ

เมื่อทำเก็บรวบรวมและศึกษาข้อมูลข้อมูลปฐมภูมิสำเร็จ ทางผู้จัดทำงานวิจัยได้ข้อมูลขั้นพื้นฐานเกี่ยวกับการพัฒนาระบบต้นแบบ ขั้นตอนต่อไปคือการศึกษานวัตกรรมและเทคโนโลยีต่างๆ ที่เกี่ยวข้องกับงานวิจัย เพื่อให้ได้ข้อมูลเชิงลึกที่จำเป็นต่อการพัฒนาระบบ Visual Search

- 3.2.2.1. ศึกษาและค้นคว้าแนวคิดและเทคโนโลยีต่างๆ ที่เกี่ยวข้องกับเทคโนโลยี FPGA โดยสามารถสรุปแนวคิดและเทคโนโลยีที่ศึกษามาได้ในหัวข้อที่ 2.5 เทคโนโลยี Data Center accelerator ของ Xilinx และ 2.6 การพัฒนาและใช้งาน Model AI บนการ์ด Data Center accelerator Cards
- 3.2.2.2. ศึกษาและค้นคว้าเทคนิคการพัฒนาโมเดล AI สำหรับการทำ Object Detection โดยสามารถสรุปการศึกษาได้ในหัวข้อที่ 2.3 Object Detection และ 2.4 การพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการทำ Object Detection
- 3.2.2.3. ศึกษาและค้นคว้าหลักการการพัฒนาแอปพลิเคชันมือถือบน React Native โดยสามารถสรุปการศึกษาได้ในหัวข้อที่ 2.1 การพัฒนา Mobile Application
- 3.2.2.4. ค้นคว้าและรวบรวมหลักการพัฒนาระบบ REST API แบบ Microservice โดยสามารถสรุปการศึกษาได้ในหัวข้อที่ 2.2 การพัฒนา API
- 3.2.2.5. วิเคราะห์ข้อมูลทั้งหมดที่ได้รับจากการค้นคว้าเพื่อใช้ในการออกแบบระบบ Visual Search โดยสามารถวิเคราะห์และสรุปข้อมูลได้เป็นทฤษฎีการพัฒนาและใช้งาน YOLOv4 บนการ์ด FPGA ในหัวข้อที่ 2.6.5 และออกแบบจำลองหน้าจอ Mobile Application ในหัวข้อที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3. ขั้นตอนการวิเคราะห์ระบบ

หลักจากการเก็บรวบรวมและศึกษาข้อมูลที่เป็นต่อการพัฒนาระบบทั้งหมด ทางผู้จัดทำงานวิจัยได้วิเคราะห์ข้อมูลที่ได้รับและสรุปได้ดังนี้

#### 3.3.1. ความต้องการหลักของระบบ (Functional Requirement)

ความต้องการของระบบสามารถแบ่งได้ออกเป็น 3 ส่วนใหญ่ ๆ ดังนี้

##### 1. Mobile Application

- FR01: ระบบแอปพลิเคชันมือถือต้องสามารถแสดงข้อมูลข่าวสารและโปรโมชั่นของห้างสรรพสินค้าทั้งหมด ในหน้าแรกของการใช้งานแอปพลิเคชันมือถือ
- FR02: ระบบแอปพลิเคชันมือถือต้องสามารถค้นหาสินค้าด้วยภาพ ด้วยการส่งรูปภาพสินค้าที่มีอยู่ในระบบในแนวตั้ง ที่ถ่ายรูปด้วยกล้องหลังมือถือหรือเลือกรูปภาพที่มีอยู่ในคลังรูปภาพอย่างใดอย่างหนึ่งเพียงภาพเดียว เข้าไปประมวลการตรวจจับวัตถุในระบบเบื้องหลัง โดยส่งรูปภาพผ่าน API ที่กำหนด รูปภาพที่ส่งไปจะต้องเป็นรูปภาพที่มีนามสกุลไฟล์ JPEG หรือ PNG ไม่จำกัดขนาด มีการแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบ Base64 ก่อนส่งเข้าระบบเบื้องหลัง โดยไม่มีการเก็บหรือบันทึกข้อมูลรูปภาพไว้ในเครื่องโทรศัพท์มือถือ
- FR03: ระบบแอปพลิเคชันมือถือต้องสามารถค้นหาข้อมูลสินค้าในระบบด้วยตัวอักษรชื่อสินค้าผ่าน API ของระบบเบื้องหลัง โดยค้นหาแบบ Insensitive case และสามารถค้นหาด้วยตัวอักษรที่มีส่วนประกอบรูปแบบเดียวกันกับชื่อสินค้า
- FR04: ระบบแอปพลิเคชันมือถือต้องมีความสามารถในการแสดงรายการรายการสินค้าที่เกี่ยวข้องกับการค้นหาสินค้า ทั้งการค้นหาสินค้าด้วยรูปภาพและการค้นหาสินค้าด้วยตัวอักษรชื่อสินค้า โดยในรายการสินค้าประกอบไปด้วย รูปสินค้า ชื่อสินค้า ชื่อร้านค้าที่มีสินค้านั้น ราคา และราคาโปรโมชั่นถ้ามี
- FR05: ระบบแอปพลิเคชันมือถือต้องสามารถค้นหาข้อมูลร้านค้าในระบบจากตัวอักษรชื่อร้านผ่าน API ของระบบเบื้องหลัง โดยค้นหาแบบ Insensitive case และสามารถค้นหาด้วยตัวอักษรที่มีส่วนประกอบรูปแบบเดียวกันกับชื่อร้านค้า
- FR06: ระบบแอปพลิเคชันมือถือต้องมีความสามารถในการแสดงรายการร้านค้าที่เกี่ยวข้องกับการค้นหาร้านค้าด้วยตัวอักษรชื่อร้านค้า โดยในรายการร้านค้าประกอบไปด้วย โลโก้ร้านค้า และชื่อร้านค้า
- FR07: ผู้ใช้งานระบบสามารถเลือกดูรายละเอียดของร้านค้าได้แก่ ชื่อร้านค้า โลโก้ร้านค้า รูปภาพร้านค้า สินค้าในร้านค้าทั้งหมด และรายการส่งเสริมการขายถ้ามี โดยสามารถเลือกร้านค้าที่ต้องการผ่านการค้นหาสินค้าด้วยภาพ หรือค้นหาสินค้าด้วยตัวอักษรชื่อสินค้า หรือค้นหาร้านค้าด้วยตัวอักษรชื่อร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- FR08: ระบบแอปพลิเคชันมือถือต้องสามารถส่งข้อมูลร้านค้าที่ผู้ใช้งานเลือกดูรายละเอียดไปให้ระบบ Navigation Path ในรูปแบบตัวเลขจำนวนเต็มที่เป็น ID ร้านค้าของระบบ Navigation Path เพื่อให้ระบบ Navigation Path ประมวลผลแสดงตำแหน่งบนแผนที่ของร้านค้าปลายทางที่ผู้ใช้งานเลือก โดยจะต้องผ่านขั้นตอนการดูรายละเอียดร้านค้าก่อน
- FR09: ระบบแอปพลิเคชันมือถือต้องสามารถระบุตำแหน่งร้านค้าต้นทางในห้างสรรพสินค้า เมื่อระบบ Navigation Path ร้องขอ ด้วยการส่งรูปภาพหน้าร้านค้าในห้างสรรพสินค้าแนวตั้ง ที่ถ่ายรูปด้วยกล้องหลังมือถือหรือเลือกรูปภาพที่มีอยู่ในคลังรูปภาพอย่างใดอย่างหนึ่งเพียงภาพเดียว เข้าไปประมวลผลการตรวจจับวัตถุในระบบเบื้องหลัง โดยส่งรูปภาพผ่าน API รูปภาพที่ส่งไปจะต้องเป็นรูปภาพที่มีนามสกุลไฟล์ JPEG หรือ PNG ไม่จำกัดขนาด มีการแปลงข้อมูลรูปภาพให้อยู่ในรูปแบบ Base64 ก่อนส่งเข้าระบบเบื้องหลัง ไม่มีการเก็บข้อมูลรูปภาพนั้นไว้ในเครื่องโทรศัพท์มือถือ
- FR10: ระบบแอปพลิเคชันมือถือต้องมีความสามารถในการแปลงข้อมูลผลลัพธ์การตรวจจับหน้าร้านค้าต้นทางที่ได้รับจากระบบเบื้องหลัง ให้อยู่ในรูปแบบจำนวนเต็มที่เป็น ID ร้านค้าของระบบ Navigation Path และส่ง ID ร้านค้าต้นทาง ไปให้ระบบ Navigation Path เพื่อให้ระบบประมวลผลแสดงเส้นทางจากร้านค้าต้นทางไปยังร้านค้าปลายทาง
- FR11: ระบบแอปพลิเคชันมือถือต้องมีความสามารถในการแจ้งข้อผิดพลาดในการตรวจจับวัตถุในกรณีที่โมเดล AI ไม่สามารถตรวจจับวัตถุได้ หรือระบบเบื้องหลังมีปัญหา โดยแสดงเป็นข้อความข้อผิดพลาดที่เกิดขึ้นบนหน้าจอ และสามารถย้อนกลับเพื่อดำเนินการใหม่อีกครั้งได้
2. ระบบเบื้องหลัง (Back-end)
- FR12: ระบบเบื้องหลังต้องมีความสามารถในการรับรูปภาพที่ส่งมาจากแอปพลิเคชันมือถือในรูปแบบ Base64 ได้
- FR13: ระบบเบื้องหลังต้องมีความสามารถแปลงข้อมูล Base64 ให้เป็นไฟล์รูปภาพที่มีนามสกุลไฟล์ JPEG หรือ PNG ตามนามสกุลไฟล์ที่ได้รับจากระบบแอปพลิเคชันมือถือ
- FR14: ระบบเบื้องหลังต้องสามารถบันทึกรูปภาพที่ได้รับจากระบบแอปพลิเคชันมือถือ บันทึกไว้ใน Folder ของการประมวลผลรูปภาพของแต่ละประเภทการตรวจจับ โดยตั้งชื่อไฟล์รูปภาพเป็น ID ตัวอักษรที่ไม่ซ้ำกันของแต่ละ Session การติดต่อนั้น
- FR15: ระบบเบื้องหลังต้องมีความสามารถในการกำหนดระยะเวลาในการติดต่อแต่ละ Session เมื่อเกินระยะเวลาที่กำหนดแล้วจะต้องตอบกลับด้วยข้อความแจ้งสถานะหมดเวลาการติดต่อ
- FR16: ระบบเบื้องหลังต้องสามารถรอผลลัพธ์การประมวลผลการตรวจจับจาก Python Application ภายในระยะเวลาที่กำหนด
- FR17: ระบบเบื้องหลังต้องสามารถอ่านผลลัพธ์จากไฟล์ txt ที่ตรงกับชื่อ ID ของ Session นั้นๆ โดยไฟล์ผลลัพธ์ในหนึ่งบรรทัดจะประกอบไปด้วยข้อมูลชื่อคลาสที่เจอ และค่าความมั่นใจที่จะเป็นคลาสนั้น ซึ่งอาจมีผลลัพธ์มากกว่า 1 บรรทัดหรือไม่มีเลยก็ได้

เอกสารนี้เป็นเอกสารหนึ่งของระบบสารสนเทศที่ใช้ในการดำเนินงาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- FR18: ระบบเบื้องหลังต้องสามารถค้นหาสินค้าจากชื่อคลาสสินค้าที่เจอในไฟล์ผลลัพธ์ทุกบรรทัด และส่งผลลัพธ์การค้นหา เป็นรายการสินค้ากลับไปให้ระบบแอปพลิเคชันมือถือที่ติดต่อมาใน Session นั้นๆ ในรูปแบบ JSON
- FR19: ระบบเบื้องหลังต้องสามารถค้นหาร้านค้าจากชื่อคลาสร้านค้าที่เจอในไฟล์ผลลัพธ์บรรทัดแรกสุด และส่งผลลัพธ์การค้นหา เป็นข้อมูลร้านค้ากลับไปให้ระบบแอปพลิเคชันมือถือที่ติดต่อมาใน Session นั้นๆ ในรูปแบบ JSON
- FR20: ระบบเบื้องหลังต้องสามารถตอบกลับข้อผิดพลาดจากการตรวจจับไม่พบในกรณีที่ไม่มีชื่อคลาสในไฟล์ผลลัพธ์
- FR21: ระบบเบื้องหลังต้องมีความสามารถในการสร้างข้อมูลร้านค้าและสินค้า ในฐานข้อมูลที่เป็นฐานข้อมูลประเภทการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS)
- FR22: ระบบเบื้องหลังต้องมีความสามารถในการอ่านข้อมูลร้านค้าและสินค้า ในฐานข้อมูลที่เป็นฐานข้อมูลประเภทการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS)
- FR23: ระบบเบื้องหลังต้องมีความสามารถในการอัปเดตข้อมูลร้านค้าและสินค้า ในฐานข้อมูลที่เป็นฐานข้อมูลประเภทการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS)
- FR24: ระบบเบื้องหลังต้องมีความสามารถในการลบข้อมูลร้านค้าและสินค้า ในฐานข้อมูลที่เป็นฐานข้อมูลประเภทการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (ORDBMS)

### 3. โมเดลตรวจจับสินค้า หน้าร้านค้า และ Python Application

- FR25: โมเดลตรวจจับสินค้าจะต้องเป็นโมเดลปัญญาประดิษฐ์ประเภท Object Detection ที่มีความสามารถในการทำนายผลลัพธ์ตรวจจับสินค้าจากรูปภาพสินค้าในห้างสรรพสินค้ามีจำนวนคลาสที่สามารถตรวจจับรูปสินค้ามากกว่า 50 คลาส มีค่าเฉลี่ยความแม่นยำสูง (mAP) กว่า 0.5 ขึ้นไป โดยจะต้องใช้โครงสร้างโครงข่ายประสาทแบบคอนโวลูชัน (CNN Architecture) ที่ Vitis-AI 2.5 รองรับและสามารถแปลงค่าน้ำหนักโมเดลจาก Floating-point เป็น Fixed-Point ได้เพื่อรองรับกับการทำงานบนการ์ด FPGA
- FR26: โมเดลตรวจจับหน้าร้านค้าจะต้องเป็นโมเดลปัญญาประดิษฐ์ประเภท Object Detection ที่มีความสามารถในการทำนายผลลัพธ์ตรวจจับหน้าร้านค้าจากรูปภาพหน้าร้านค้าในห้างสรรพสินค้ามีจำนวนคลาสที่สามารถตรวจจับรูปหน้าร้านค้ามากกว่า 20 คลาส มีค่าเฉลี่ยความแม่นยำ (mAP) สูงกว่า 0.5 ขึ้นไป โดยจะต้องใช้โครงสร้างโครงข่ายประสาทแบบคอนโวลูชัน (CNN Architecture) ที่ Vitis-AI 2.5 รองรับและสามารถแปลงค่าน้ำหนักโมเดลจาก Floating-point เป็น Fixed-Point ได้เพื่อรองรับกับการทำงานบนการ์ด FPGA
- FR27: Python Application ต้องสามารถโหลดโครงสร้างและค่าน้ำหนักของโมเดลปัญญาประดิษฐ์ Fixed-Point เข้าประมวลผลในการ์ด FPGA ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

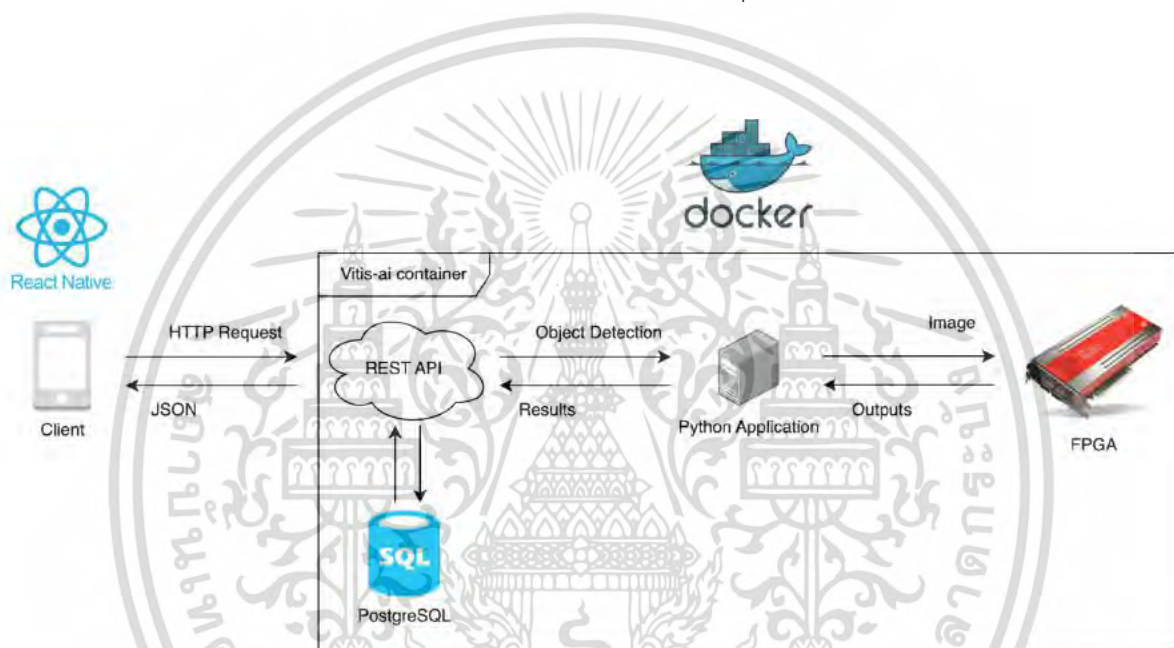
- FR28: Python Application ต้องสามารถอ่านข้อมูลรูปภาพทั้งนามสกุลไฟล์ JPEG หรือ PNG ใน Folder ของการประมวลผลรูปภาพของแต่ละประเภทการตรวจจับได้ โดยมีการเรียงคิวการอ่านรูปภาพเข้าประมวลผลตามลำดับเวลาที่ระบบเบื่องหลังสร้างรูปภาพ
- FR29: Python Application ต้องสามารถเตรียมรูปภาพเบื้องต้นก่อนการตรวจจับสินค้าหรือหน้าร้านค้าได้ โดยจะต้องปรับรูปภาพให้อยู่ในขนาดตามขนาดที่โมเดลกำหนด และทำข้อมูลรูปภาพให้อยู่ในรูปแบบ NumPy array
- FR30: Python Application ต้องสามารถนำรูปภาพที่ผ่านการเตรียมเบื้องต้นเข้าไปประมวลผลการตรวจจับสินค้าหรือหน้าร้านค้า โดยสามารถนำเข้ารูปภาพเพื่อประมวลผลได้พร้อมกันสูงสุด 5 รูปต่อการประมวลผล 1 ครั้ง
- FR31: Python Application ต้องสามารถถอดรหัสผลลัพธ์การตรวจจับสินค้าหรือหน้าร้านค้าให้เป็นชื่อคลาสที่ตรวจจับเจอจากการประมวลผลได้ และจะต้องมีความมั่นใจมากกว่า 0.5 (มากกว่า 50%)
- FR32: Python Application ต้องสามารถสร้างไฟล์ผลลัพธ์การตรวจจับเป็น txt ไฟล์ที่มีชื่อตรงกันกับชื่อของรูปภาพก่อนนำเข้าประมวลผลการตรวจจับสินค้าหรือหน้าร้านค้า และลบรูปภาพนั้นหลังจากประมวลผลเสร็จ โดยผลลัพธ์ในหนึ่งบรรทัดจะประกอบไปด้วยชื่อคลาสที่ตรวจจับเจอ ค่าความมั่นใจ และเรียงลำดับบรรทัดตามค่าความมั่นใจจากมากไปน้อย
- 3.3.2. ความต้องการเสริมของระบบ (Non-functional Requirement)
- NFR01:ระบบต้องใช้เวลาในการประมวลผลการตรวจจับสินค้าหรือหน้าร้านค้าไม่เกิน 1 วินาทีต่อหนึ่งครั้ง
- NFR02:ระบบต้องมีความสามารถในการใช้งานบนโทรศัพท์มือถือระบบปฏิบัติการทั้ง iOS และ Android
- NFR03:ระบบต้องมีการออกแบบให้เข้าใจง่ายต่อการใช้งานมีคำอธิบายฟังก์ชันที่ชัดเจนไม่ซับซ้อน
- NFR04:ระบบจะไม่มีเก็บข้อมูลจากผู้ใช้งาน เพื่อความปลอดภัยของข้อมูลส่วนบุคคลของผู้ใช้งาน
- NFR05:ระบบต้องมีคู่มือคำแนะนำสำหรับผู้ดูแลระบบเพื่อให้ง่ายต่อการใช้งานฟังก์ชันการจัดการข้อมูลในฐานข้อมูล
- NFR06:ระบบต้องมีคู่มือรายละเอียดระบบสำหรับการบำรุงรักษาระบบ และการต่อยอดและพัฒนาระบบในขอบเขตอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4. ขั้นตอนการออกแบบระบบและขั้นตอนการดำเนินงาน

#### 3.4.1. โครงสร้างสถาปัตยกรรมหลักของระบบต้นแบบ

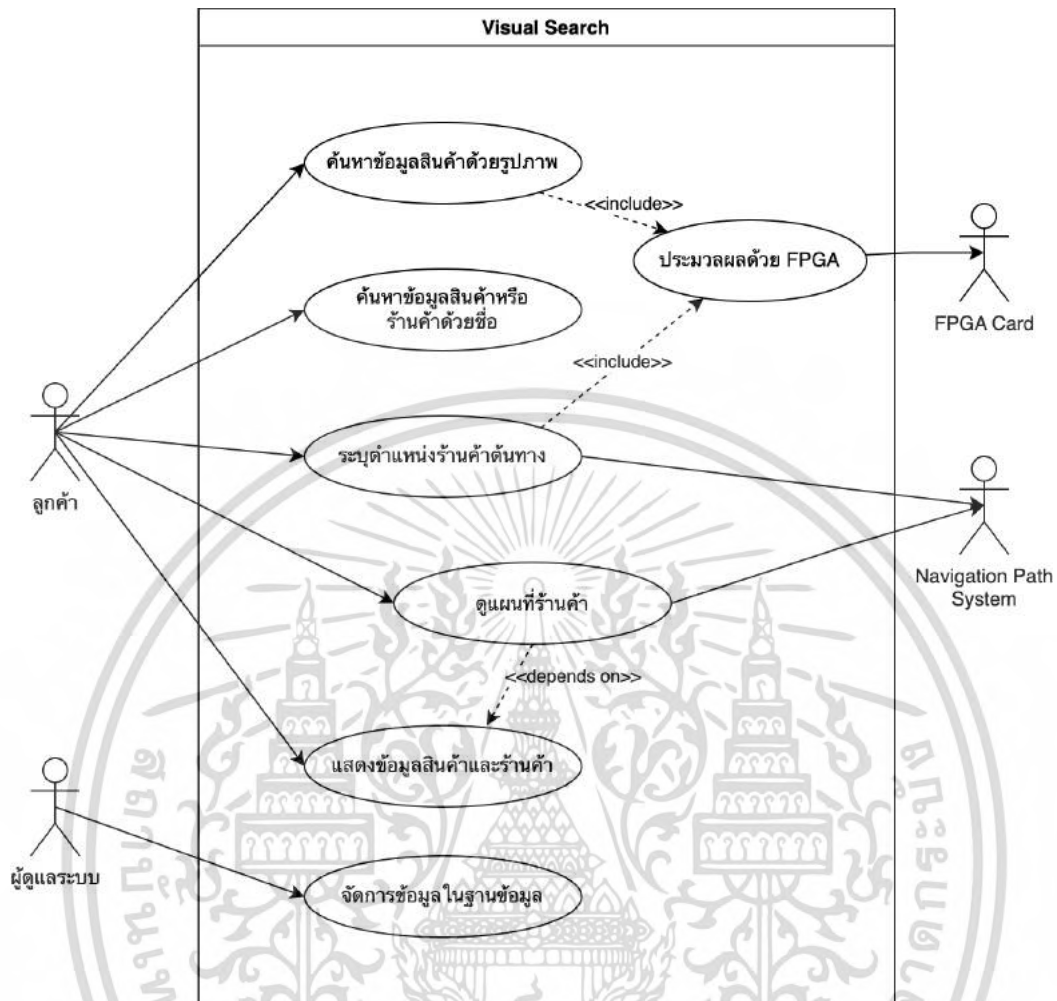
ในการจัดทำระบบต้นแบบ Visual Search คณะผู้จัดทำได้ออกแบบโครงสร้างสถาปัตยกรรมของระบบต้นแบบ โดยให้ Client ที่พัฒนาด้วย React Native ส่งข้อมูลคำร้องขอ HTTP Method ต่างๆ กับ Server ที่พัฒนาด้วย Node.JS โดยมีการติดต่อกับ Database ใน docker container และตอบกลับคำขอข้อมูลในรูปแบบ JSON เมื่อ Client ส่งคำร้องขอการประมวลผลการตรวจจับวัตถุในรูปภาพ Server จะส่งต่อรูปภาพนั้นให้กับ Python Application และมีดำเนินการตรวจจับวัตถุด้วยโมเดล AI บน FPGA ดังนี้



รูปที่ 3.2 สถาปัตยกรรมของระบบ Visual Search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2. Use Case Diagram



รูปที่ 3.3 Use Case Diagram

Use Case Narrative

จาก Use Case Diagram ในรูปที่ 3.2 สามารถสร้างเป็นกลุ่มตารางแสดงรายละเอียดของแต่ละ Use case ได้ดังนี้

ตารางที่ 3.2 ตารางแสดงคำบรรยาย Use Case ค้นหาข้อมูลสินค้าด้วยรูปภาพ

Use-Case Name:	ค้นหาข้อมูลสินค้าด้วยรูปภาพ
Use-Case ID:	VS-01
Primary Actor:	ลูกค้า
Other Participating Actors:	-
Other Interested Stakeholders:	ร้านค้าในห้างสรรพสินค้า

การที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Precondition:	ผู้ใช้งานต้องมีสินค้าที่มีอยู่ในระบบ นำเข้าข้อมูลสินค้าเป็นรูปภาพสินค้าที่ถ่ายจากกล้องหลังใน แอปพลิเคชันมือถือ หรือเลือกรูปภาพสินค้าจากคลังรูปภาพใน โทรศัพท์
Description:	ผู้ใช้งานสามารถค้นหาสินค้าด้วยรูปภาพจากแอปพลิเคชันมือถือ สามารถถ่ายรูปสินค้าด้วยกล้องหลังหรือเลือกจากคลังรูปภาพก็ได้ โดยส่งรูปภาพที่ได้จากแอปพลิเคชันมือถือเข้าไปในเครื่อง Server ผ่าน API และดึงรูปภาพไปทำ Preprocess รูปภาพเบื้องต้นก่อนส่ง เข้าไปประมวลผล Object Detection ด้วยการ์ด FPGA บนเครื่อง Server และส่ง Output ที่เป็นข้อมูลสินค้าให้แอปพลิเคชันมือถือ แสดงรายการสินค้าที่พบ

ตารางที่ 3.3 ตารางแสดงคำบรรยาย Use Case ค้นหาข้อมูลสินค้าหรือร้านค้าด้วยชื่อ

Use-Case Name:	ค้นหาข้อมูลสินค้าหรือร้านค้าด้วยชื่อ
Use-Case ID:	VS-02
Primary Actor:	ลูกค้า
Other Participating Actors:	-
Other Interested Stakeholders:	ร้านค้าในห้างสรรพสินค้า
Precondition:	ผู้ใช้งานต้องนำเข้าข้อมูลชื่อสินค้าหรือร้านค้า
Description:	Use Case นี้ อธิบายถึงกิจกรรมการค้นหาข้อมูลสินค้าและร้านค้า ด้วยชื่อในระบบ Visual Search โดยผู้ใช้งานสามารถกรอกชื่อ ร้านค้าหรือสินค้าที่ต้องการค้นหาในระบบ และแสดงผลลัพท์รายการ การค้นหาที่เกี่ยวข้องทั้งหมดให้กับผู้ใช้งานและสามารถเลือกเข้าไปดู รายละเอียดข้อมูลสินค้าหรือร้านค้าที่ต้องการได้

ตารางที่ 3.4 ตารางแสดงคำบรรยาย Use Case ระบุตำแหน่งที่อยู่ผู้ใช้งาน

Use-Case Name:	ระบุตำแหน่งร้านค้าต้นทาง
Use-Case ID:	VS-03
Primary Actor:	รถที่ส่งวนไว้สำหรับลูกค้า เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Other Participating Actors:	ไม่มีให้ Navigation Path System ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Other Interested Stakeholders:	ร้านค้าในห้างสรรพสินค้า
Precondition:	ลูกค้าต้องอยู่ในห้างสรรพสินค้า หรือมีรูปภาพหน้าร้านค้าในห้างสรรพสินค้า ลูกค้าต้องใช้งานฟังก์ชันการดูเส้นทางร้านค้าปลายทางก่อน และมีการร้องขอการเรียกใช้งานฟังก์ชันการระบุตำแหน่งร้านค้าต้นทางจากระบบ Navigation Path
Description:	Use Case นี้ อธิบายถึงกิจกรรมการระบุตำแหน่งร้านค้าต้นทาง ด้วยรูปภาพหน้าร้านค้าที่ลูกค้าอยู่ โดยสามารถถ่ายรูปหน้าร้านค้าด้วยกล้องหลังหรือเลือกจากคลังรูปภาพก็ได้ จากนั้นส่งรูปภาพที่ได้จากแอปพลิเคชันมือถือเข้าไปในเครื่อง Server ผ่าน API และดึงรูปภาพไปทำ Preprocess รูปภาพเบื้องต้นก่อนส่งเข้าไปประมวลผล Object Detection ด้วยการ์ด FPGA บนเครื่อง Server และส่ง Output ที่เป็น ID ร้านค้าต้นทางให้ระบบ Navigation Path ประมวลผลนำทางจากร้านค้าต้นทางไปยังร้านค้าปลายทาง

ตารางที่ 3.5 ตารางแสดงคำบรรยาย Use Case ประมวลผลด้วย FPGA

Use-Case Name:	ประมวลผลด้วย FPGA
Use-Case ID:	VS-04
Primary Actor:	FPGA Card
Other Participating Actors:	-
Other Interested Stakeholders:	-
Precondition:	ต้องมีโมเดล AI ที่โครงสร้างโครงข่ายที่ FPGA Card รองรับและสามารถแปลงโมเดล AI จาก Floating-point ให้เป็น Fix-point รูปภาพที่จะผ่านการประมวลผลต้องผ่านกระบวนการปรับขนาดรูปภาพให้อยู่ในขนาดที่โมเดลต้องการ เช่น 416x416 Pixel หรือ 512x512 Pixel
Description:	Use Case นี้ อธิบายถึงกิจกรรมการประมวลผลโมเดล AI ด้วยการ์ด FPGA ของ Xilinx สามารถประมวลได้ทั้งการตรวจจับรูปภาพสินค้า และรูปภาพหน้าร้านค้า โดยใช้ Docker ของ Vitis-AI ซึ่งมีเครื่องมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	สำหรับการใช้งานการ์ด FPGA ของ Xilinx โดยใช้ภาษา Python ในการสั่งการใช้งานการ์ด FPGA ผ่าน High level API
--	---

ตารางที่ 3.6 ตารางแสดงคำบรรยาย Use Case ดูแผนที่ร้านค้า

Use-Case Name:	ดูแผนที่ร้านค้า
Use-Case ID:	VS-05
Primary Actor:	ลูกค้า
Other Participating Actors:	Navigation Path System
Other Interested Stakeholders:	ร้านค้าในห้างสรรพสินค้า
Precondition:	ลูกค้าต้องเลือกรายละเอียดหน้าร้านค้าก่อนดูแผนที่ร้านค้า
Description:	Use Case นี้ อธิบายถึงกิจกรรมการดูแผนที่ร้านค้าในห้างสรรพสินค้าที่ลูกค้าเลือก โดยหลังจากที่ผู้ใช้งานเลือกรายละเอียดสินค้าที่ต้องการ ลูกค้าสามารถใช้งานฟังก์ชันการดูแผนที่ร้านค้าผ่านระบบ Navigation Path โดยจะส่งเป็น ID ร้านค้า ให้ระบบ Navigation Path แสดงตำแหน่งในรูปแบบปักหมุดร้านค้าบนแผนที่ในห้าง

ตารางที่ 3.7 ตารางแสดงคำบรรยาย Use Case ดูข้อมูลสินค้าและร้านค้า

Use-Case Name:	แสดงข้อมูลสินค้าและร้านค้า
Use-Case ID:	VS-06
Primary Actor:	ลูกค้า
Other Participating Actors:	-
Other Interested Stakeholders:	ร้านค้าในห้างสรรพสินค้า
Precondition:	ลูกค้าต้องค้นหาหรือเลือกข้อมูลของสินค้าในระบบ Visual Search ที่ต้องการจะดูรายละเอียด
Description:	ลูกค้าสามารถดูรายละเอียดสินค้าและร้านค้าที่มีอยู่ในระบบฐานข้อมูลของ Visual Search ซึ่งประกอบไปด้วยข้อมูล ชื่อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ชื่อร้านค้า ที่อยู่ร้านค้า คำอธิบายสินค้า ราคาสินค้า โปรโมชั่นของสินค้า และขอแผนที่ยี่ห้อร้านค้ากับระบบ Navigation Path ได้
--	---

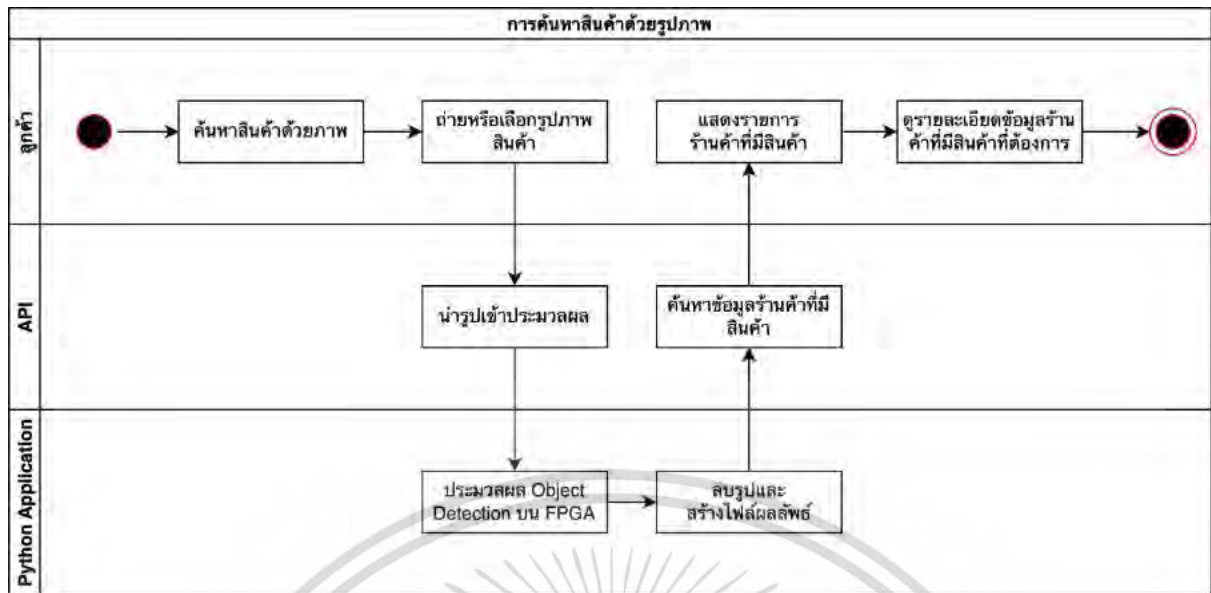
ตารางที่ 3.8 ตารางแสดงคำบรรยาย Use Case ระบุตำแหน่งที่อยู่ผู้ใช้งาน

Use-Case Name:	จัดการข้อมูลสินค้าและร้านค้า
Use-Case ID:	VS-07
Primary Actor:	ผู้ดูแลระบบ
Other Participating Actors:	-
Other Interested Stakeholders:	ร้านค้าในห้างสรรพสินค้า
Precondition:	ผู้ดูแลระบบต้องมีข้อมูลที่จำเป็นของร้านค้าหรือสินค้าที่ต้องการจัดการ
Description:	Use Case นี้ อธิบายถึงกิจกรรมการจัดการข้อมูลสินค้าและร้านค้า ในฐานะข้อมูลของระบบ Visual Search การจัดการข้อมูลในฐานข้อมูลจะมีเพียงผู้ดูแลระบบเท่านั้นที่สามารถทำได้ โดยผู้ดูแลระบบจะสามารถจัดการข้อมูลในระบบได้ผ่าน API ระบบเบื้องหลังในรูปแบบที่กำหนดของแต่ละ API

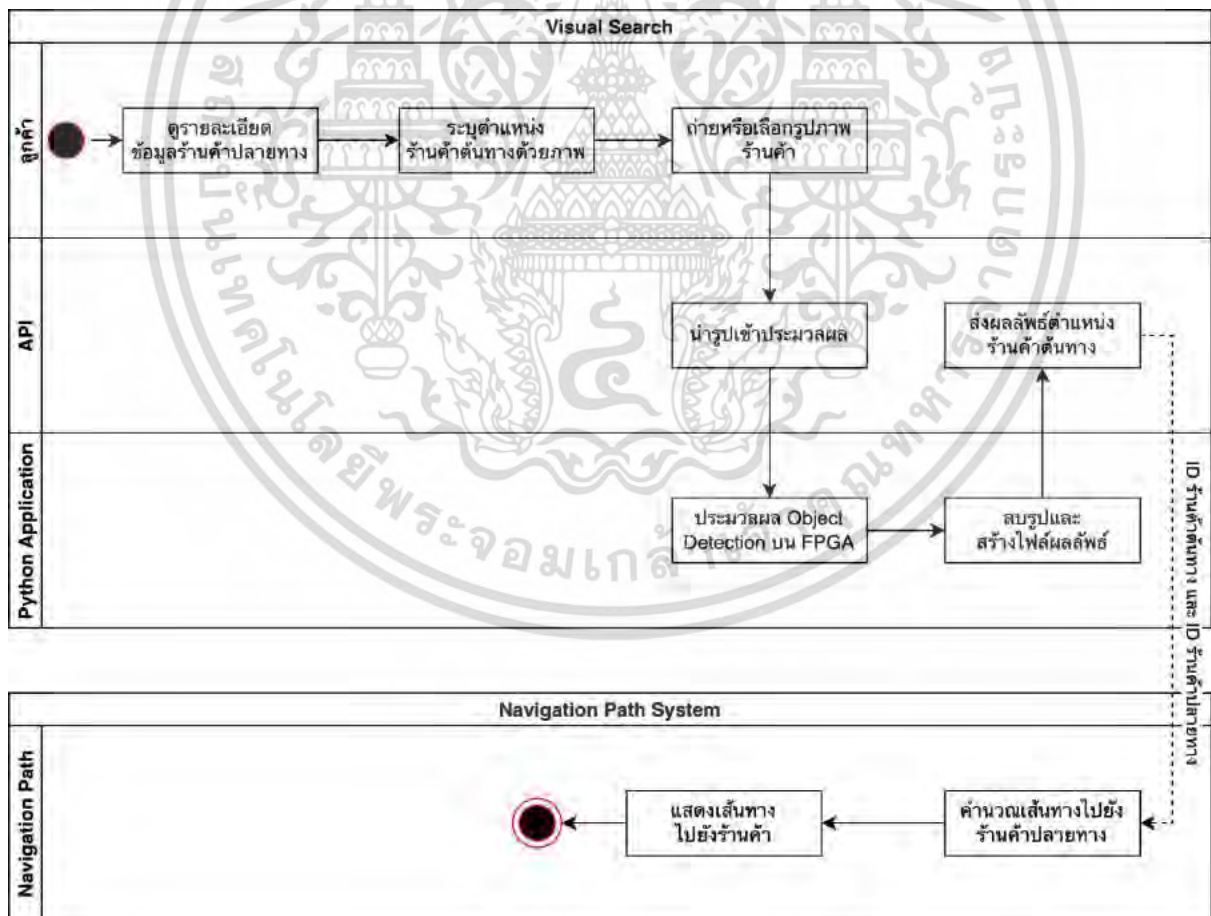
### 3.4.3. Process Flow Diagram

เป็นแผนภาพสำหรับแสดงขั้นตอนการทำงานของกระบวนการหนึ่ง ๆ เพื่อเป็นเครื่องมือที่ช่วยในการทำ ความเข้าใจและวิเคราะห์โปรเจคได้อย่างรวดเร็วและแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

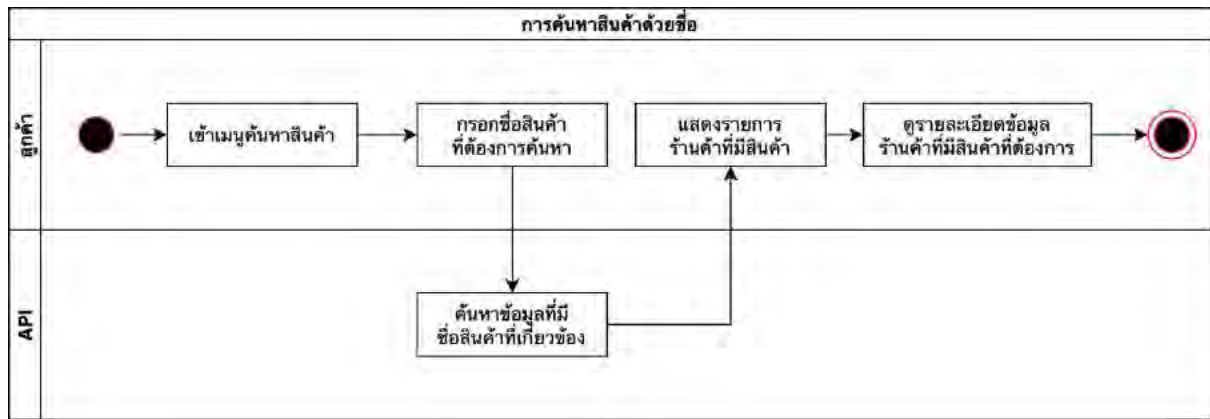


รูปที่ 3.4 แผนภาพแสดงกระบวนการค้นหาสินค้าด้วยรูปภาพ

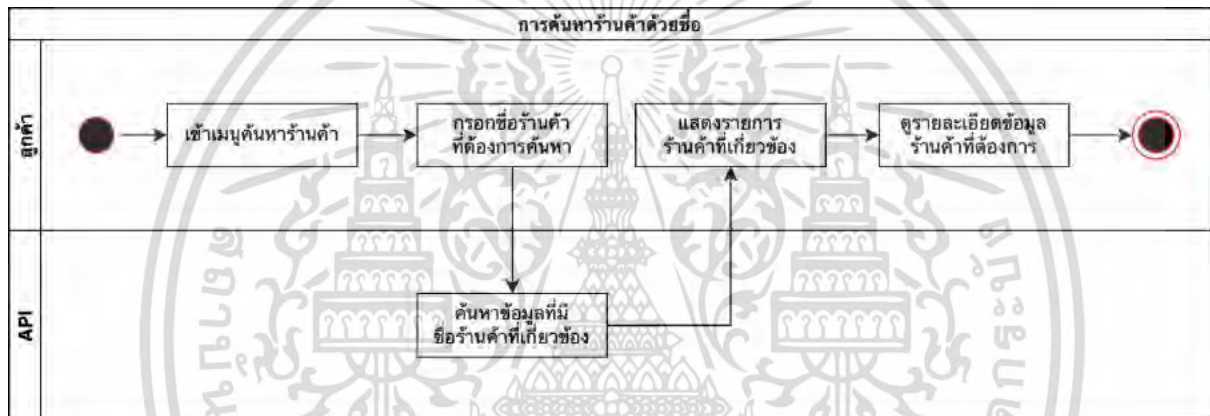


รูปที่ 3.5 แผนภาพแสดงกระบวนการระบุตำแหน่งร้านค้าด้วยรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แผนภาพแสดงกระบวนการค้นหาสินค้า



รูปที่ 3.7 แผนภาพแสดงกระบวนการค้นหาร้านค้า

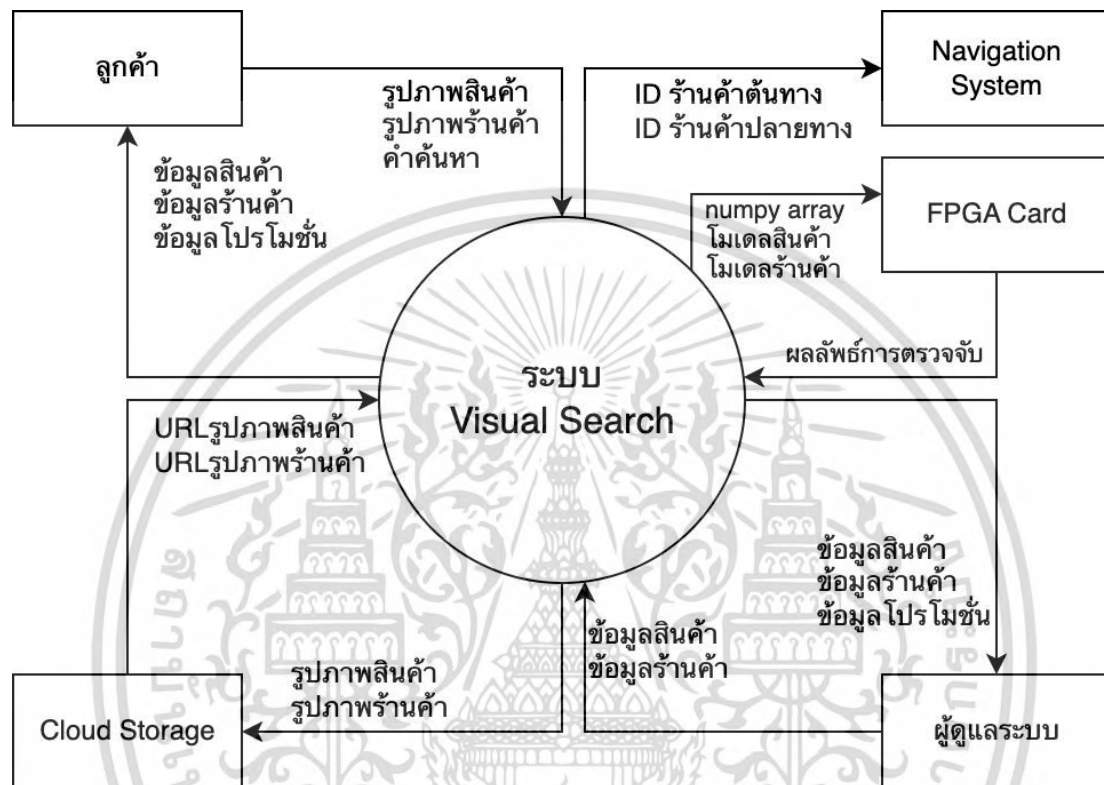


รูปที่ 3.8 แผนภาพแสดงกระบวนการจัดการข้อมูลในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.4. Context Diagram

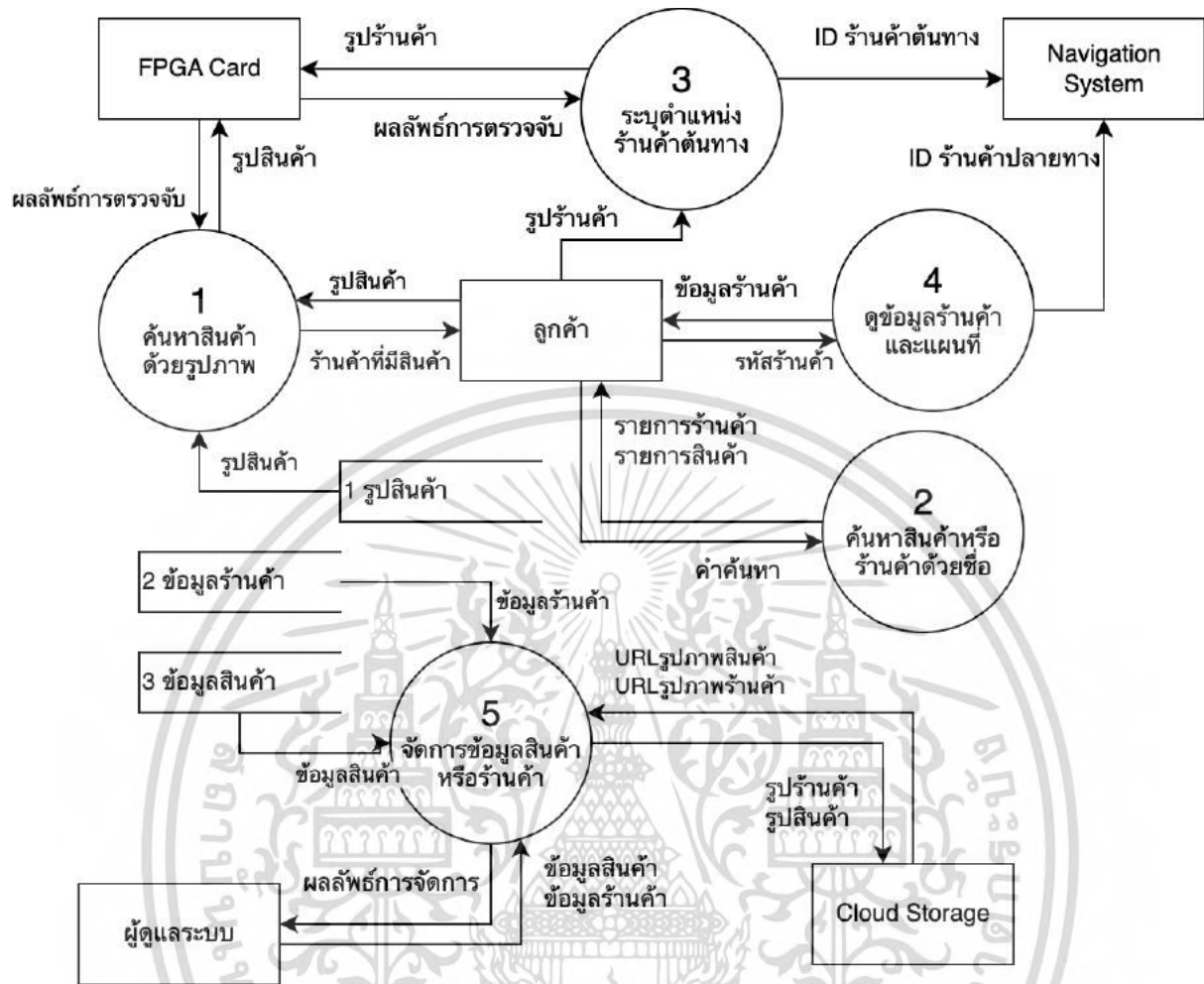
เป็นแผนภาพสำหรับแสดงขอบของระบบ โดยแสดงขอบเขตของระบบ และการเชื่อมโยงระหว่างระบบ เป็นเครื่องมือที่ช่วยในการทำความเข้าใจและวิเคราะห์ความสัมพันธ์ระหว่างระบบภายในและภายนอก และเป็นขั้นตอนแรกในการออกแบบระบบ



รูปที่ 3.9 แผนภาพ Context Diagram

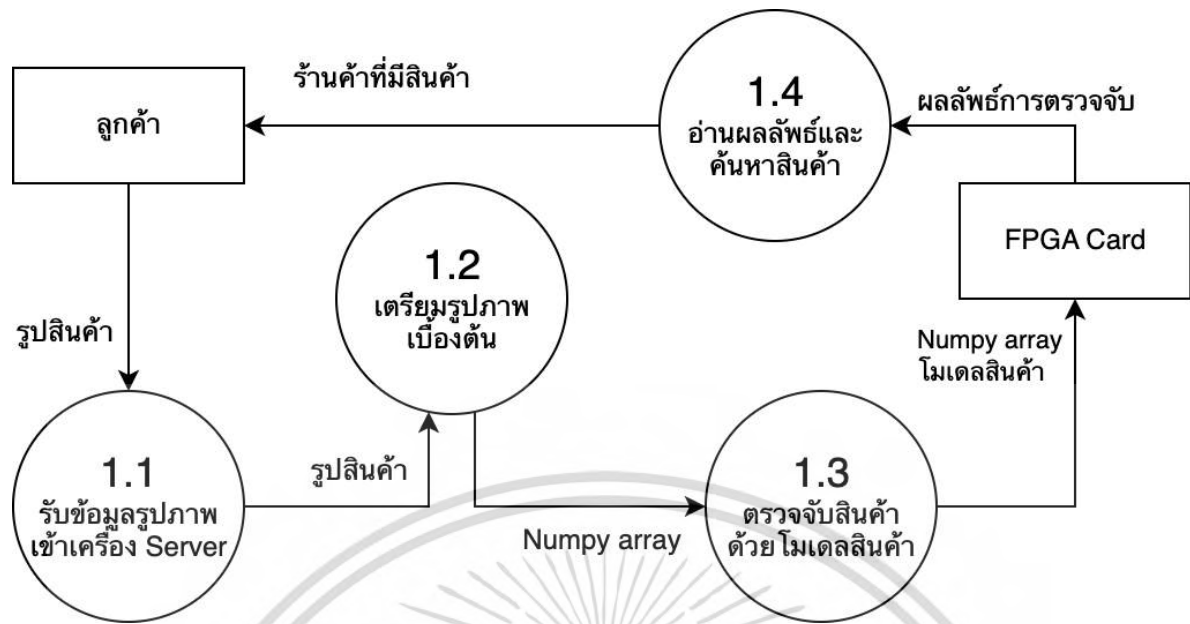
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.5. Data Description

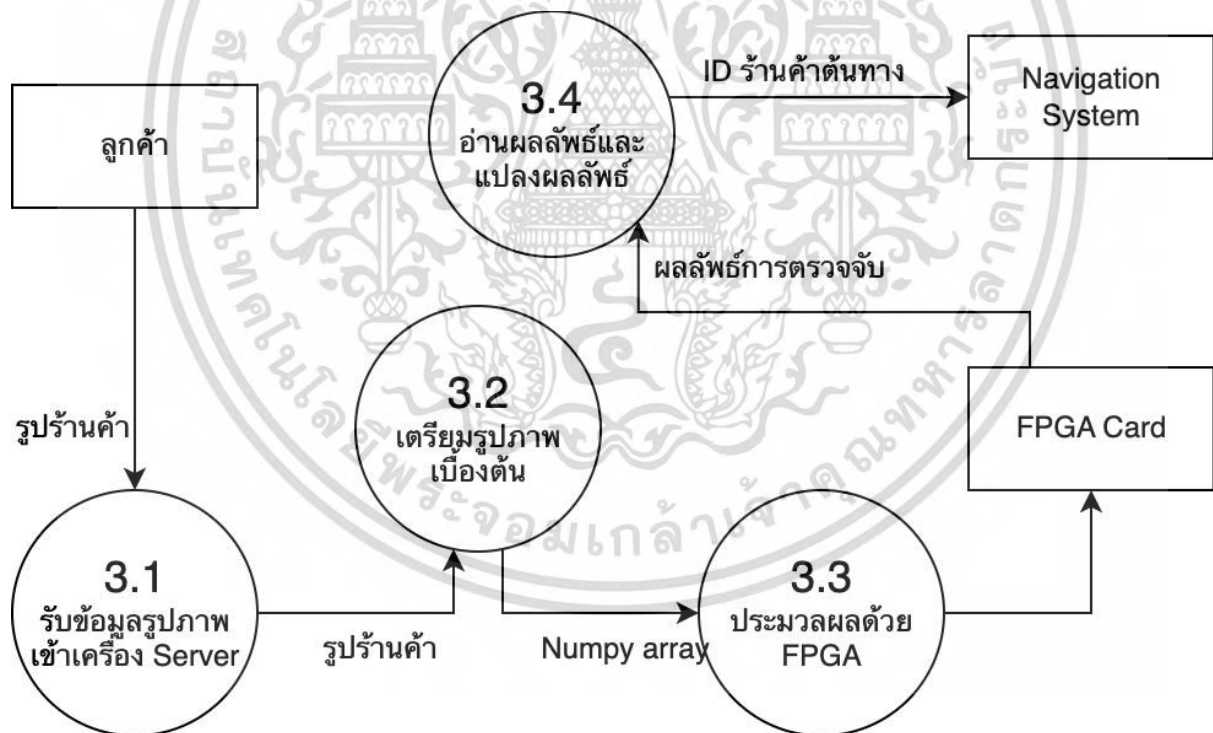


รูปที่ 3.10 แผนภาพ Data Flow Diagram Level 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

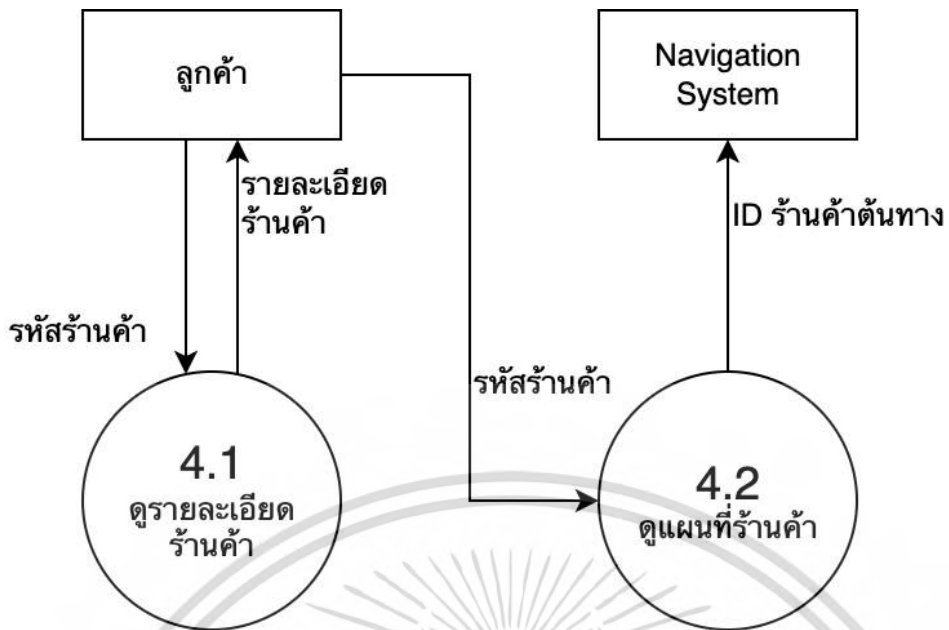


รูปที่ 3.11 แผนภาพ Data Flow Diagram Level 2-1

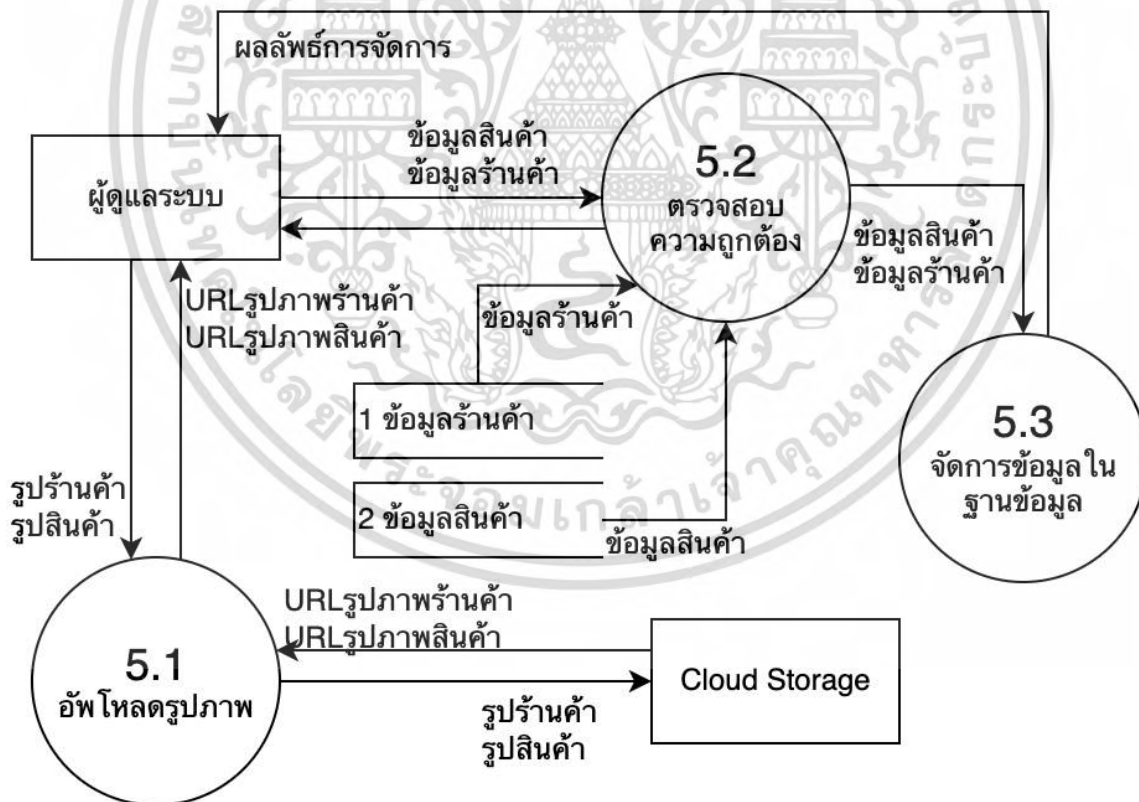


รูปที่ 3.12 แผนภาพ Data Flow Diagram Level 2-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แผนภาพ Data Flow Diagram Level 2-4



รูปที่ 3.14 แผนภาพ Data Flow Diagram Level 2-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### อัลกอริทึม 1.4 การอ่านผลลัพธ์และค้นหาสินค้า

---

##### อัลกอริทึม 1.4 การอ่านผลลัพธ์และค้นหาสินค้า

---

**ข้อมูลนำเข้า:** ผลลัพธ์จากการประมวลผลการตรวจจับ

**ผลลัพธ์:** รายการร้านค้าที่มีสินค้า

- 1: ถอดรหัสผลลัพธ์ด้วยอัลกอริทึม postprocess ของ yolo
  - 2: เก็บผลลัพธ์ที่ถอดรหัสในรูป {ตำแหน่งกรอบ , ชื่อคลาส, ค่าความมั่นใจ}
  - 3: เรียงผลลัพธ์ที่ถอดรหัสแล้วด้วยค่าความมั่นใจ จากมากไปน้อย
  - 4: สร้าง txt ไฟล์ผลลัพธ์จากชื่อรูปภาพ
  - 5: วนซ้ำ ผลลัพธ์ที่ถอดรหัสทุกตัว
  - 6: เขียนผลลัพธ์ลงใน txt ไฟล์รูปแบบ ชื่อคลาส,ค่าความมั่นใจ
  - 7: ขึ้นบรรทัดใหม่
  - 8: จบloop
  - 9: ลบรูปภาพที่ประมวลผลแล้ว
  - 10: วนloop ผลลัพธ์ในไฟล์ txt
  - 11: แยกชื่อคลาสกับค่าความมั่นใจด้วยคอมม่า
  - 12: ค้นหาชื่อคลาสในฐานข้อมูลสินค้า
  - 13: จบloop
  - 14: ลบไฟล์ txt ผลลัพธ์
  - 15: ส่งรายการร้านค้าที่มีสินค้าให้ลูกค้า
- 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### อัลกอริทึม 3.4 การอ่านผลลัพธ์และแปลงผลลัพธ์

---

#### อัลกอริทึม 3.4 การอ่านผลลัพธ์และแปลงผลลัพธ์

---

**ข้อมูลนำเข้า:** ผลลัพธ์จากการประมวลผลการตรวจจับ

**ผลลัพธ์:** ID ร้านค้า

- 1: ถอดรหัสผลลัพธ์ด้วยอัลกอริทึม postprocess ของ yolo
  - 2: เก็บผลลัพธ์ที่ถอดรหัสในรูป {ตำแหน่งกรอบ , ชื่อคลาส, ค่าความมั่นใจ}
  - 3: เรียงผลลัพธ์ที่ถอดรหัสแล้วด้วยค่าความมั่นใจ จากมากไปน้อย
  - 4: สร้าง txt ไฟล์ผลลัพธ์จากชื่อรูปภาพ
  - 5: วนซ้ำ ผลลัพธ์ที่ถอดรหัสทุกตัว
  - 6: เขียนผลลัพธ์ลงใน txt ไฟล์รูปแบบ ชื่อคลาส,ค่าความมั่นใจ
  - 7: ขึ้นบรรทัดใหม่
  - 8: จบloop
  - 9: ลบรูปภาพที่ประมวลผลแล้ว
  - 10: วนloop ผลลัพธ์ในไฟล์ txt
  - 11: แยกชื่อคลาสกับค่าความมั่นใจด้วยคอมม่า
  - 12: ค้นหาชื่อคลาสในฐานข้อมูลร้านค้า
  - 13: จบloop
  - 14: ลบไฟล์ txt ผลลัพธ์
  - 15: ส่ง ID ร้านค้าให้ *Navigation Path System*
- 

### อัลกอริทึม 5.2 การตรวจสอบความถูกต้อง

---

#### อัลกอริทึม 5.2 การตรวจสอบความถูกต้อง

---

**ข้อมูลนำเข้า:** ข้อมูลสินค้าหรือข้อมูลร้านค้า

**ผลลัพธ์:** ผลลัพธ์การจัดการข้อมูล

- 1: ตรวจสอบข้อมูลนำเข้า ข้อมูลในช่องที่จำเป็นต้องไม่ว่าง และเป็นประเภทที่ถูกต้อง
  - 2: ถ้า ข้อมูลสินค้าหรือข้อมูลร้านค้าถูกต้องครบถ้วน
  - 3: เพิ่มดำเนินการจัดการข้อมูลในฐานข้อมูล
  - 4: ถ้าไม่ถูกต้อง
  - 5: ปฏิเสธการจัดการข้อมูล
  - 6: ส่งกลับผลลัพธ์การจัดการ
- 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

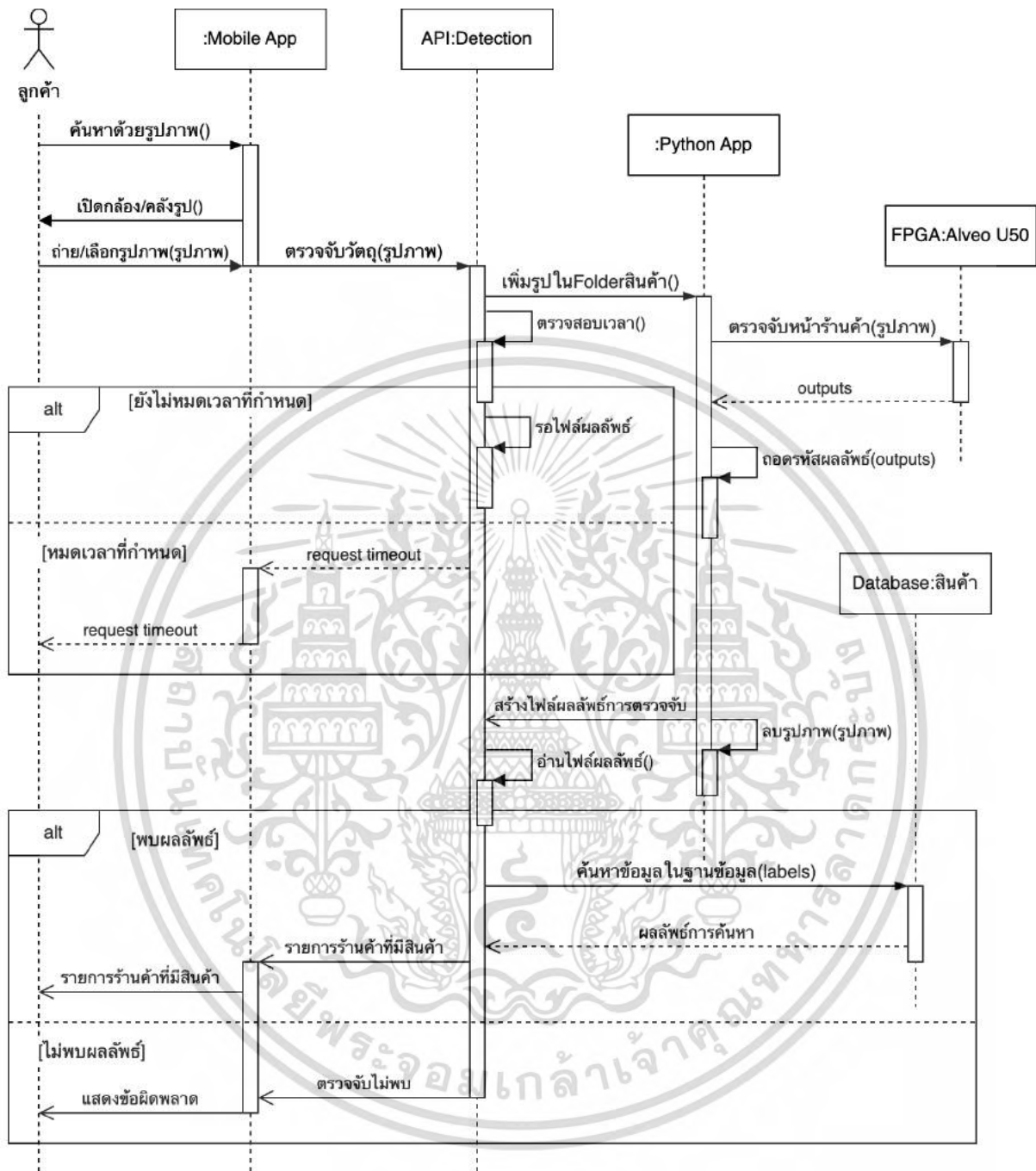
### 3.4.6. Sequence Diagram

เป็นแผนภาพที่ใช้แสดงการเชื่อมโยงระหว่างออบเจกต์ (object) และการเรียกใช้ของฟังก์ชัน (method) ในโปรแกรม โดยใช้เส้นเชื่อมโยงและเครื่องหมายบรรทัดเพื่อแสดงโครงสร้างการเรียกใช้ของฟังก์ชันและการเรียกใช้ระหว่างออบเจกต์ เป็นเครื่องมือที่ช่วยในการทำความเข้าใจและวิเคราะห์การเรียกใช้ของฟังก์ชันและการเชื่อมโยงระหว่างออบเจกต์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.6.1. Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยรูปภาพ



รูปที่ 3.15 Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยรูปภาพ

Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยรูปภาพ เป็นแผนภาพแสดงขั้นตอนการปฏิบัติงานของ ลูกค้า ดำเนินการค้นหาสินค้าด้วยรูปภาพ โดยสามารถอธิบายได้เป็นขั้นตอนได้ดังนี้

1. ลูกค้าเข้าเมนูการค้นหาด้วยรูปภาพ

2. แอปพลิเคชันมือถือทำการเปิดกล้องหรือคลังรูปภาพเพื่อเลือกรูปภาพที่ต้องการค้นหา โดยขั้นตอนการคัดเลือก

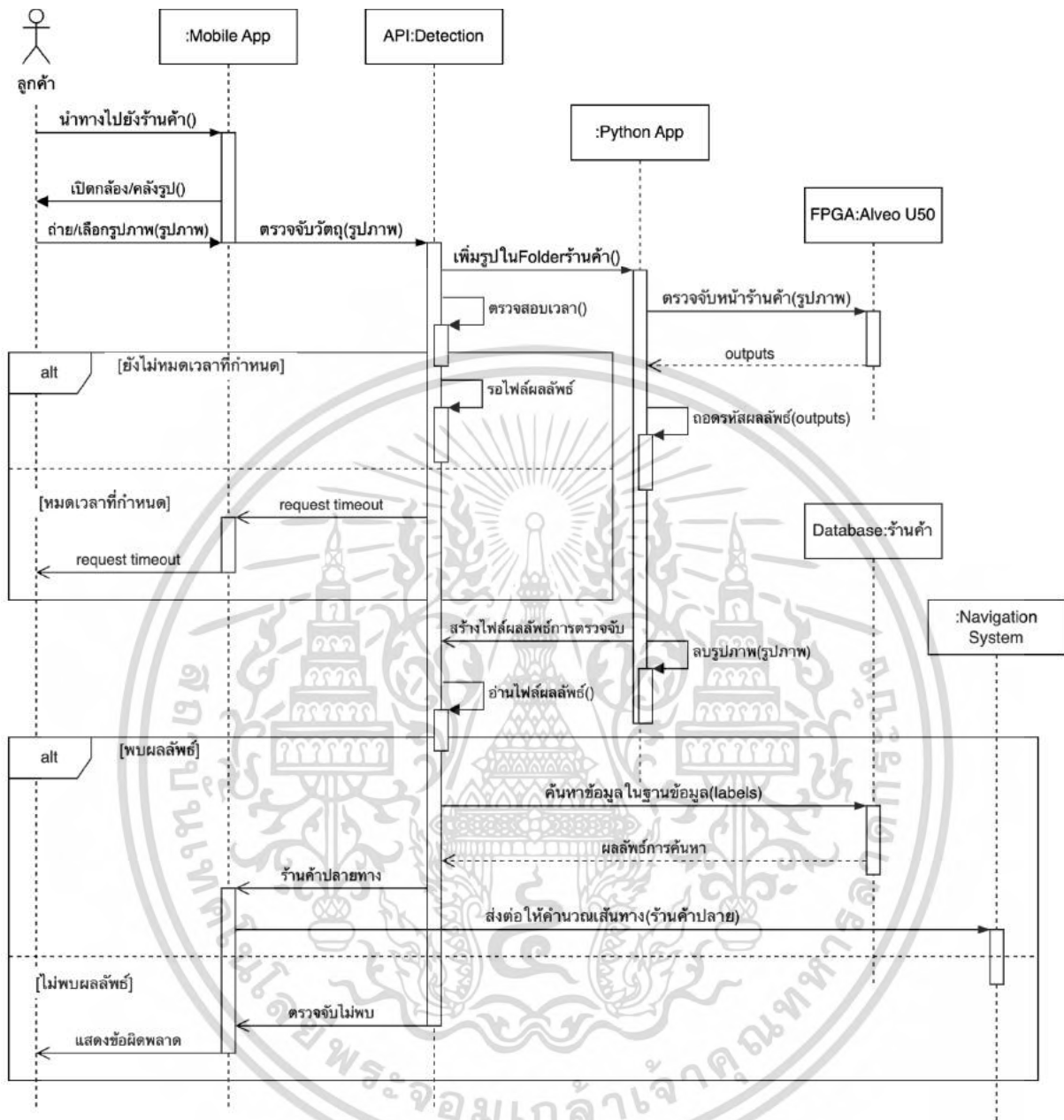
3. ลูกค้าถ่ายรูปหรือเลือกรูปภาพที่ต้องการค้นหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แอปพลิเคชันมือถือส่งคำขอการตรวจจับไปให้ API Endpoint: Detect
5. API Endpoint: Detect ทำการส่งรูปภาพเข้าไปใน Folder ที่เป็นแหล่งเก็บรูปภาพสินค้าที่ยังไม่ได้ตรวจจับ
6. แอปพลิเคชัน Python ทำการดึงรูปภาพที่อยู่ใน Folder สินค้าไปตรวจจับ
7. API Endpoint: Detect ตรวจสอบเวลาที่กำหนดเพื่อรอไฟล์ผลลัพธ์การตรวจจับ
8. ถ้าหมดเวลาตามที่กำหนดจะถือว่าเป็นข้อผิดพลาดของระบบ และแจ้งข้อผิดพลาดของระบบให้ ลูกค้าทราบ
9. แอปพลิเคชัน Python สร้างไฟล์ผลลัพธ์การตรวจจับ และลบรูปภาพที่ตรวจจับแล้ว
10. ถ้ามีตรวจจับพบผลลัพธ์จะดำเนินการค้นหาผลิตภัณฑ์นั้นในฐานข้อมูล
11. ส่งรายการสินค้ากลับให้ลูกค้า
12. ถ้าตรวจจับไม่พบจะดำเนินการแจ้งข้อผิดพลาดให้ลูกค้าทราบ และลองดำเนินการตรวจจับใหม่อีกครั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.6.2. Sequence Diagram แสดงรายละเอียดการแสดงผลเส้นทางไปยังร้านค้าปลายทาง



รูปที่ 3.16 Sequence Diagram แสดงรายละเอียดการแสดงผลเส้นทางไปยังร้านค้าปลายทาง

Sequence Diagram แสดงรายละเอียดการแสดงผลเส้นทางไปยังร้านค้าปลายทาง เป็นแผนภาพแสดงขั้นตอนการปฏิบัติงานของ ลูกค้า ดำเนินเรียกดูเส้นทางไปยังร้านค้าปลายทาง โดยสามารถอธิบายได้เป็นขั้นตอนได้ดังนี้

1. ลูกค้ากดปุ่มแสดงเส้นทางไปยังร้านค้า

2. แอปพลิเคชันมือถือทำการเปิดกล้องหรือคลังรูปภาพเพื่อเลือกรูปภาพหน้าร้านค้าปัจจุบัน

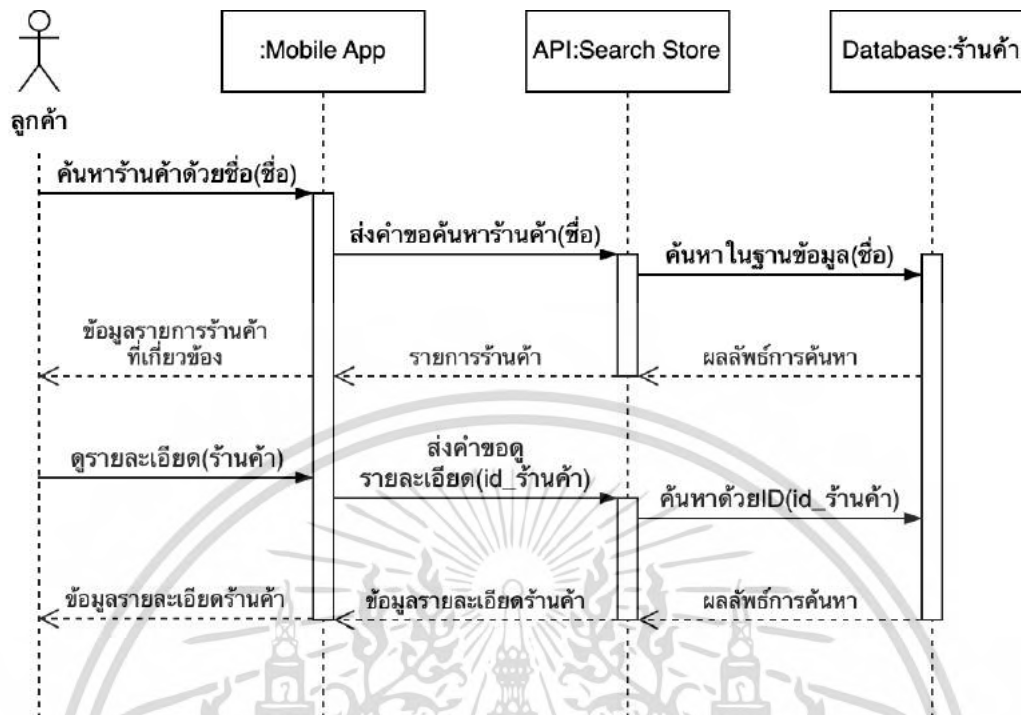
3. ลูกค้าถ่ายรูปหรือเลือกรูปภาพหน้าร้านค้าปัจจุบัน

หมายเหตุ: เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่การวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แอปพลิเคชันมือถือส่งคำขอการตรวจจับไปให้ API Endpoint: Detect
5. API Endpoint: Detect ทำการส่งรูปภาพเข้าไปใน Folder ที่เป็นแหล่งเก็บรูปภาพหน้าร้านค้าที่ยังไม่ได้ตรวจจับ
6. แอปพลิเคชัน Python ทำการดึงรูปภาพที่อยู่ใน Folder หน้าร้านค้าไปตรวจจับ
7. API Endpoint: Detect ตรวจสอบเวลาที่กำหนดเพื่อรอไฟล์ผลลัพธ์การตรวจจับ
8. ถ้าหมดเวลาตามที่กำหนดจะถือว่าเป็นข้อผิดพลาดของระบบ และแจ้งข้อผิดพลาดของระบบให้ ลูกค้าทราบ
9. แอปพลิเคชัน Python สร้างไฟล์ผลลัพธ์การตรวจจับ และลบรูปภาพที่ตรวจจับแล้ว
10. ถ้ามีตรวจจับพบผลลัพธ์จะดำเนินการค้นหาผลลัพธ์นั้นในฐานข้อมูล
11. ส่ง ID ร้านค้าปลายทางให้แอปพลิเคชันมือถือ
12. แอปพลิเคชันมือถือส่งตำแหน่งต้นทางและปลายทางให้ระบบ Navigation คำนวณและดำเนินการต่อไป
13. ถ้าตรวจจับไม่พบจะดำเนินการแจ้งข้อผิดพลาดให้ลูกค้าทราบ และลองดำเนินการตรวจจับใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.6.3. Sequence Diagram แสดงรายละเอียดการค้นหาร้านค้าด้วยชื่อ



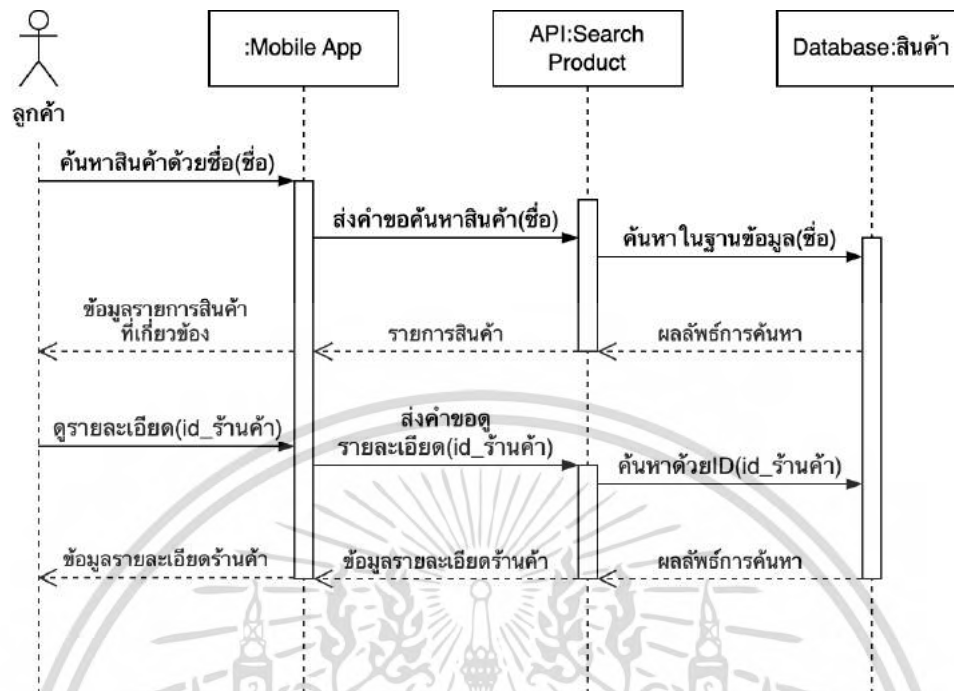
รูปที่ 3.17 Sequence Diagram แสดงรายละเอียดการค้นหาร้านค้าด้วยชื่อ

Sequence Diagram แสดงรายละเอียดการค้นหาร้านค้าด้วยชื่อ เป็นแผนภาพแสดงขั้นตอนการปฏิบัติงานของ ลูกค้า ดำเนินการค้นหาสินค้าด้วยชื่อ โดยสามารถอธิบายได้เป็นขั้นตอนได้ดังนี้

1. ลูกค้าเข้าเมนูค้นหาร้านค้าด้วยชื่อ โดยส่งชื่อร้านไปให้กับแอปพลิเคชันมือถือ
2. แอปพลิเคชันมือถือส่งคำค้นไปให้ API Endpoint: Search Store
3. API Endpoint: Search Store ค้นหาชื่อร้านในฐานข้อมูลร้านค้าที่มีความเกี่ยวข้องกับคำค้นหาที่รับมา
4. ส่งผลลัพธ์การค้นหากลับให้แอปพลิเคชันมือถือ
5. ลูกค้าได้รับรายการร้านค้าที่เกี่ยวข้องกับคำที่ค้นหา
6. ลูกค้ากดเลือกดูรายละเอียดร้านค้าที่ต้องการ
7. แอปพลิเคชันมือถือส่งคำขอดูรายละเอียดร้านค้า โดยอ้างอิงด้วย ID ไปให้ API Endpoint: Search Store
8. API Endpoint: Search Store ค้นหารายละเอียดร้านค้าโดยอ้างอิงด้วย ID
9. ส่งผลลัพธ์การค้นหากลับให้แอปพลิเคชันมือถือ
10. ลูกค้าได้รับรายละเอียดข้อมูลร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.6.4. Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยชื่อ



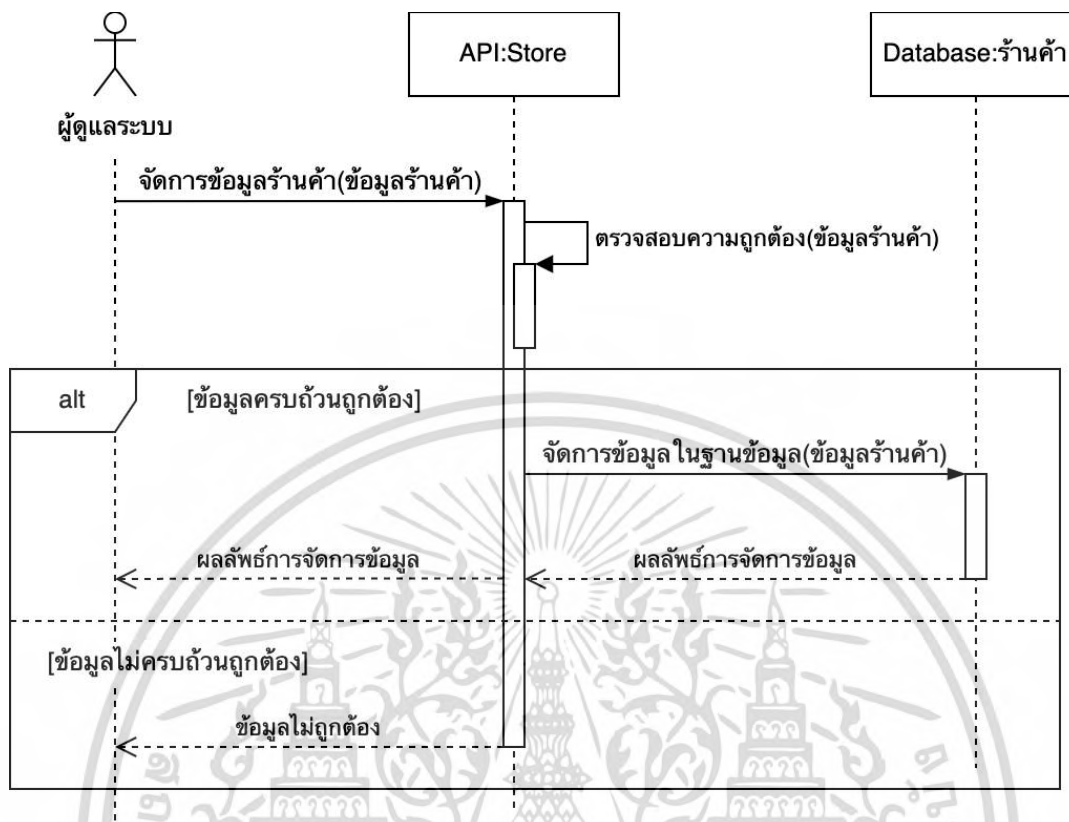
รูปที่ 3.18 Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยชื่อ

Sequence Diagram แสดงรายละเอียดการค้นหาสินค้าด้วยชื่อ เป็นแผนภาพแสดงขั้นตอนการปฏิบัติงานของ ลูกค้า ดำเนินการค้นหาสินค้าด้วยชื่อ โดยสามารถอธิบายได้เป็นขั้นตอนได้ดังนี้

1. ลูกค้าเข้าเมนูค้นหาสินค้าด้วยชื่อ โดยส่งชื่อร้านไปให้กับแอปพลิเคชันมือถือ
2. แอปพลิเคชันมือถือส่งคำค้นไปให้ API Endpoint: Search Product
3. API Endpoint: Search Store ค้นหาชื่อร้านในฐานข้อมูลสินค้าที่มีความเกี่ยวข้องกับคำค้นหาที่ได้รับมา
4. ส่งผลลัพธ์การค้นหากลับให้แอปพลิเคชันมือถือ
5. ลูกค้าได้รับรายการสินค้าที่เกี่ยวข้องกับคำที่ค้นหา
6. ลูกค้ากดเลือกดูรายละเอียดร้านค้าที่ต้องการ
7. แอปพลิเคชันมือถือส่งคำขอรูยละเอียดร้านค้า โดยอ้างอิงด้วย ID ไปให้ API Endpoint: Search Store
8. API Endpoint: Search Store ค้นหารายละเอียดร้านค้าโดยอ้างอิงด้วย ID
9. ส่งผลลัพธ์การค้นหากลับให้แอปพลิเคชันมือถือ
10. ลูกค้าได้รับรายละเอียดข้อมูลร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.4.6.5. Sequence Diagram แสดงรายละเอียดจัดการข้อมูลร้านค้า



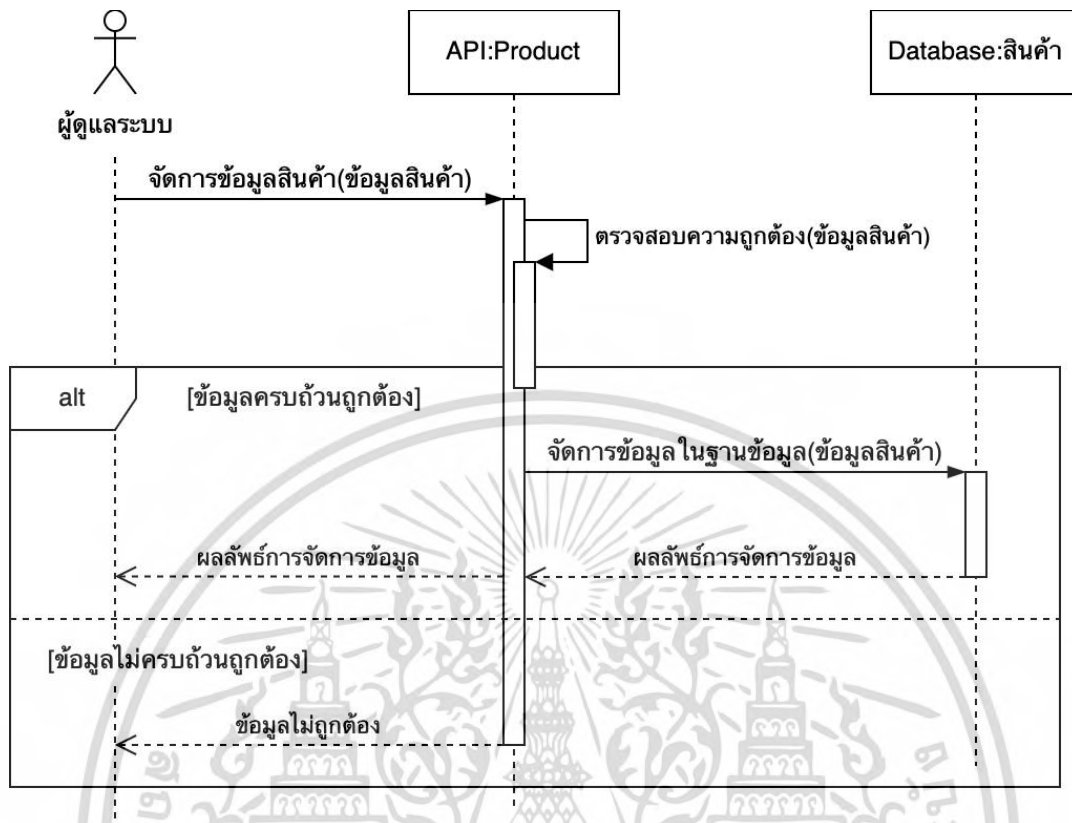
รูปที่ 3.19 Sequence Diagram แสดงรายละเอียดจัดการข้อมูลร้านค้า

Sequence Diagram แสดงรายละเอียดการจัดการข้อมูลร้านค้า เป็นแผนภาพแสดงขั้นตอนการปฏิบัติงานของ ผู้ดูแลระบบ ดำเนินการจัดการข้อมูลร้านค้า โดยสามารถอธิบายได้เป็นขั้นตอนได้ดังนี้

1. ผู้ดูแลส่งข้อมูลร้านค้าที่ต้องการจัดการให้ API Endpoint: Store
2. API Endpoint: Store ตรวจสอบความถูกต้องและความครบถ้วนของข้อมูลที่เป็นต่อการจัดการ
3. ถ้าหากข้อมูลมีความถูกต้องและครบถ้วน API Endpoint: Store จะดำเนินการจัดการกับข้อมูลในฐานข้อมูลร้านค้า
4. ตอบกลับผลลัพธ์การดำเนินการจัดการข้อมูลให้ผู้ดูแลระบบทราบ
5. ถ้าข้อมูลไม่ถูกต้องหรือไม่ครบถ้วน API Endpoint: Store จะตอบกลับข้อผิดพลาดให้ผู้ดูแลระบบทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.6.6. Sequence Diagram แสดงรายละเอียดจัดการข้อมูลสินค้า



รูปที่ 3.20 Sequence Diagram แสดงรายละเอียดจัดการข้อมูลสินค้า

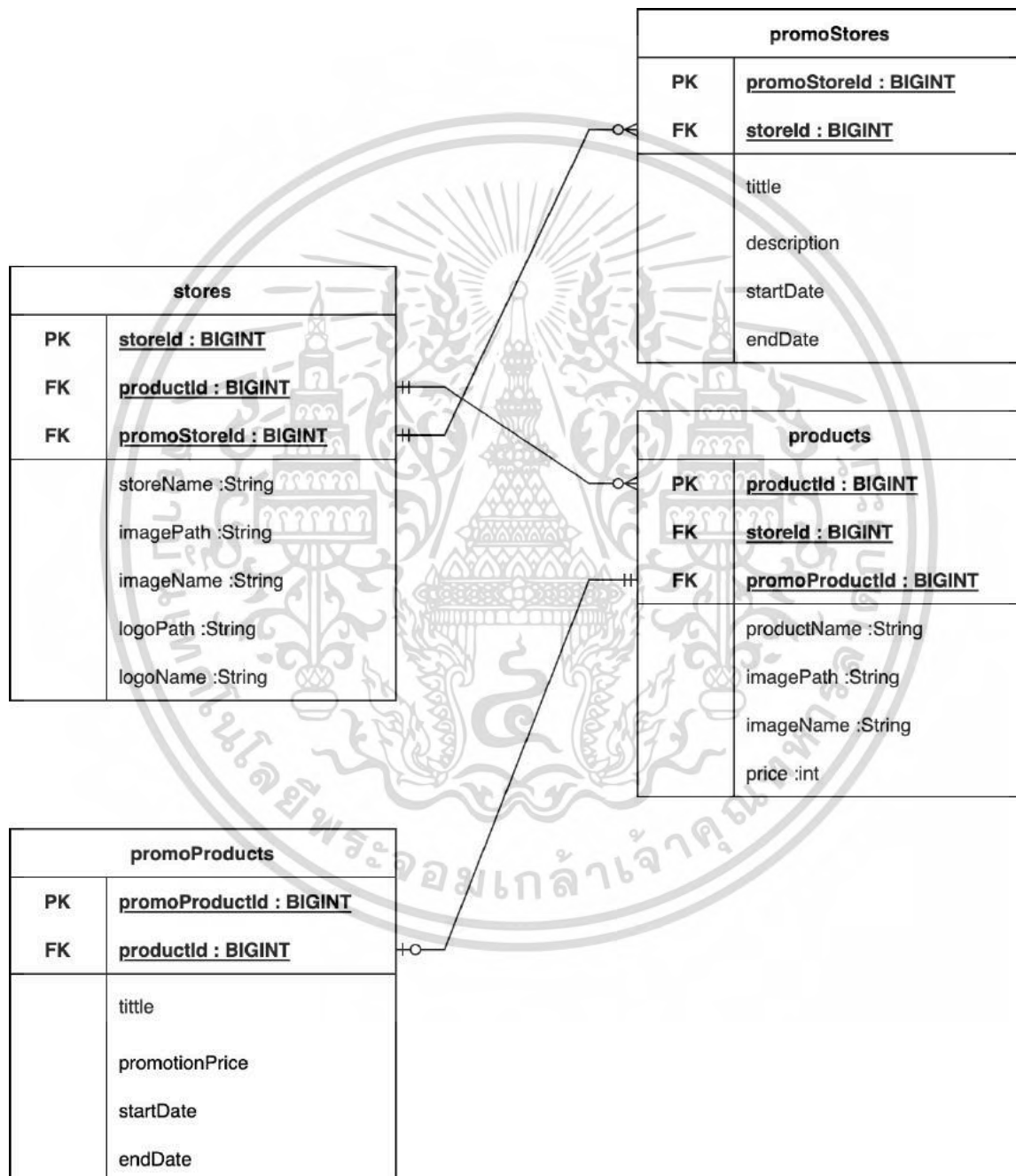
Sequence Diagram แสดงรายละเอียดการจัดการข้อมูลสินค้า เป็นแผนภาพแสดงขั้นตอนการปฏิบัติงานของ ผู้ดูแลระบบ ดำเนินการจัดการข้อมูลสินค้า โดยสามารถอธิบายได้เป็นขั้นตอนได้ดังนี้

1. ผู้ดูแลส่งข้อมูลสินค้าที่ต้องการจัดการให้ API Endpoint: Product
2. API Endpoint: Product ตรวจสอบความถูกต้องและความครบถ้วนของข้อมูลที่จำเป็นต่อการจัดการ
3. ถ้าหากข้อมูลมีความถูกต้องและครบถ้วน API Endpoint: Product จะดำเนินการจัดการกับข้อมูลในฐานข้อมูลสินค้า
4. ตอบกลับผลลัพธ์การดำเนินการจัดการข้อมูลให้ผู้ดูแลระบบทราบ
5. ถ้าข้อมูลไม่ถูกต้องหรือไม่ครบถ้วน API Endpoint: Product จะตอบกลับข้อผิดพลาดให้ผู้ดูแลระบบทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.7. Entity Relations Diagram

เป็นแบบจำลองการเชื่อมโยงระหว่างตารางของฐานข้อมูล ซึ่งใช้ในการแสดงความสัมพันธ์ระหว่างตารางในฐานข้อมูล พร้อมกับแสดงความสัมพันธ์ของข้อมูลในแต่ละตารางในรูปแบบของคุณสมบัติและความสัมพันธ์ ซึ่งจะช่วยให้การออกแบบฐานข้อมูลให้มีประสิทธิภาพและป้องกันปัญหาที่อาจเกิดขึ้นในภายหลัง



รูปที่ 3.21 Entity Relations Diagram แสดงโครงสร้างของฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Data Dictionary

ตารางบอกรายละเอียดของข้อมูลแต่ละตาราง และหน้าที่ของข้อมูลและคุณลักษณะในการเก็บข้อมูล

ตารางที่ 3.9 พจนานุกรมตาราง products

Field	Attribute Name	Data Type	Description	Key	Reference Table
1	productId	BIGINT	รหัสสินค้า	PK	
2	storeId	BIGINT	รหัสร้านค้า	FK	stores
3	promoProductId	BIGINT	รหัสโปรโมชั่นของสินค้า	FK	promoProducts
4	productName	VARCHAR(255)	ชื่อสินค้า		
5	imagePath	VARCHAR(255)	ที่อยู่ URL รูปภาพสินค้า		
6	imageName	VARCHAR(255)	ชื่อรูปภาพสินค้า		
7	price	INTEGER	ราคาสินค้า		

ตารางที่ 3.10 พจนานุกรมตาราง stores

Field	Attribute Name	Data Type	Description	Key	Reference Table
1	storeId	BIGINT	รหัสร้านค้า	PK	
2	productId	BIGINT	รหัสสินค้า	FK	products
3	promoStoresId	BIGINT	รหัสโปรโมชั่นของร้านค้า	FK	promoStores
4	productName	VARCHAR(255)	ชื่อร้านค้า		
5	imagePath	VARCHAR(255)	ที่อยู่ URL รูปภาพหน้าร้านค้า		
6	imageName	VARCHAR(255)	ชื่อรูปภาพร้านค้า		
7	logoPath	VARCHAR(255)	ที่อยู่ URL รูปภาพโลโก้ของร้านค้า		
8	logoName	VARCHAR(255)	ชื่อรูปภาพโลโก้ร้านค้า		

ตารางที่ 3.11 พจนานุกรมตาราง promoProducts

Field	Attribute Name	Data Type	Description	Key	Reference Table
1	promoProductId	BIGINT	รหัสโปรโมชั่นของสินค้า	PK	
2	productId	BIGINT	รหัสสินค้า	FK	products
3	tittle	VARCHAR(255)	ชื่อโปรโมชั่น		
4	promotionPrice	INTEGER	ราคาสินค้าที่ลดแล้ว		
5	startDate	DATETIME	วันและเวลาเริ่มต้นของโปรโมชั่น		
6	endDate	DATETIME	วันและเวลาสิ้นสุดของโปรโมชั่น		

ตารางที่ 3.12 พจนานุกรมตาราง promoStores

Field	Attribute Name	Data Type	Description	Key	Reference Table
1	promoStoresId	BIGINT	รหัสโปรโมชั่นของร้านค้า	PK	
2	storeId	BIGINT	รหัสร้านค้า	FK	stores
3	tittle	VARCHAR(255)	ชื่อโปรโมชั่น		
4	description	VARCHAR(255)	คำอธิบายรายละเอียดโปรโมชั่น		
5	startDate	DATETIME	วันและเวลาเริ่มต้นของโปรโมชั่น		
6	endDate	DATETIME	วันและเวลาสิ้นสุดของโปรโมชั่น		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5. การออกแบบจำลองหน้าจอระบบ

หลังจากวิเคราะห์ระบบ และสกัดความต้องการของระบบเรียบร้อยแล้วขั้นตอนต่อมาคือการออกแบบหน้าจอ และกระบวนการทำงานของระบบ โดยคณะผู้จัดทำสามารถออกแบบจำลองหน้าจอตามความต้องการของระบบออกมาได้ดังนี้

#### 3.5.1. Mobile Application Mockup

คณะผู้จัดทำได้ดำเนินการสร้างแบบจำลองหน้าจอ Mobile Application หรือ Mock up โดยออกแบบร่วมกันกับกลุ่ม Navigation System เพื่อให้ง่ายต่อการพัฒนา Mobile Application และความชัดเจนในการพัฒนาระบบ โดยได้แบ่งหน้าจอออกเป็น 2 ขั้นตอนกระบวนการทำงานหลักดังนี้

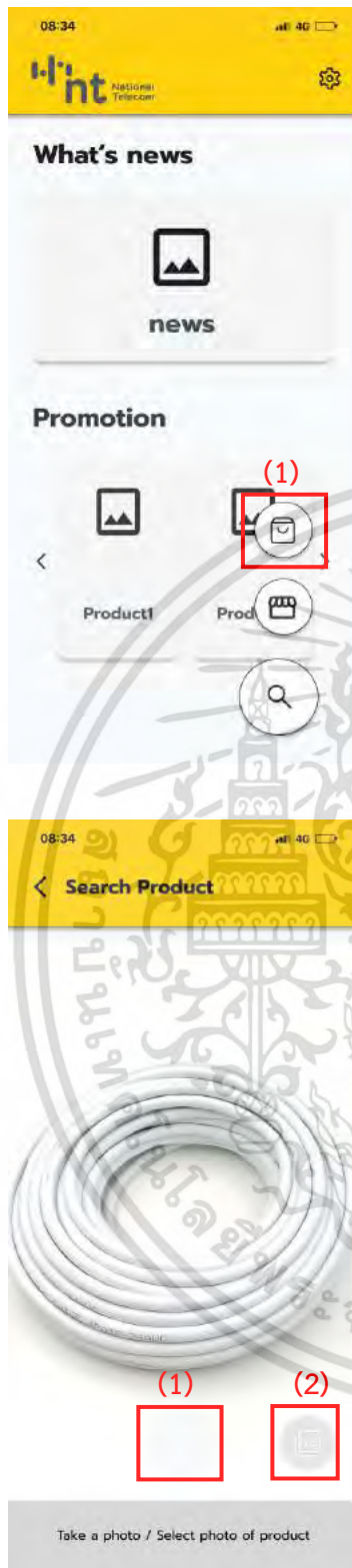
##### 3.5.1.1. Search Product with Image Flow



รูปที่ 3.22 Mockup: Search Product with Image Flow

Search with Image Flow เป็นขั้นตอนกระบวนการค้นหาสินค้าด้วยรูปภาพ โดยทุกหน้าที่มีลูกศรชี้ไปด้านซ้ายจะสามารถกดลูกศรย้อนกลับเพื่อกลับไปหน้าที่ผ่านมาได้ และสามารถอธิบายเป็นขั้นตอนรายละเอียดได้ดังนี้

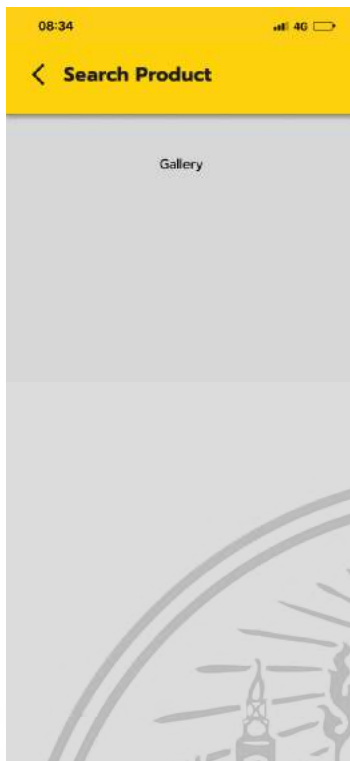
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



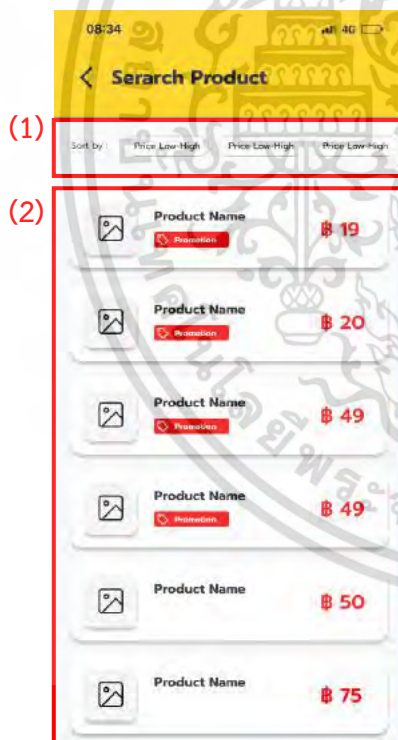
1. Home Screen เป็นหน้าแรกของแอปพลิเคชัน เมื่อกดปุ่มเมนู (1) Search Product with Image จะนำผู้ใช้งานไปหน้า 2. Product Search Screen เพื่อเปิดกล้องสำหรับถ่ายภาพสินค้า

2. Product Search Screen เป็นหน้าจอที่เปิดกล้องสำหรับถ่ายภาพสินค้า โดยกดปุ่ม (1) Caption เพื่อเป็นการถ่ายรูป และส่งภาพสินค้าไปประมวลผลในระบบเบื้องหลัง หรือสามารถเลือกรูปภาพสินค้าจากในคลังสินค้าได้โดยการกดที่ปุ่ม (2) คลังสินค้าเพื่อเปิดหน้า 3. Gallery Screen เมื่อส่งรูปภาพเข้าระบบเบื้องหลังแล้วจะไปหน้าต่อไป 4. List Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3. Gallery Screen สำหรับเลือกรูปภาพในคลังรูปภาพ เมื่อกดเลือกรูปภาพแล้ว แอปพลิเคชันจะส่งภาพสินค้าไปประมวลผลในระบบเบื้องหลัง



4. List Screen หน้าแสดงรายการสินค้า เมื่อระบบเบื้องหลังประมวลรูปภาพเสร็จสิ้นระบบจะส่งข้อมูลรายการสินค้ากลับมาให้ แอปพลิเคชันแสดงผลรายการสินค้า โดยสามารถเลือกการจัดเรียงรายการสินค้าได้ในช่องที่ (1) เรียงราคาจากน้อยไปมาก เรียงราคาจากมากไปน้อย และสามารถกดที่รายการสินค้า (2) เพื่อไปหน้า 5. Store Location Screen สำหรับดูแผนที่ร้านค้าด้วยระบบ Navigation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5. **Store Location Screen** หน้าแสดงแผนที่ร้านค้าในห้างสรรพสินค้า โดยเป็นหน้าจอการแสดงผลแผนที่ร้านค้าของระบบ Navigation สามารถกดปุ่ม (2) More Detail เพื่อดูรายละเอียดข้อมูลร้านค้านั้นๆได้ โดยจะเข้าสู่หน้าที่ 6. **Detail Screen** และสามารถกดปุ่ม (1) Enter Your Location เพื่อนำทางผู้ใช้งานจากจุดปัจจุบันไปยังร้านค้าที่กำลังดูอยู่ โดยจะเข้าสู่หน้า 7. **Front Store Screen**



6. **Detail Screen** หน้าแสดงรายละเอียดร้านค้า โดยจะแสดงรายละเอียดดังนี้ (1) โลโก้ร้านค้า (2) ชื่อร้านค้า ชั้นที่อยู่ (3) รูปภาพร้านค้า (4) โปรโมชั่นของร้านค้า (5) และรายการสินค้าทั้งหมดในร้านค้านั้นโดยสามารถเลือกการจัดเรียงรายการสินค้าได้ในช่องที่ (6) เรียงราคาจากน้อยไปมาก เรียงราคาจากมากไปน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



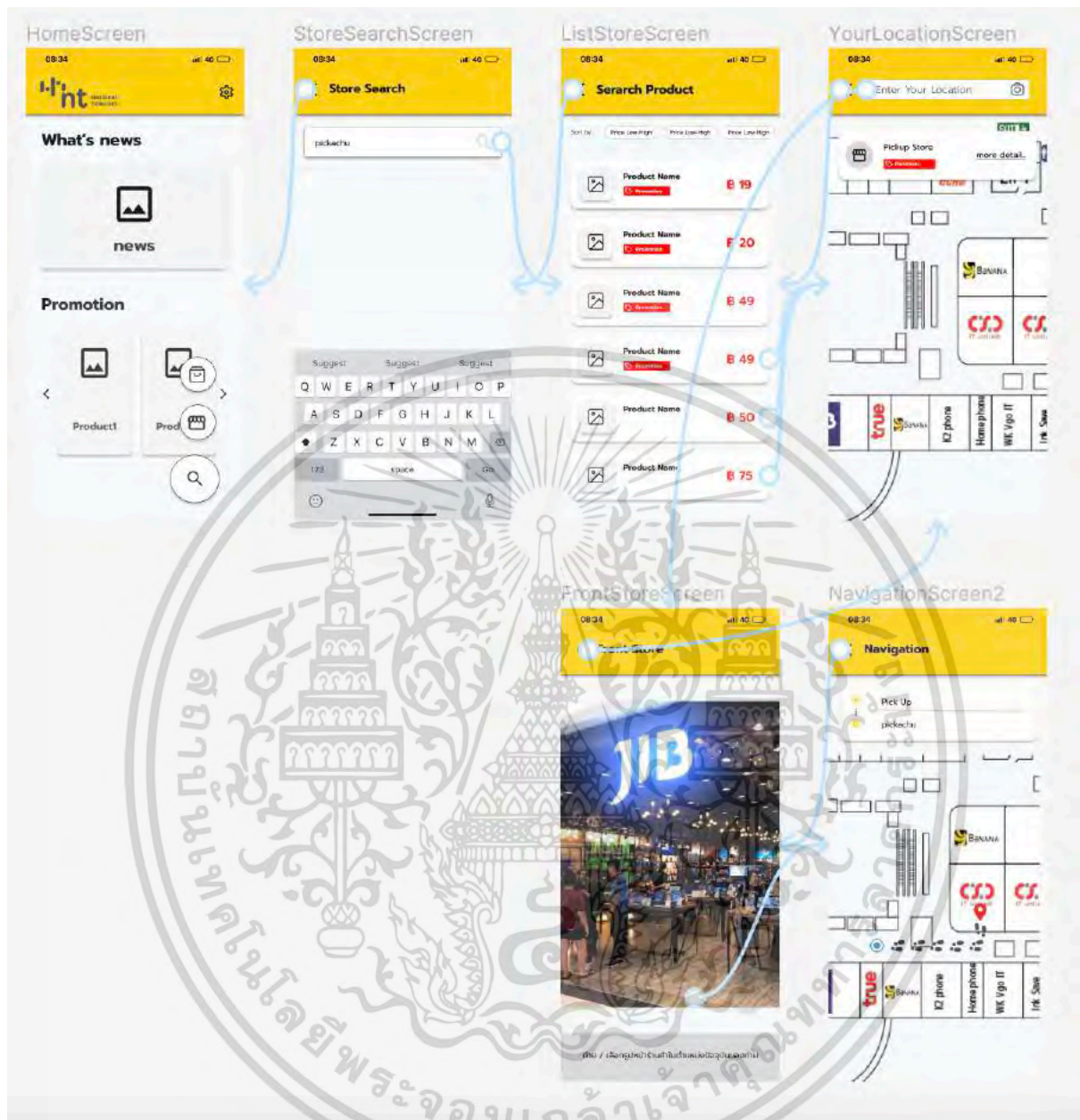
7. **Front Store Screen** หน้าจอสำหรับถ่ายรูปหน้าร้านค้า โดยกดปุ่ม (1) Caption เพื่อเป็นการถ่ายรูป และส่งภาพสินค้าไปประมวลผลในระบบเบื้องหลัง หรือสามารถเลือกรูปภาพหน้าร้านค้าจากในคลังสินค้าได้โดยการกดที่ปุ่ม (2) คลังสินค้าเพื่อเปิดหน้า 3. **Gallery Screen** เมื่อส่งรูปภาพเข้าระบบเบื้องหลังแล้วจะไปหน้าต่อไป 8. **Navigation Screen**



8. **Navigation Screen** หน้าจอแสดงการนำทางด้วยระบบ Navigation หลังจากผ่านการประมวลรูปภาพหน้าร้าน ระบบเบื้องหลังจะส่งกลับเป็นรหัสหน้าร้านปัจจุบันไปให้ระบบ Navigation ดำเนินการแสดงเส้นทางให้ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

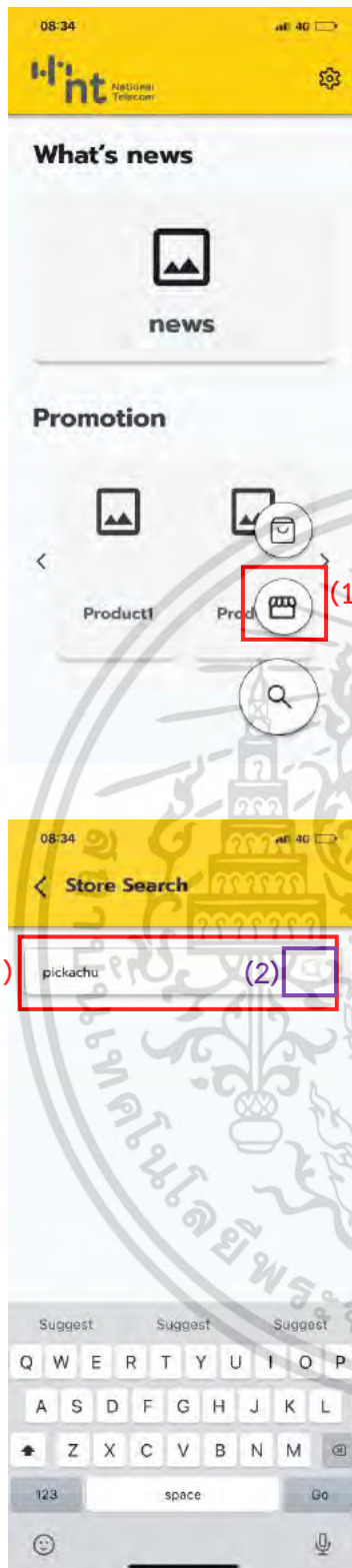
## 3.5.1.2. Search Store Flow



รูปที่ 3.23 Mockup: Search Store Flow

Search with Image Flow เป็นขั้นตอนกระบวนการค้นหาร้านค้าด้วยชื่อ โดยทุกหน้าที่มีลูกศรชี้ไปด้านซ้ายจะสามารถกดลูกศรย้อนกลับเพื่อกลับไปหน้าที่ผ่านมาได้ และสามารถอธิบายเป็นขั้นตอนรายละเอียดได้ดังนี้

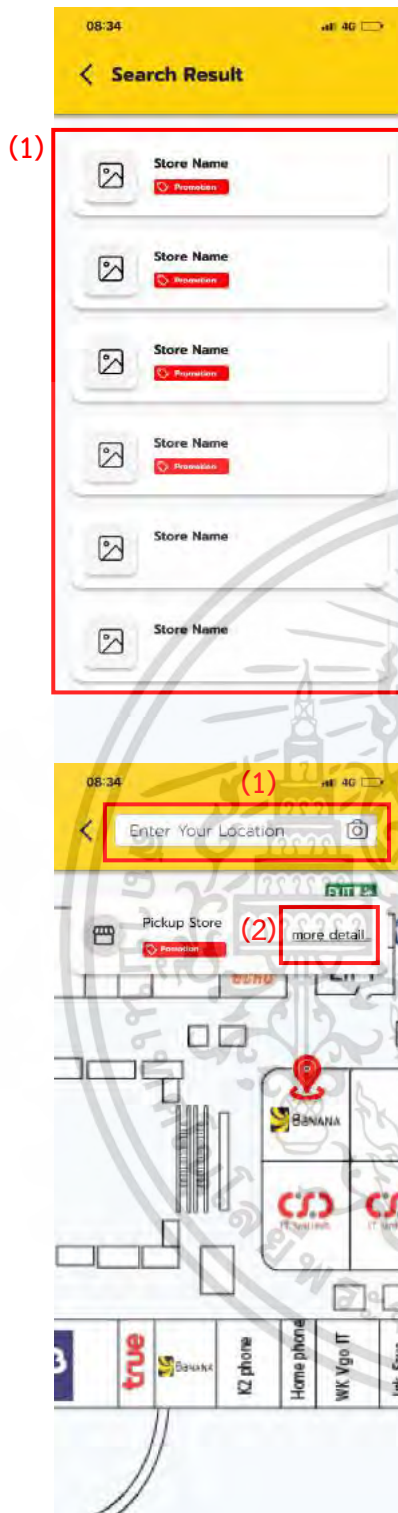
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. Home Screen เป็นหน้าแรกของแอปพลิเคชัน เมื่อกดปุ่มเมนู (1) Search store จะนำผู้ใช้งานไปหน้า 2. Store Search Screen เพื่อค้นหาร้านค้าด้วยชื่อ

2. Store Search Screen เป็นหน้าจอสำหรับค้นหาร้านค้าด้วยชื่อ โดยใส่คำค้นหาในช่อง (1) และกดค้นหา (2) เพื่อส่งคำค้นหาไปค้นหาในระบบเบื้องหลัง เมื่อการค้นหาเสร็จสิ้นระบบเบื้องหลังจะตอบกลับมาเป็นรายการร้านค้าที่ค้นพบและจะไปหน้าต่อไป 4. List Screen

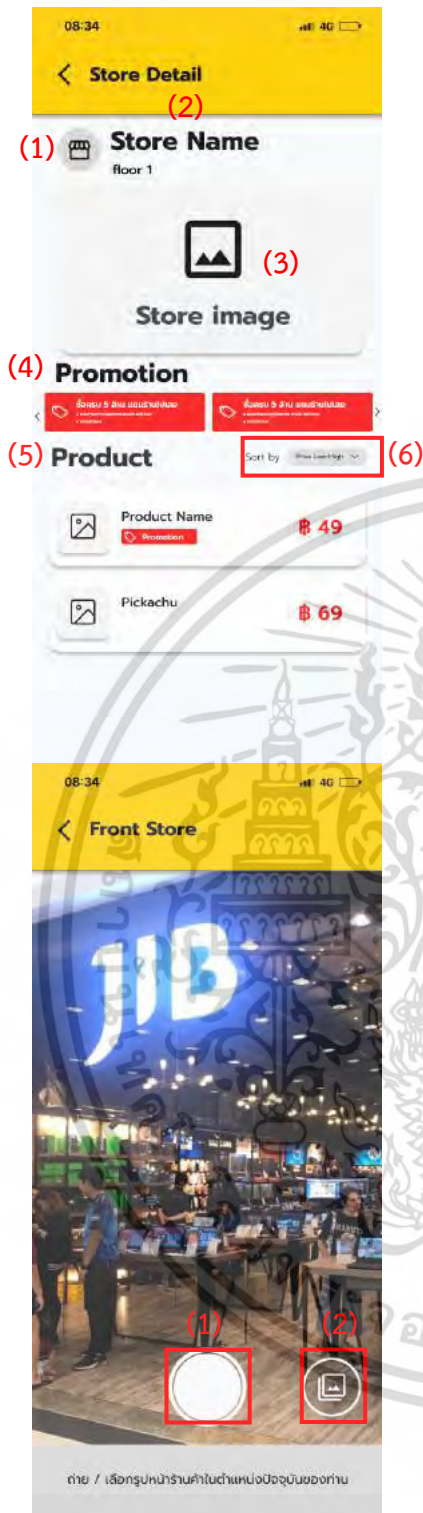
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



3. List Screen หน้าแสดงรายการร้านค้า เมื่อระบบเบื้องหลังค้นหาเสร็จสิ้นระบบจะส่งข้อมูลรายการร้านค้ากลับมาให้แอปพลิเคชันแสดงผลรายการร้านค้า โดยสามารถกดที่รายการร้านค้า (1) เพื่อไปหน้า 4. Store Location Screen สำหรับดูแผนที่ร้านค้าด้วยระบบ Navigation

4. Store Location Screen หน้าแสดงแผนที่ร้านค้าในห้างสรรพสินค้า โดยเป็นหน้าจอการแสดงผลแผนที่ร้านค้าของระบบ Navigation สามารถกดปุ่ม (2) More Detail เพื่อดูรายละเอียดข้อมูลร้านค้านั้นๆได้ โดยจะเข้าสู่หน้าที่ 5. Detail Screen และสามารถกดปุ่ม (1) Enter Your Location เพื่อนำทางผู้ใช้งานจากจุดปัจจุบันไปยังร้านค้าที่กำลังดูอยู่ โดยจะเข้าสู่หน้า 6. Front Store Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5. Detail Screen หน้าแสดงรายละเอียดร้านค้า โดยจะแสดงรายละเอียดดังนี้ (1) โลโก้ร้านค้า (2) ชื่อร้านค้า ชั้นที่อยู่ (3) รูปภาพร้านค้า (4) โปรโมชั่นของร้านค้า (5) และรายการสินค้า ทั้งหมดในร้านค้านั้นโดยสามารถเลือกการจัดเรียงรายการสินค้าได้ในช่องที่ (6) เรียงราคาจากน้อยไปมาก เรียงราคาจากมากไปน้อย

6. Front Store Screen หน้าจอสำหรับถ่ายรูปหน้าร้านค้า โดยกดปุ่ม (1) Caption เพื่อเป็นการถ่ายรูป และส่งภาพสินค้าไปประมวลผลในระบบเบื้องหลัง หรือสามารถเลือกรูปภาพหน้าร้านค้าจากในคลังสินค้าได้โดยการกดที่ปุ่ม (2) คลังสินค้าเพื่อเปิดหน้า Gallery Screen เมื่อส่งรูปภาพเข้าระบบเบื้องหลังแล้วจะไปหน้าต่อไป 7. Navigation Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

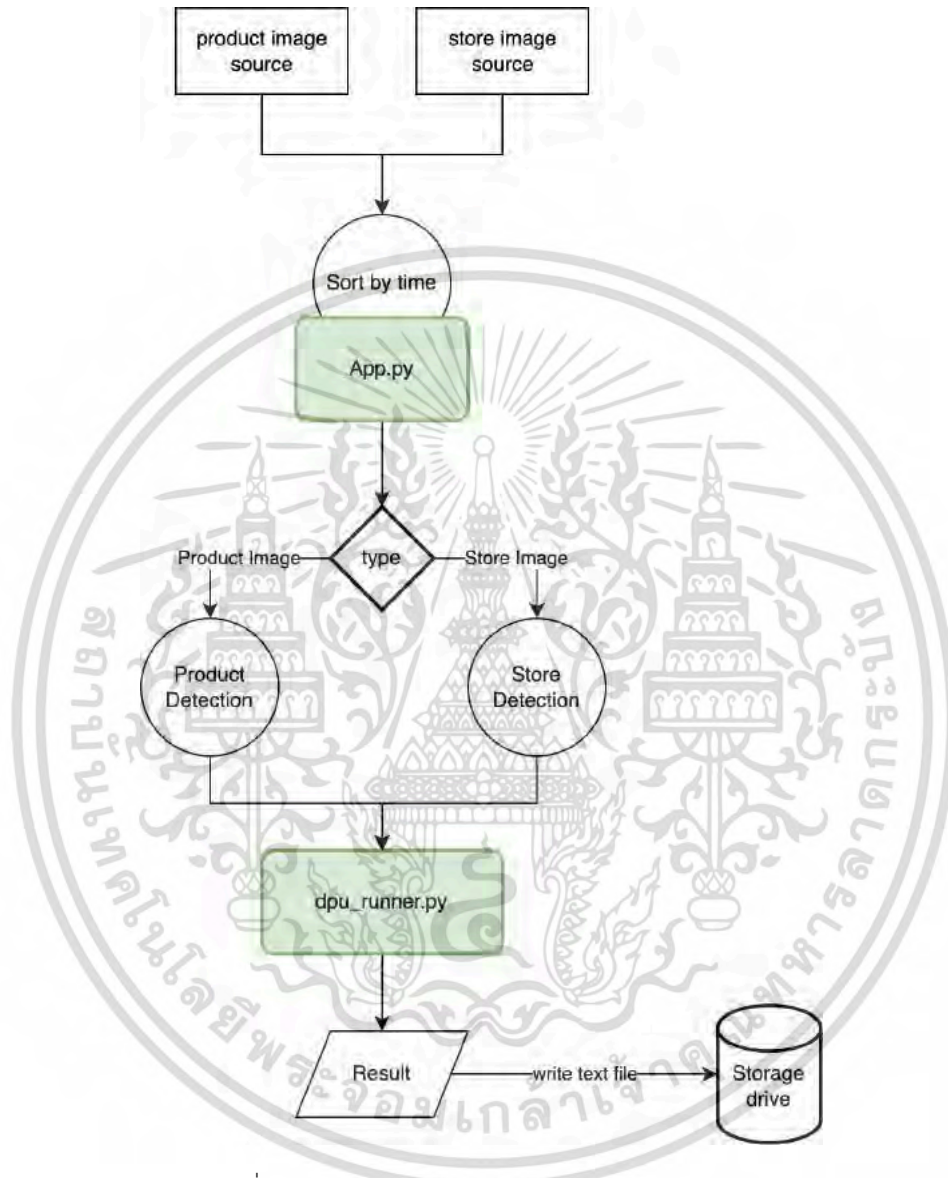


7. Navigation Screen หน้าจอแสดงการนำทาง  
ด้วยระบบ Navigation หลังจากผ่านการ  
ประมวลรูปภาพหน้าร้าน ระบบเบื้องหลังจะ  
ส่งกลับเป็นรหัสหน้าร้านปัจจุบันไปให้ระบบ  
Navigation ดำเนินการแสดงเส้นทางให้ผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5.2. Flow Chart Python Application

หลังจากออกแบบ Mockup หน้าจอแล้วขั้นตอนต่อไปคือการออกแบบ Flow Chart สำหรับการทำงานของ Python Application ที่จะเป็นส่วนประสานงานในการเรียกใช้งานการ์ด FPGA



รูปที่ 3.24 Process Flow Python Application

Python Application จะทำหน้าที่ในการตรวจสอบและนำรูปภาพจากแหล่ง Folder ที่กำหนด เข้าไปประมวลผลการตรวจจับวัตถุ โดยจะมีการเรียงลำดับรูปภาพตามเวลาที่รูปถูกส่งเข้ามาใน Folder นั้นๆ และมีการแยกประเภทการตรวจจับตามประเภทที่ต้องการ เมื่อประมวลผลรูปภาพเรียบร้อยแล้วจะส่งผลลัพธ์ไปเก็บไว้ใน Folder ผลลัพธ์ที่ Storage ภายในของเครื่อง Server ที่เป็น เพื่อให้ API Backend ส่งผลลัพธ์การตรวจจับให้ Client ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6. การฝึกสอนโมเดลปัญญาประดิษฐ์และปรับแต่งเป็น Fix-point โมเดล

ในการพัฒนาระบบต้นแบบ Visual Search มีการฝึกสอนโมเดลปัญญาประดิษฐ์ประเภท CNN ซึ่งเป็นโครงสร้างโมเดลสำหรับการทำ Object Detection หรือการตรวจจับวัตถุ และได้อ้างอิงตามทฤษฎีการฝึกสอนจากหัวข้อ 2.7.5.การพัฒนาและใช้งาน YOLOv4 บนการ์ด FPGA โดยคณะผู้จัดทำได้แบ่งออกเป็น 2 โมเดลด้วยกัน โดยแบ่งตามประเภทการตรวจจับเป็น โมเดลสำหรับตรวจจับหน้าร้านค้า และโมเดลสำหรับตรวจจับสินค้า โดยโมเดลร้านจะมี Class วัตถุที่ต้องการตรวจจับทั้งหมด 30 Class และโมเดลสินค้ามี 59 Class ซึ่งขั้นตอนในการทำฝึกสอนโมเดลและปรับจะแบ่งออกเป็น 3 หัวข้อใหญ่ ๆ ดังนี้

#### 3.6.1. ขั้นตอนการเตรียมข้อมูลสำหรับการฝึกสอน

เนื่องจากการฝึกสอนโมเดล AI สำหรับการทำ Object Detection ต้องใช้วิธีการฝึกสอนแบบ Supervised Learning หรือการเรียนรู้แบบมีคณสอน ในขั้นตอนการเตรียมข้อมูล จึงต้องแบ่งการเตรียมข้อมูลออกเป็น 2 ส่วนส่วนที่ 1 การเก็บรวบรวมข้อมูล และส่วนที่ 2 การ Labeling ข้อมูล หรือการอธิบายข้อมูลให้โมเดลเรียนรู้

##### 3.6.1.1. การเก็บรวบรวมข้อมูล

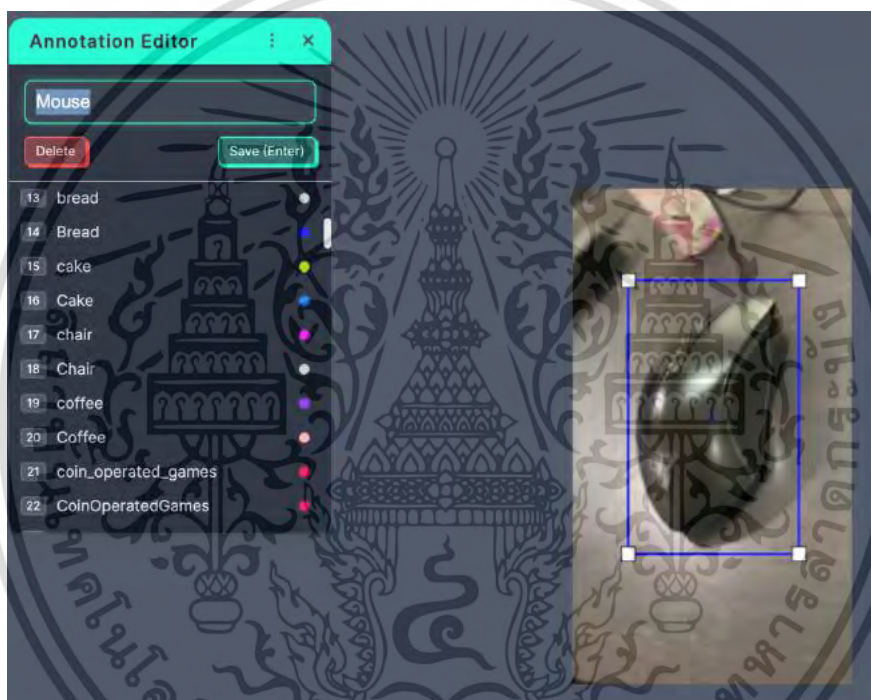
ข้อมูลสำหรับการฝึกสอนสำหรับทั้ง 2 โมเดลร้านค้าและสินค้า จะต้องใช้ข้อมูลที่เป็นรูปภาพที่มีวัตถุที่เราต้องการตรวจจับ คณะผู้จัดทำได้ทำการการบันทึกวิดีโอของวัตถุที่ต้องการจะตรวจจับในหลาย ๆ มุมมอง หลายสภาพแวดล้อม และนำวิดีโอที่ได้มา Snapshot แต่ละเฟรมของวิดีโอเพื่อให้ได้รูปภาพวัตถุในหลาย ๆ มุม และค้นหาภาพวัตถุเพิ่มเติมจาก Internet



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิใช้รูปที่ 3.25 ตัวอย่างข้อมูลหน้าร้านค้าของเอกสารทุกครั้งที่มีการนำไปใช้

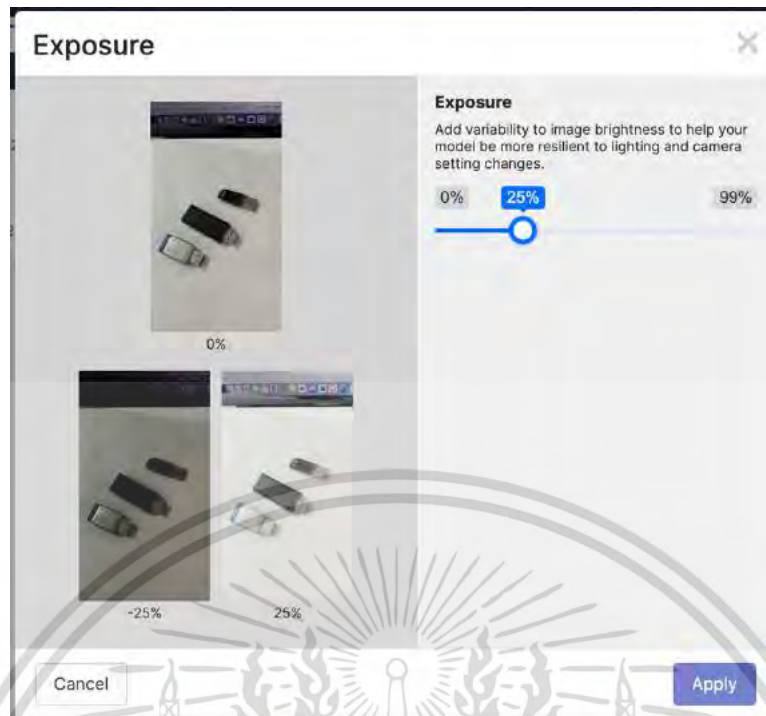
### 3.6.1.2. การ Labeling ข้อมูล

การ Labeling ข้อมูล หรือการอธิบายข้อมูลให้โมเดลเรียนรู้ ในการ Labeling ให้โมเดล AI สำหรับการทำให้ Object Detection นั้นจะเรียกว่า Annotation โดยการทำให้ Annotation รูปภาพเรา จะใช้การติกรอบเพื่อบอกโมเดลว่าในรูปภาพนั้น มีวัตถุอะไรบ้าง และมีมันอยู่ตำแหน่งไหน โดยคณะผู้จัดทำได้ทำการ Annotation ด้วยเครื่องมือ roboflow ซึ่งเป็นเว็บแอปพลิเคชันสำหรับการ Annotation ออนไลน์ สามารถทำงานร่วมกันหลายคนพร้อมกันได้ เหมาะสำหรับการทำงานเป็นทีม สามารถปรับแต่งรูปภาพเพื่อเพิ่มความหลากหลายให้กับข้อมูล และสามารถ export ให้ข้อมูลอยู่ในรูปแบบที่ Framework การฝึกสอนโมเดลต่างๆ รับรองได้หลายรูปแบบ

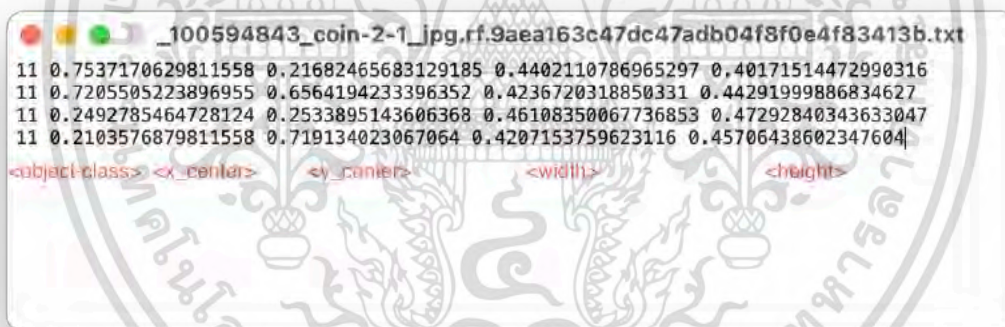


รูปที่ 3.26 ตัวอย่างการ Annotation รูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 ตัวอย่างการปรับแต่งรูปภาพ



รูปที่ 3.28 ตัวอย่างไฟล์รูปแบบ Label ของ Darknet

### 3.6.2. การฝึกสอนโมเดล Floating-point

ในการฝึกสอนโมเดล AI สามารถฝึกสอนได้หลากหลายรูปแบบ ซึ่งคณะผู้จัดทำได้เลือกการฝึกสอนด้วย Darknet Framework โดยใช้โครงสร้างของ YOLOv4 ในการตรวจจับวัตถุ เนื่องจากเป็นโมเดล AI ที่เทคโนโลยี DPU ของ Xilinx รองรับ และมีคุณสมบัติที่เหมาะสมกับการตรวจจับวัตถุในระบบนี้ โดยในการฝึกสอนโมเดล AI ด้วย Darknet จะต้องมีการปรับแต่ง CFG-files ซึ่งเป็นไฟล์ที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอกโครงสร้าง CNN การตั้งค่าในการฝึกสอน ซึ่งคณะผู้จัดทำได้ใช้ CFG ของ YOLOv4 และมีการปรับแต่งการตั้งค่าสำหรับการฝึกสอนเพื่อตรวจจับร้านค้า และสินค้า ดังนี้

### 1. การตั้งค่า CFG-files

- เปลี่ยนค่า batch เป็น *batch=64*
- เปลี่ยนค่า subdivisions เป็น *subdivisions=8*
- เปลี่ยนค่า *max\_batches* เป็น จำนวนคลาส \* 2000
  - เช่น 30 คลาส *max\_batches=60000*
- เปลี่ยนค่า steps เป็น 80% และ 90% ของ *max\_batches*
  - เช่น *max\_batches=60000 => steps=48000,54000*
- เปลี่ยนค่า ขนาด เป็นจำนวนที่สามารถหารด้วย 32 ลงตัว เช่น
  - *width=416*
  - *height=416*
- เปลี่ยนค่า *classes=80* เป็น จำนวนคลาสของ object ที่ต้องการตรวจจับใน 3 layers [yolo] เช่น object 30 คลาส

[yolo] layer 1

```
963 [yolo]
964 mask = 0,1,2
965 anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
966 classes=30
967 num=9
968 jitter=.3
```

[yolo] layer 2

```
1050 [yolo]
1051 mask = 3,4,5
1052 anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
1053 classes=30
1054 num=9
1055 jitter=.3
```

[yolo] layer 3

```
1137 [yolo]
1138 mask = 6,7,8
1139 anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
1140 classes=30
1141 num=9
1142 jitter=.3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เปลี่ยนค่า `filters=255` เป็น `filters=(จำนวนคลาส + 5)*3` ในเฉพาะ 3 layers `[convolutional]` สุดท้ายที่อยู่ก่อนหน้า `[yolo]` layer เท่านั้น
  - เช่น object 30 คลาส `filters=105`

[convolutional] layer 1

```

955 [convolutional]
956 size=1
957 stride=1
958 pad=1
959 filters=105
960 activation=linear
961
962
963 [yolo]
964 mask = 0,1,2
  
```

[convolutional] layer 2

```

1042 [convolutional]
1043 size=1
1044 stride=1
1045 pad=1
1046 filters=105
1047 activation=linear
1048
1049
1050 [yolo]
1051 mask = 3,4,5
  
```

## 2. การตั้งค่าไฟล์ data

- `classes` = จำนวนคลาส
- `train` = ที่อยู่ txt file ที่เป็น list ของข้อมูลภาพ train dataset
- `valid` = ที่อยู่ txt file ที่เป็น list ของข้อมูลภาพ valid dataset
- `names` = ที่อยู่ชื่อคลาสที่เป็น .names
- `backup` = ที่อยู่ Folder สำหรับเก็บ backup weight ที่เทรน

```

1 classes= 30
2 train = data/store/list/trainStore.txt
3 valid = data/store/list/valStore.txt
4 names = data/store.names
5 backup = cfg/yolov4_store/snapshots
6
  
```

รูปที่ 3.29 ตัวอย่างการตั้งค่าไฟล์ data ของ Darkent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากฝึกสอนโมเดลแล้วจำเป็นจะต้องมีการแปลงโมเดลที่ทำงานด้วย Darcnet Framework ซึ่งเป็นภาษา C ให้เป็น TensorFlow เนื่องจากเป็น Deep Learning Library ที่เครื่องมือ Vitis-AI รองรับ

### 3.6.3. การปรับแต่ง Floating-point โมเดลเป็น Fix-point โมเดล

หลังจากที่ฝึกสอนโมเดล AI จนได้โมเดลที่ตรงต่อความต้องการ และแปลงเป็นโมเดลที่ Vitis-AI รองรับแล้ว ขั้นตอนต่อไปคือการปรับแต่งโมเดลที่เป็น Floating-point ทำงานบน GPU ให้เป็น Fix-point เพื่อลดขนาดโมเดลให้สามารถทำงานบนการ์ด FPGA โดยที่ประสิทธิภาพการตรวจจับยังใกล้เคียงเท่าเดิม โดยการปรับแต่งโมเดลให้ทำงานกับเทคโนโลยี DPU ของ Xilinx ได้นั้นต้องใช้เครื่องมือที่ Xilinx เตรียมไว้ให้ นั่นคือ Vitis-AI Development Platform และใช้คำสั่งเพื่อเรียกใช้งานโปรแกรมการปรับแต่งโมเดล ซึ่งคณะผู้จัดทำได้ตั้งค่า option คำสั่งในการปรับแต่งโมเดลตามทฤษฎีในบทที่ 2 ดังนี้

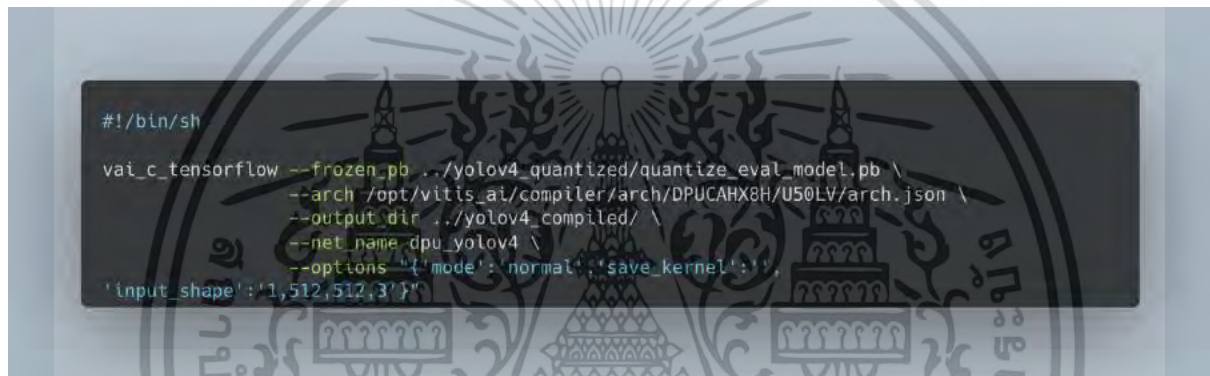
1. Quantization: ดำเนินกระบวนการแปลงโมเดลปัญญาประดิษฐ์ที่ทำงานบน GPU ให้เป็นโมเดล fixed-point โดยใช้คำสั่ง `vai_q_tensorflow quantize` และมี option ดังต่อไปนี้
  - `input_frozen_graph` = model ที่เป็น Floatin-point
  - `input_fn` = ฟังก์ชันการทำ preprocess รูปภาพก่อนประมวลผล
  - `output_dir` = ที่อยู่ในการเก็บโมเดลที่แปลงแล้ว
  - `input_nodes` = ชื่อ input node
  - `output_nodes` = ชื่อ output node
  - `input_shapes` = ขนาดของรูปภาพที่จะนำเข้าประมวลผล
  - `gpu` = GPU ตัวที่ใช้
  - `calib_iter` = จำนวนรอบในการเปรียบเทียบระหว่าง Floating-point โมเดล กับ Fix-point โมเดล

```
#!/bin/sh

vai_q_tensorflow quantize --input_frozen_graph ../tf_model/tf_model.pb \
--input_fn yolov4_graph_input_keras_fn.calib_input \
--output_dir ../yolov4_quantized \
--input_nodes image_input \
--output_nodes conv2d_93/BiasAdd,conv2d_101/BiasAdd,conv2d_109/BiasAdd \
--input_shapes ?,512,512,3 \
--gpu 0 \
--calib_iter 100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 3.30 ตัวอย่างการตั้งค่า option การ Quantization  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Compile: ขั้นตอนการ Compile โมเดล Fixed-point ให้เข้ากับการโครงสร้างการทำงานของ DPU แต่ละตระกูล โดยใช้คำสั่ง `vai_c_tensorflow` และมี option ดังต่อไปนี้
- `frozen_pb` = โมเดลที่ผ่านการ Quantization แล้ว
  - `arch` = ไฟล์โครงสร้างของ DPU ตระกูลที่ต้องการ Compile ซึ่งในโครงการนี้ใช้ U50LV มี DPU naming เป็น DPUCAHX8H
  - `output_dir` = ที่อยู่ในการเก็บโมเดลที่ Compile แล้ว
  - `net_name` = ชื่อโมเดล
  - `options` = option เพิ่มเติม



```
#!/bin/sh
vai_c_tensorflow --frozen_pb ../yolov4_quantized/quantize_eval_model.pb \
--arch /opt/vitis_ai/compiler/arch/DPUCAHX8H/U50LV/arch.json \
--output_dir ../yolov4_compiled/ \
--net_name dpu_yolov4 \
--options '{mode: normal, save_kernel: 1}' \
'input_shape': '1,512,512,3'
```

รูปที่ 3.31 ตัวอย่างการตั้งค่า option การ Compile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

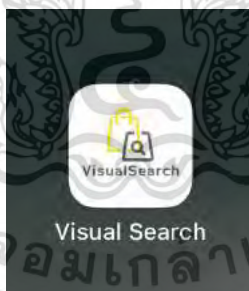
### ผลการวิจัยและการอภิปรายผล

ผลจากการดำเนินการพัฒนาระบบต้นแบบ Visual Search ตามที่ได้วางแผนไว้ สามารถแบ่งการพัฒนาออกได้เป็น 3 ส่วนดังนี้ 1) ส่วนของ Client เป็นระบบเบื้องหน้า (Front-end) ที่จัดทำในรูปแบบแอปพลิเคชันมือถือเพื่อให้ง่ายและสะดวกต่อการใช้งานในห้างสรรพสินค้า 2) ส่วนของ Server เป็นระบบเบื้องหลัง (Back-end) ที่จัดทำในรูปแบบ RESTful APIs สำหรับการติดต่อกับระบบเบื้องหน้า และเป็นตัวกลางในการประมวลผล Object Detection บนการ์ด FPGA 3) ส่วน Python Application เป็นส่วนที่ใช้ในการติดต่อเรียกใช้งานกับการ์ด FPGA ในการประมวลผล Object Detection อีกทั้งยังมีการทำ Functional Testing เพื่อทดสอบการทำงานโดยรวมของระบบ โดยมีรายละเอียดผลการวิจัยดังนี้

#### 4.1. Mobile Application

ในส่วนของการพัฒนา Mobile Application ทางคณะผู้จัดทำได้ดำเนินการพัฒนาแอปพลิเคชันตามความต้องการของระบบ และออกแบบหน้าจอตามแบบจำลองในหัวข้อที่ 3.5.1. Mobile Application Mockup ได้ผลลัพธ์การดำเนินการเป็นแอปพลิเคชันที่มีชื่อว่า Visual Search และมีรายละเอียดของแอปพลิเคชันดังต่อไปนี้

Apps icon



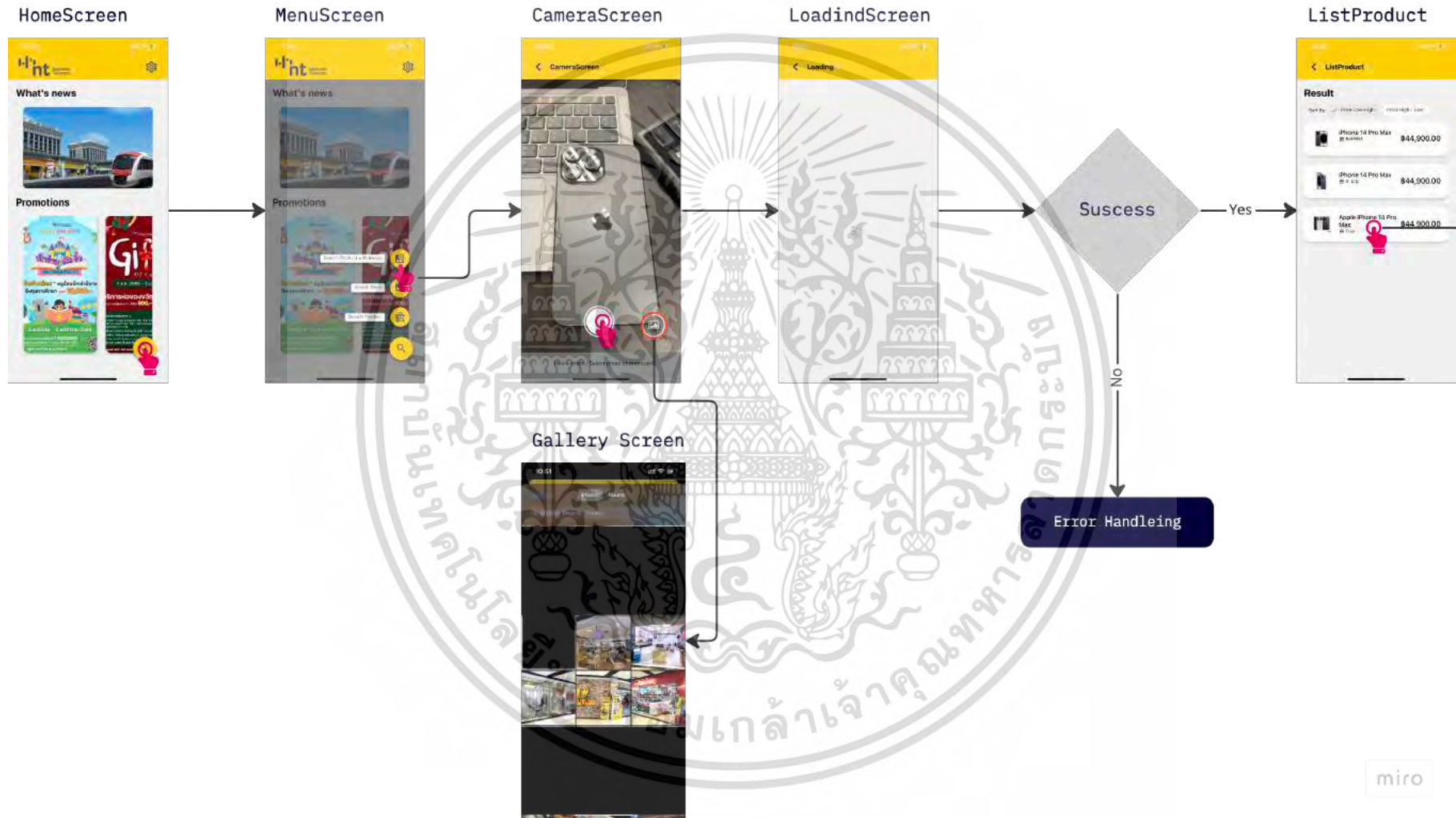
รูปที่ 4.1 Apps icon

#### UI Flow หน้าจอและขั้นตอนการใช้งาน

แอปพลิเคชัน Visual Search ที่ได้พัฒนาตามความต้องการของระบบ มี Feature ที่สามารถทำงานได้ดังนี้

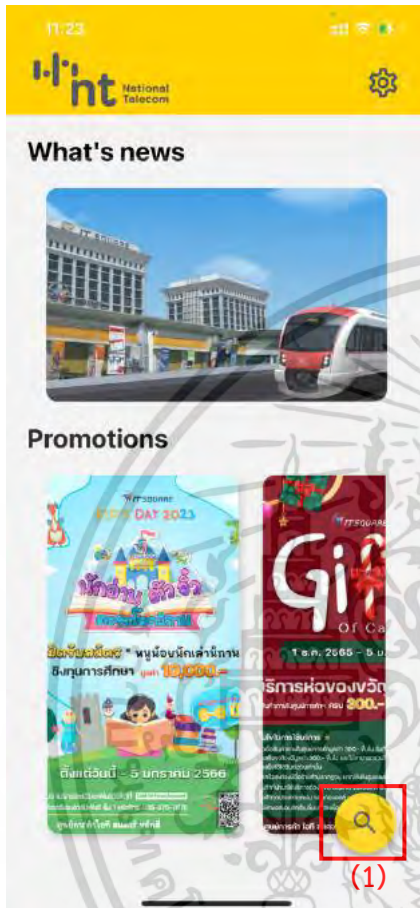
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1. Search Product with Image



รูปที่ 4.2 Search with Image Flow

Search with Image Flow เป็นขั้นตอนกระบวนการค้นหาสินค้าด้วยรูปภาพ โดยทุกหน้าจะสามารถกดลูกศรย้อนกลับเพื่อกลับไปหน้าที่ผ่านมาได้ และสามารถอธิบายเป็นขั้นตอนรายละเอียดได้ดังนี้



รูปที่ 4.3 รูปหน้าจอหลักเมื่อเข้าสู่แอปพลิเคชัน



รูปที่ 4.4 รูปหน้าจอหลักเมื่อกด Floating button

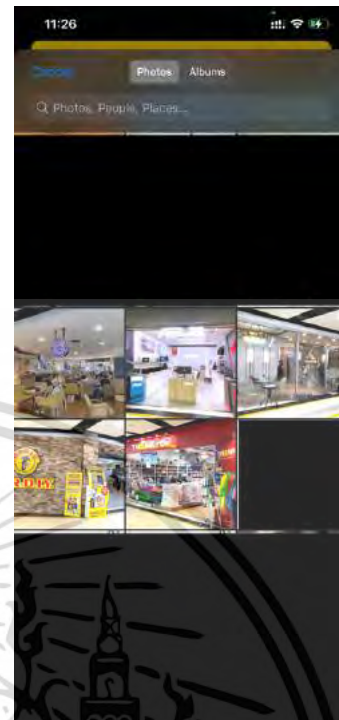
Home Screen เป็นหน้าจอหลักที่แสดงขึ้นเป็นหน้าแรกเมื่อผู้ใช้งานเปิดแอปพลิเคชัน ซึ่งจะแสดงข้อมูลข่าวสารและโปรโมชั่นต่างๆ ของห้างสรรพสินค้า โดย Home Screen ทำหน้าที่เป็นแหล่งรวมการเข้าใช้งานฟังก์ชันหลักต่างๆ ของแอปพลิเคชัน เช่น ฟังก์ชันการค้นหาสินค้าด้วยรูปภาพ ฟังก์ชันการค้นหาร้าน และฟังก์ชันการค้นหาสินค้า โดยผู้ใช้งานสามารถเข้าใช้งานฟังก์ชันหลักได้โดยเข้าผ่านเมนูใน Floating button (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

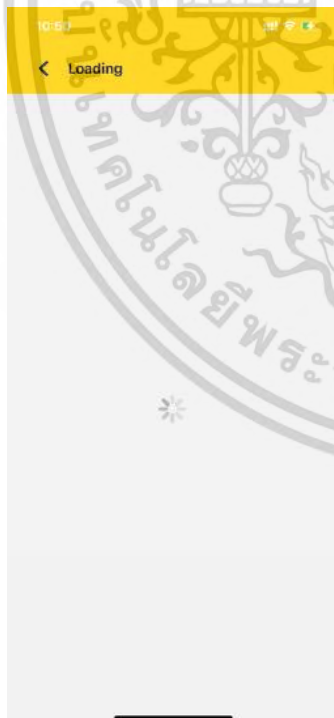
## 4.1.2. Search Product with Image Screen



รูปที่ 4.5 Camera Screen



รูปที่ 4.6 Gallery Screen



รูปที่ 4.7 Loading Screen



รูปที่ 4.8 List Screen

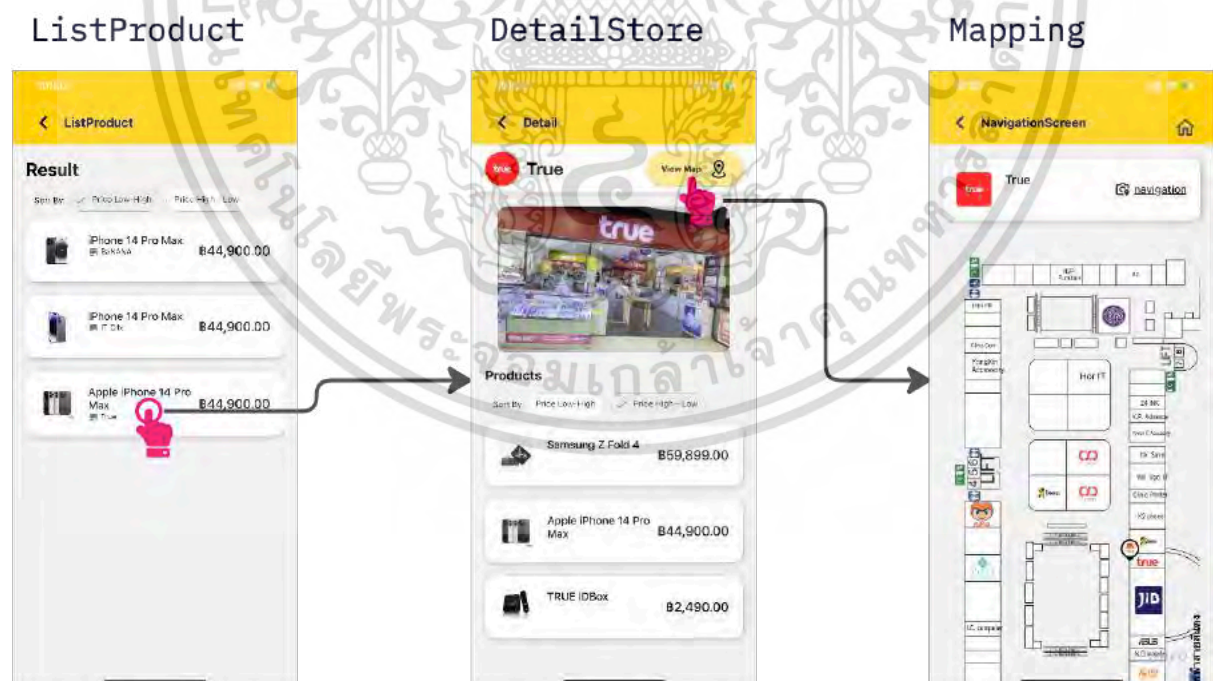
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้งานกดเข้ามาที่เมนู Search Product with Image (2) แอปพลิเคชันจะเรียกใช้งานฟังก์ชันการค้นหาสินค้าด้วยรูปภาพ โดยการทำงานของฟังก์ชันนี้จะเรียกใช้งาน Camera Screen ที่สามารถเรียกใช้งานกล้องหลังโทรศัพท์เพื่อถ่ายรูปลินค้า หรือเลือกรูปลินค้าจากในคลังรูปภาพได้ และส่งรูปภาพเข้าไปประมวลผลในระบบ Backend และแสดงรายการสินค้าที่ระบบตรวจพบ และสามารถกดที่รายการสินค้าเพื่อดูรายละเอียดของร้านค้าที่มีสินค้านั้นๆ ได้

#### ขั้นตอนการค้นหาสินค้าด้วยรูปภาพ

1. กดปุ่ม Search Product with Image (2) เข้าหน้า Camera Screen
2. กดปุ่ม Capture (3) เพื่อถ่ายรูปลินค้า หรือเลือกรูปลินค้าจากในคลังรูปภาพด้วยปุ่ม Gallery (4)
3. รอผลลัพธ์การประมวลผลจาก Server ในหน้า Loading Screen
4. ถ้าหากพบผลลัพธ์การตรวจจับจะแสดงผลรายการสินค้าที่ตรวจจับได้ในหน้า List Screen
5. หากไม่พบผลลัพธ์การตรวจจับหรือเกิดความผิดพลาดของระบบเบื้องหลังจะแสดงหน้าจอ Error Handling ในหัวข้อที่ 4.1.8 Error Handling

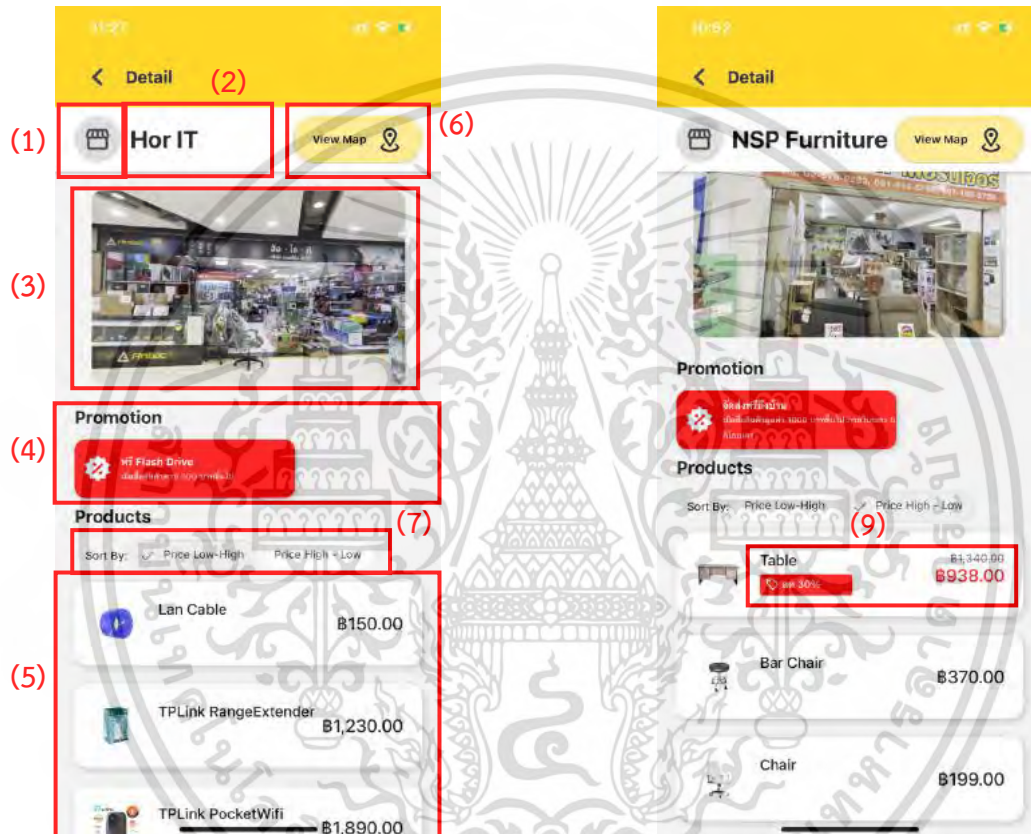
#### 4.1.3. Store Detail



รูปที่ 4.9 Store Detail Flow

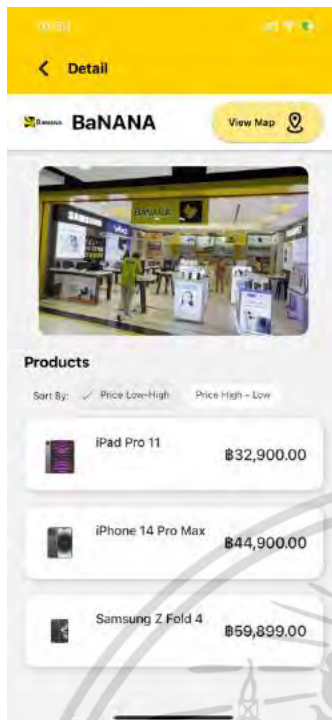
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Store Detail Flow เป็นขั้นตอนกระบวนการดูรายละเอียดร้านค้า หลังจากที่ใช้งานค้นหาและเข้าหน้า List Screen โดยหน้ารายละเอียดร้านค้าจะแสดง โลโก้ร้าน(1) ชื่อร้าน(2) รูปหน้าร้านค้า(3) โปรโมชั่นร้านค้า(4) และรายการสินค้าทั้งหมดในร้านค้า (5) พร้อมแสดงโปรโมชั่นของสินค้า (9) นั้นๆ โดยสามารถเรียงรายการสินค้าตามราคา (7) จากน้อยไปมาก หรือมากไปน้อยได้ และสามารถกดที่ปุ่ม View Map (6) เพื่อดูตำแหน่งที่อยู่ของร้านค้าได้ ซึ่งมีหน้าร้านทั้งหมด 28 ร้านในระบบ

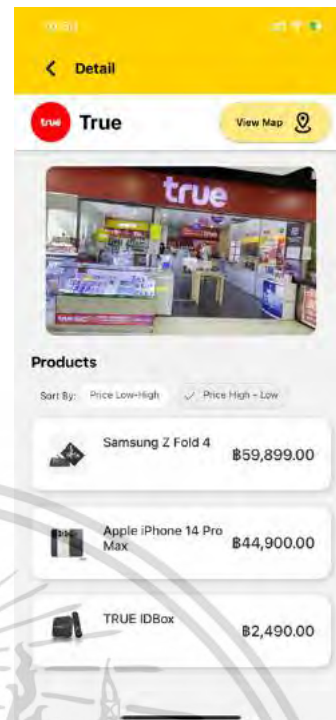


รูปที่ 4.10 ตัวอย่างรายละเอียดหน้าร้าน Hor IT รูปที่ 4.11 รายละเอียดหน้าร้าน NSP Furniture

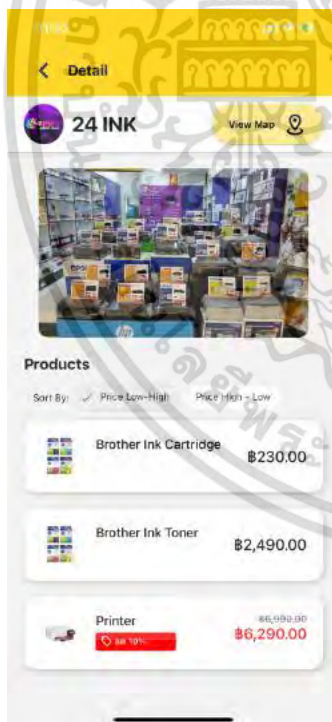
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 รายละเอียดหน้าร้าน BaNANA



รูปที่ 4.13 รายละเอียดหน้าร้าน True

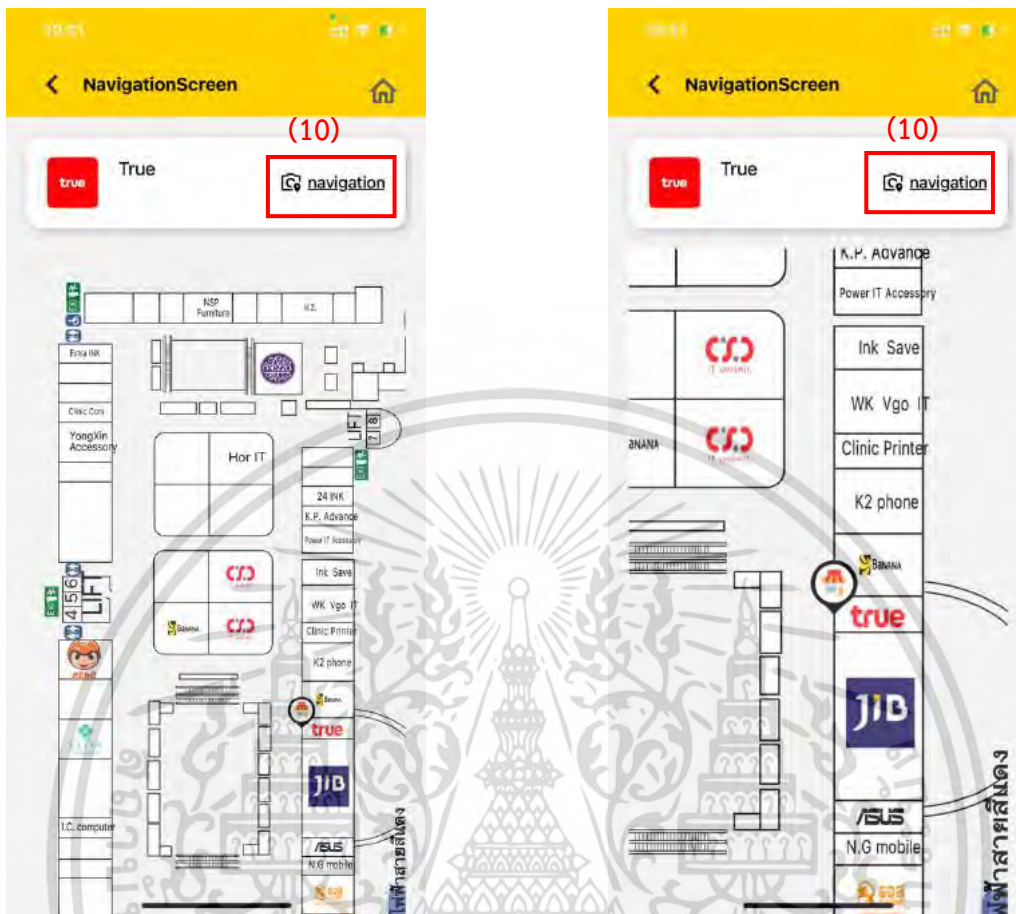


รูปที่ 4.14 รายละเอียดหน้าร้าน 24 INK

รูปเพิ่มเติมรูป  
รูปที่ 4.15 รายละเอียดหน้าร้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.1.4. Navigation Screen (ส่วนที่ 2 ของโครงการ)



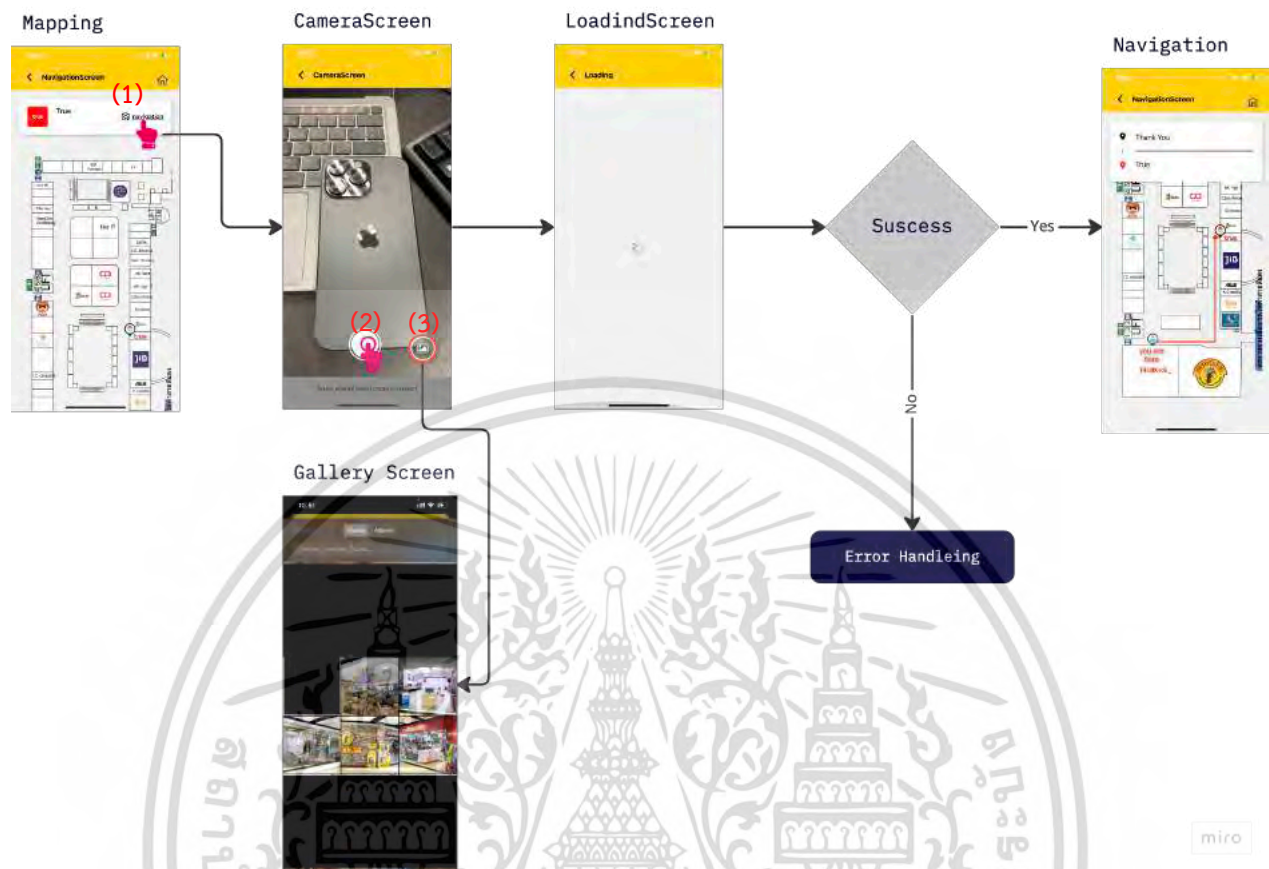
รูปที่ 4.16 แผนที่แสดงตำแหน่งร้าน True

รูปที่ 4.17 แผนที่แสดงตำแหน่งร้าน True แบบขยาย

หน้า Navigation Screen จะแสดงตำแหน่งของร้านค้าที่ต้องการโดยจะแสดงตำแหน่งร้านค้าเป็นรูปไอคอนร้านค้า สามารถขยายแผนที่เพื่อดูรูปให้ชัดเจนขึ้นได้ และสามารถกดปุ่ม navigation (10) เพื่อนำทางจากจุดที่ผู้ใช้งานอยู่ไปยังร้านค้าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.1.5. Navigation System Flow



รูปที่ 4.18 Navigation System Flow

เมื่อกดปุ่ม navigation (1) แอปพลิเคชันจะเรียกใช้งาน Camera Screen เพื่อใช้งานกล้องหลัง โทรศัพท์ถ่ายรูปหน้าร้านค้า หรือเลือกรูปหน้าร้านค้าจากในคลังรูปภาพในหน้า Gallery Screen และส่งรูปภาพเข้าไปประมวลผลในระบบ Backend และระบุตำแหน่งหน้าร้านที่ระบบตรวจพบ จากนั้นส่งข้อมูลกลับไปให้ระบบ Navigation ประมวลผลหาเส้นทางที่สั้นที่สุดจากตำแหน่งปัจจุบันไปยังร้านค้าที่ต้องการได้

ขั้นตอนการระบุตำแหน่งหน้าร้านด้วยรูปภาพ

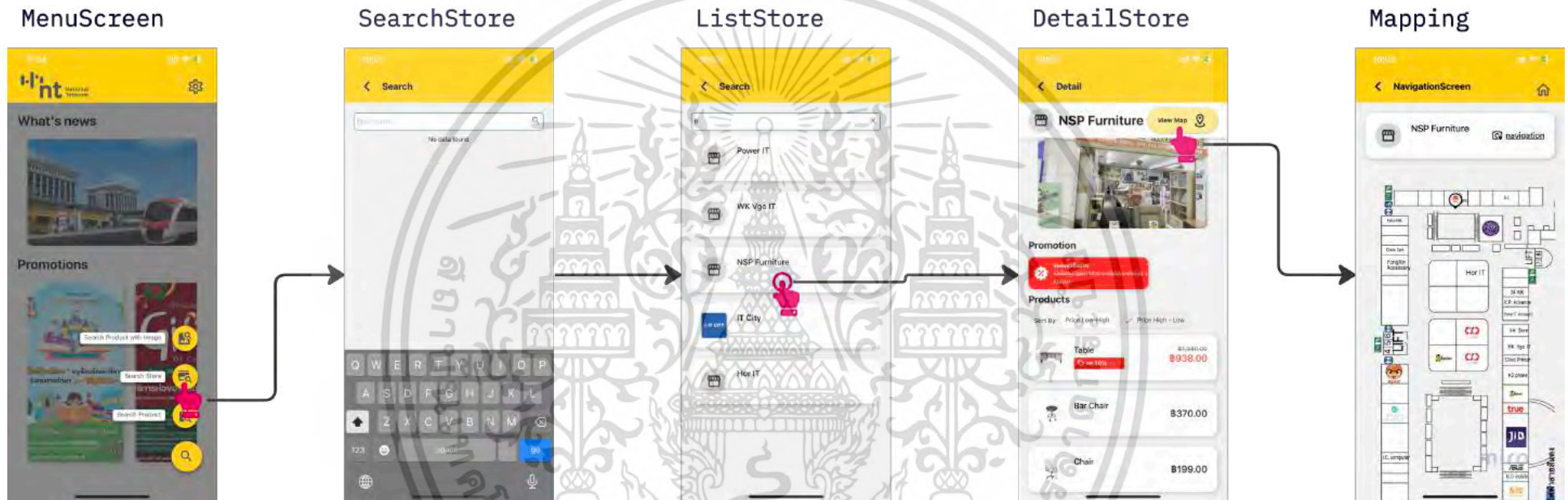
1. กดปุ่ม navigation (1) เข้าหน้า Camera Screen
2. กดปุ่ม Capture (2) เพื่อถ่ายรูปหน้าร้านค้าหรือเลือกรูปหน้าร้านค้าจากในคลังรูปภาพด้วยปุ่ม Gallery (4)
3. รอผลลัพธ์การประมวลผลจาก Server ในหน้า Loading Screen
4. ถ้าพบผลลัพธ์การตรวจจับจะส่งผลลัพธ์การประมวลผลไปยังระบบ Navigation เพื่อนำทางผู้ใช้งานในหน้า Navigation Screen

5. หากไม่พบผลลัพธ์การตรวจจับหรือเกิดความผิดพลาดของระบบเบื้องหลังจะแสดงหน้าจอเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

Error Handling ในหัวข้อที่ 4.1.8 Error Handling

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่มีเหตุแต่สิ่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

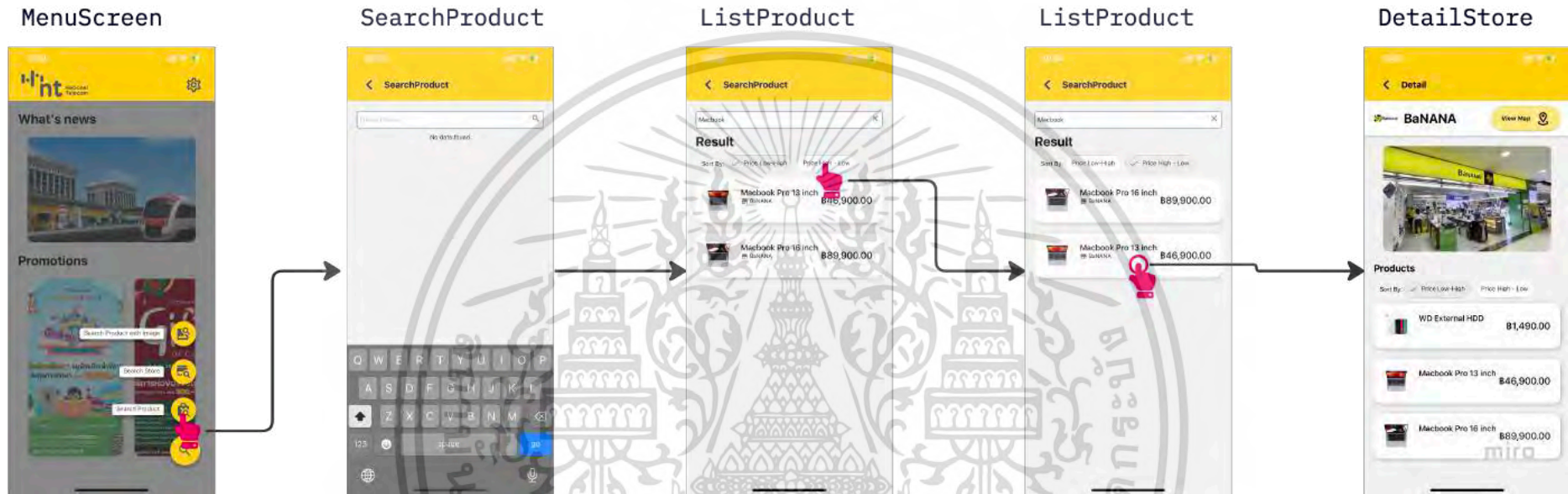
## 4.1.6. Search Store



รูปที่ 4.19 Search Store Flow

เมื่อผู้ใช้งานกดเข้ามาที่เมนู Search Store แอปพลิเคชันจะเข้าไปที่หน้า Search Screen ผู้ใช้งานสามารถพิมพ์ชื่อร้านเพื่อค้นหาข้อมูลร้านในฐานข้อมูลได้โดยระบบจะค้นหาแบบ Insensitive case แสดงรายการร้านค้าที่ใกล้เคียงกับชื่อที่ค้นหา และสามารถกดที่รายการร้านค้าเพื่อแสดงรายละเอียดและใช้งานระบบ Navigation ได้

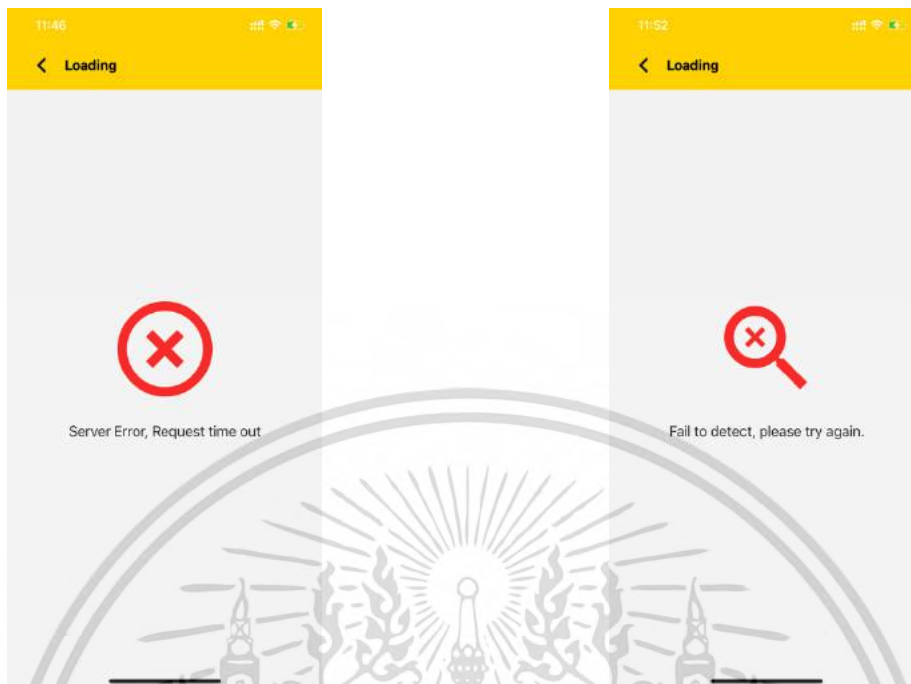
## 4.1.7. Search Product



รูปที่ 4.20 Search Product Flow

เมื่อผู้ใช้งานกดเข้ามาที่เมนู Search Product แอปพลิเคชันจะเข้าไปที่หน้า Search Screen ผู้ใช้งานสามารถพิมพ์ชื่อสินค้าเพื่อค้นหาข้อมูลสินค้าในฐานข้อมูลได้ โดยระบบจะค้นหาแบบ Insensitive case แสดงรายการสินค้าที่ใกล้เคียงกับชื่อที่ค้นหา โดยสามารถเรียงรายการสินค้าตามราคาจากน้อยไปมากหรือมากไปน้อยได้ และสามารถกดที่รายการสินค้าเพื่อแสดงรายละเอียดร้านค้าและใช้งานระบบ Navigation ได้

## 4.1.8. Error Handling



รูปที่ 4.21 Server Error

รูปที่ 4.22 Fail to detect

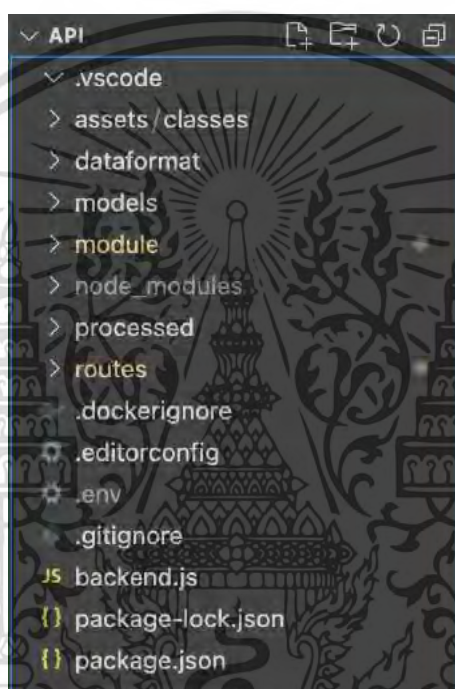
การจัดการข้อผิดพลาดของแอปพลิเคชันนี้จะมีอยู่ 2 รูปแบบใหญ่ๆ ด้วยกันคือ

1. Server Error (ความผิดพลาดที่เกิดขึ้นจาก Server) โดยจะเกิดขึ้นเมื่อแอปพลิเคชันไม่สามารถติดต่อกับ Server ได้ หรือส่งรูปภาพไปประมวลกับ Server และ Server ไม่สามารถตอบสนองได้ตามเวลาที่กำหนดก็จะเกิดข้อผิดพลาดขึ้นและแจ้งให้กับผู้ใช้งานทราบ
2. Fail to detect (ความผิดพลาดที่เกิดขึ้นจากโมเดลตรวจจับวัตถุ) โดยจะเกิดขึ้นเมื่อแอปพลิเคชันส่งรูปภาพไปประมวลกับ Server และ Application Object Detection ไม่สามารถตรวจจับวัตถุได้ หรือไม่มีข้อมูลวัตถุอยู่ในโมเดล Server จะตอบกลับแอปพลิเคชันว่าไม่สามารถตรวจจับได้และแจ้งให้กับผู้ใช้งานทราบให้ลองใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2. Server-side

เป็นระบบเบื้องหลังที่พัฒนาแบบ RESTful APIs ใช้ในการสื่อสารและแลกเปลี่ยนข้อมูลโดยใช้ HTTP ตามมาตรฐานของ REST (Representational State Transfer) ซึ่งเป็นรูปแบบการแบ่งแยกความสัมพันธ์ระหว่าง client และ server โดยใช้ URI และ HTTP Method (GET, POST, PUT, DELETE) ในการอ้างอิงและจัดการกับข้อมูลในฐานข้อมูลโดยปฏิบัติตามหลักการ CRUD (Create, Read, Update, Delete) นอกจากนี้ยังเป็นระบบเบื้องหลังที่ช่วยในการเป็นส่วนประสาน Input ให้กับการทำ Object Detection และส่งผลลัพธ์ให้กับ client โดยคณะผู้จัดทำได้ออกแบบโครงสร้างระบบเบื้องหลังดังนี้



รูปที่ 4.23 โครงสร้างไฟล์ระบบเบื้องหลัง

ตารางที่ 4.1 ตารางอธิบายโครงสร้างไฟล์ระบบเบื้องหลัง

ชื่อ	คำอธิบาย
assets	Folder สำหรับเก็บทรัพยากรต่างๆที่จำเป็นในระบบ API
dataformat	Folder ไฟล์รูปแบบของข้อมูลใน Database
models	สำหรับไฟล์ Models ใน Database โดยแต่ละไฟล์จะแทนใช้การเรียกใช้ Table
module	รวมไฟล์ Functions สนับสนุนต่างๆ
routes	Folder ไฟล์ route API ต่างๆ
.editorconfig	ไฟล์เก็บค่าตัวแปรสำหรับการจัดรูปแบบ code
.env	ไฟล์เก็บค่าตัวแปรสำหรับการตั้งค่า environment ระบบ
backend.js	ไฟล์เริ่มต้นการทำงาน Web Server

package.json	ไฟล์บันทึก Package ไฟล์ที่จำเป็นในการใช้งานระบบ
--------------	---

## Database ในระบบ

ตารางที่ 4.2 ตารางอธิบาย Table ใน Database

ชื่อตาราง	คำอธิบาย
products	Table สำหรับเก็บข้อมูลสินค้า โดยประกอบไปด้วย id (pk), storeId(fk), promoProductId(fk), productName, imagePath, imageName, price
stores	Table สำหรับเก็บข้อมูลร้านค้า โดยประกอบไปด้วย id(pk), productId(fk), promoStoreId(fk), storeName, imagePath, imageName, logoPath, logoName
PromoProducts	Table สำหรับเก็บข้อมูล Promotion ของสินค้า โดยประกอบไปด้วย id(pk), productId(fk), title, promotionPrice, startDate, endDate
PromoStores	Table สำหรับเก็บข้อมูล Promotion ของร้านค้า โดยประกอบไปด้วย id(pk), storeId(fk), title, description, startDate, endDate

## API Definition

### 4.2.1. API ประมวลผลรูปภาพ

ตารางที่ 4.3 ตารางอธิบาย API ประมวลผลรูปภาพ

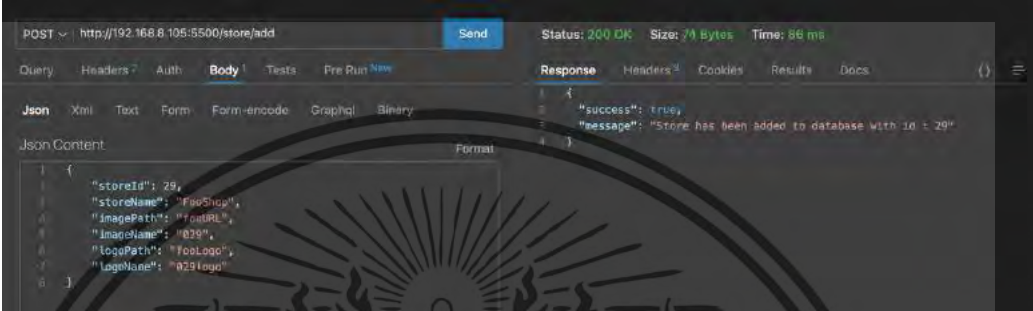
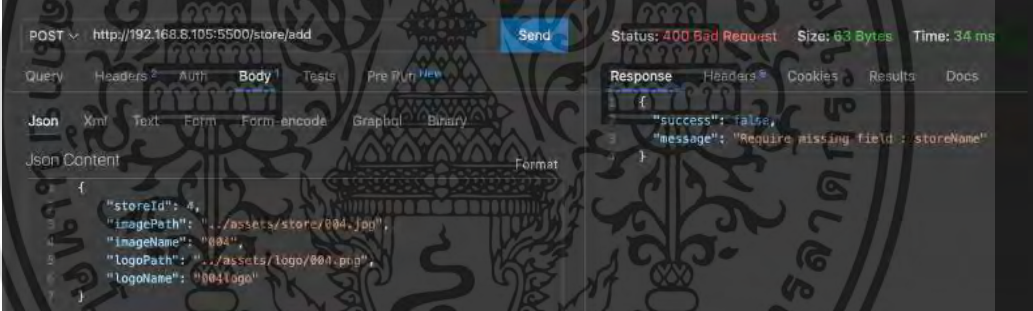
API ID	01						
API Name	ประมวลผลรูปภาพ						
Endpoint	/detection						
HTTP method	POST						
Variable Request	Body JSON: <table border="1" data-bbox="405 1688 1366 1845"> <thead> <tr> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>base64</td> <td>รูปภาพในรูปแบบ Base64</td> </tr> <tr> <td>type</td> <td>ตัวแปรสำหรับระบุประเภทการใช้งานโมเดล Object Detection</td> </tr> </tbody> </table>	Property	Description	base64	รูปภาพในรูปแบบ Base64	type	ตัวแปรสำหรับระบุประเภทการใช้งานโมเดล Object Detection
Property	Description						
base64	รูปภาพในรูปแบบ Base64						
type	ตัวแปรสำหรับระบุประเภทการใช้งานโมเดล Object Detection						
Description	รับ Variable Request ผ่าน Body JSON โดย API จะแปลง Base64 เป็น image/jpg ไฟล์แล้วนำไปเก็บไว้ใน Storage เครื่อง Server โดยมีการอ้างอิงรูปภาพจาก Client ที่ติดต่อมาด้วยการตั้งชื่อไฟล์รูปภาพเป็น ID ตัวอักษรภาษาอังกฤษที่ไม่ซ้ำกัน จากนั้นรอให้ Python Application นำไปใช้						

	ประมวลผลรูปภาพและนำผลลัพธ์การทำงานที่ได้มาค้นหาข้อมูลใน Database ส่งข้อมูลที่ค้นเจอกลับให้ Client
Sample Response	<p>ผลลัพธ์ตรวจจับรูปภาพสินค้า</p> <pre>{   "labels": [     "externalhdd"   ],   "results": [     {       "PromoProduct": null,       "id": "78",       "imageName": "0078",       "imagePath":       "https://firebasestorage.googleapis.com/v0/b/sahakit-       project.appspot.com/o/product%2FWD_ExternalHDD.jpg?alt=media&amp;token=b0a58a7c-       94e1-420f-b5b6-181c79366838",       "price": 1490,       "productName": "WD External HDD",       "store": [Object],       "storeId": "26"     }   ],   "success": true }</pre> <p>ผลลัพธ์ตรวจจับรูปภาพร้านค้า</p> <pre>{   "id": "17",   "storeName": "mezzo" }</pre>

#### 4.2.2. API เพิ่มข้อมูลร้านค้า

ตารางที่ 4.4 ตารางอธิบาย API เพิ่มข้อมูลร้านค้า

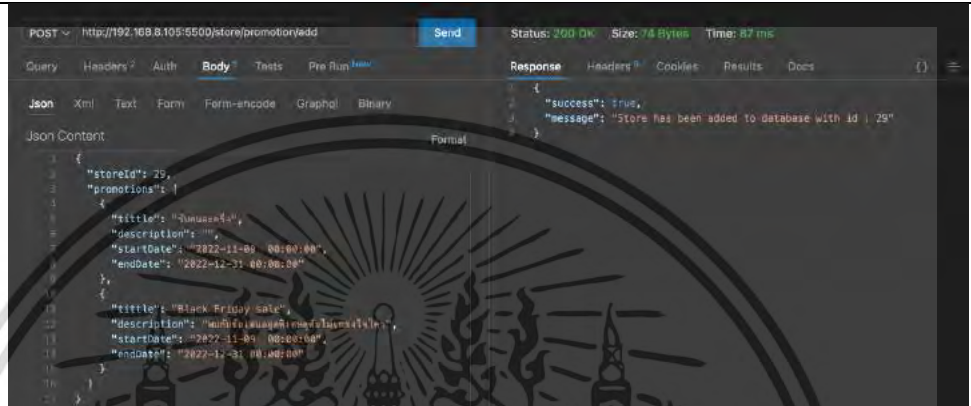
API ID	02																
API Name	เพิ่มข้อมูลร้านค้า																
Endpoint	/store/add																
Variable Request	<p>Body JSON:</p> <table border="1"> <thead> <tr> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>storeId</td> <td>Id ร้านค้าหากไม่ระบุไว้ id จะเพิ่มขึ้นเองอัตโนมัติ (optional)</td> </tr> <tr> <td>storeName</td> <td>ชื่อร้านค้า (require)</td> </tr> <tr> <td>imagePath</td> <td>ที่อยู่ URL รูปภาพหน้าร้านค้า (require)</td> </tr> <tr> <td>imageName</td> <td>ชื่อรูปภาพหน้าร้านค้า (require)</td> </tr> <tr> <td>logoPath</td> <td>ที่อยู่ URL รูปภาพโลโก้ร้านค้า (require)</td> </tr> <tr> <td>logoName</td> <td>ชื่อรูปภาพโลโก้ร้านค้า (require)</td> </tr> <tr> <td>promotions</td> <td>Arrays โปรโมชันของร้านค้า (optional)</td> </tr> </tbody> </table>	Property	Description	storeId	Id ร้านค้าหากไม่ระบุไว้ id จะเพิ่มขึ้นเองอัตโนมัติ (optional)	storeName	ชื่อร้านค้า (require)	imagePath	ที่อยู่ URL รูปภาพหน้าร้านค้า (require)	imageName	ชื่อรูปภาพหน้าร้านค้า (require)	logoPath	ที่อยู่ URL รูปภาพโลโก้ร้านค้า (require)	logoName	ชื่อรูปภาพโลโก้ร้านค้า (require)	promotions	Arrays โปรโมชันของร้านค้า (optional)
Property	Description																
storeId	Id ร้านค้าหากไม่ระบุไว้ id จะเพิ่มขึ้นเองอัตโนมัติ (optional)																
storeName	ชื่อร้านค้า (require)																
imagePath	ที่อยู่ URL รูปภาพหน้าร้านค้า (require)																
imageName	ชื่อรูปภาพหน้าร้านค้า (require)																
logoPath	ที่อยู่ URL รูปภาพโลโก้ร้านค้า (require)																
logoName	ชื่อรูปภาพโลโก้ร้านค้า (require)																
promotions	Arrays โปรโมชันของร้านค้า (optional)																

HTTP method	POST
Description	รับ Variable Request ผ่าน Body JSON โดยเป็น API สำหรับเพิ่มข้อมูลร้านค้าในฐานข้อมูลโดยจะมีการตรวจสอบความครบถ้วนของข้อมูลเบื้องต้นก่อนอนุญาตให้เพิ่มข้อมูลลงใน Database ข้อมูล Attribute ที่จำเป็นได้แก่ storeId, storeName, imagePath, imageName, logoPath, logoName เมื่อดำเนินการเสร็จสิ้น API จะส่ง response ผลลัพธ์การดำเนินการกลับให้ Client
Sample Response	<p>เพิ่มข้อมูลร้านค้าใหม่</p>  <p>การเพิ่มข้อมูลเข้าฐานข้อมูลจะมีการตรวจสอบความครบถ้วนของข้อมูลเบื้องต้น หากมีข้อมูลในช่องที่จำเป็นขาดหายไป Server ก็จะตอบกลับให้ Client ทราบว่าขาดข้อมูลในช่องใด</p> 

#### 4.2.3. API เพิ่มข้อมูลโปรโมชั่นของร้านค้า

ตารางที่ 4.5 ตารางอธิบาย API เพิ่มข้อมูลโปรโมชั่นของร้านค้า

API ID	03	
API Name	เพิ่มข้อมูลโปรโมชั่นของร้านค้า	
Endpoint	/store/promotion/add	
Variable Request	Body JSON:	
	Property	Description
	storeId	Id ร้านค้าที่เป็นเจ้าของโปรโมชั่น (require)
	title	ชื่อโปรโมชั่นร้านค้า (require)
	description	คำอธิบายโปรโมชั่น (optional)

	startDate	วันและเวลาเริ่มต้นของโปรโมชั่น (require)
	endDate	วันและเวลาสิ้นสุดของโปรโมชั่น (require)
HTTP method	POST	
Description	รับ Variable Request ผ่าน Body JSON โดยเป็น API สำหรับเพิ่มข้อมูลโปรโมชั่นของร้านค้าที่มีอยู่ในฐานข้อมูลโดยอ้างอิงจาก storeId เป็น foreign key จากนั้นส่ง response ผลลัพธ์การดำเนินการกลับให้ Client	
Sample Response	 <pre> POST http://192.168.8.105:5500/store/promotion/add Status: 200 OK, Size: 74 Bytes, Time: 87 ms  Query Headers Auth Body Tests Pre Run Response Headers Cookies Results Docs Json Xml Text Form Form-encode Graphql Binary Json Content {   "storeId": 29,   "promotions": [     {       "title": "วันศุกร์",       "description": "",       "startDate": "2022-11-09 00:00:00",       "endDate": "2022-11-11 23:59:59"     },     {       "title": "Black Friday sale",       "description": "โปรโมชั่นสุดพิเศษวันศุกร์",       "startDate": "2022-11-09 00:00:00",       "endDate": "2022-11-11 23:59:59"     }   ] }  {   "success": true,   "message": "Store has been added to database with id: 29" } </pre>	

#### 4.2.4. API แสดงรายการร้านค้าทั้งหมด

ตารางที่ 4.6 ตารางอธิบาย API แสดงรายการร้านค้าทั้งหมด

API ID	04
API Name	แสดงรายการร้านค้าทั้งหมด
Endpoint	/store/getall
Variable Request	-
HTTP method	GET
Description	เป็น API สำหรับ Query ข้อมูลร้านค้าทั้งหมดใน Table stores รวมถึงข้อมูลโปรโมชั่นร้านค้า ไม่รวมข้อมูลสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample Response	<pre> {   "storeId": "1",   "storeName": "Thank You",   "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/store%2Ffront%2F001.png?alt=media&amp;token=25dfcf89-9b6c-4bef-8416-46fe748cb33b",   "imageName": "001",   "logoPath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/store%2Flogo%2F001.jpg?alt=media&amp;token=376e7b48-2252-42e6-80f4-82fb23306292",   "logoName": "001logo",   "promotions": [     {       "id": "1",       "title": "ซื้อครบ 500 แพนน้ำใจ",       "description": "รับน้ำแข็งฟรี",       "startDate": "2022-11-09T07:00:00.000Z",       "endDate": "2022-11-20T07:00:00.000Z",       "storeId": "1"     },     {       "id": "2",       "title": "ซื้อครบ 1000 แพนน้ำดื่ม",       "description": "รับน้ำแข็งจากเทือกเขาอัลไต",       "startDate": "2022-11-09T07:00:00.000Z",       "endDate": "2022-11-20T07:00:00.000Z",       "storeId": "1"     }   ] }, {   "storeId": "2",   "storeName": "MR.DIY",   "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/store%2Ffront%2F002.png?alt=media&amp;token=25dfcf89-9b6c-4bef-8416-46fe748cb33b",   "imageName": "002",   "logoPath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/store%2Flogo%2F002.png?alt=media&amp;token=376e7b48-2252-42e6-80f4-82fb23306292",   "logoName": "002logo",   "promotions": [     {       "id": "3",       "title": "ถ้วย FoodContainer และ Bottle", </pre>
-----------------	--

#### 4.2.5. API แสดงรายละเอียดร้านค้า

ตารางที่ 4.7 ตารางอธิบาย API แสดงรายละเอียดร้านค้า

API ID	05
API Name	แสดงรายละเอียดร้านค้า
Endpoint	/store/get:id
Variable Request	Query Parameters: storeId
HTTP method	GET
Description	รับ Query Parameters เป็น storeId ใช้สำหรับการ Query รายละเอียดข้อมูลร้านค้า โดยใช้ ID ในการอ้างอิงกับข้อมูลร้านค้าในฐานข้อมูลที่ต้องการ รวมถึงข้อมูลโปรโมชั่นของร้านค้าและข้อมูลสินค้า
Sample Response	<p>ผลการดึงข้อมูลร้านค้า id ที่ 2 จาก Database</p> <pre> {   "id": "2",   "storeName": "MR.DIY",   "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/store%2Ffront%2F002.png?alt=media&amp;token=25dfcf89-9b6c-4bef-8416-46fe748cb33b",   "imageName": "002",   "logoPath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/store%2Flogo%2F002.png?alt=media&amp;token=376e7b48-2252-42e6-80f4-82fb23306292",   "logoName": "002logo",   "PromoStores": [ </pre>

เอกสารนี้เป็นเอกสาร  
ไม่ว่ากรณีใดๆ ทั้ง

	<pre> {   "id": "3",   "title": "ชื่อ FoodContainer แคม Bottle",   "description": "รับขวดน้ำลายบี๊กอ๊ฟ ทันทีเมื่อซื้อ กล่องอาหารพิเศษของโคราเอมอน",   "startDate": "2022-11-09T07:00:00.000Z",   "endDate": "2023-01-31T07:00:00.000Z",   "storeId": "2" }, {   "id": "4",   "title": "รับคนละครึ่ง",   "description": "",   "startDate": "2022-11-09T07:00:00.000Z",   "endDate": "2022-12-31T07:00:00.000Z",   "storeId": "2" } ], "products": [   {     "id": "6",     "productName": "Bottle",     "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/product%2FBottle.jpeg?alt=media&amp;token=fd518690-2dcd-47ca-9e69-13eeef2d6e8",     "imageName": "0006",     "price": 17,     "storeId": "2",     "PromoProduct": null   },   {     "id": "4",     "productName": "Scissors",     "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/product%2FScissors.jpg?alt=media&amp;token=b63c5bf8-6a55-49dc-b727-96f69dad8d6c",     "imageName": "0004",     "price": 20,     "storeId": "2",     "PromoProduct": null   },   {     "id": "5",     "productName": "Food Container",     "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/product%2FFoodContainer.jpeg?alt=media&amp;token=bbb15fac-2e1a-4150-961d-c6fdeace9026",     "imageName": "0005",     "price": 30,     "storeId": "2",     "PromoProduct": null   } ] } </pre>
--	--

#### 4.2.6. API ค้นหาร้านค้าด้วยชื่อ

ตารางที่ 4.8 ตารางอธิบาย API ค้นหาร้านค้าด้วยชื่อ

API ID	06
API Name	ค้นหาร้านค้าด้วยชื่อ
Endpoint	/store/search:query
Variable Request	Query Parameters:

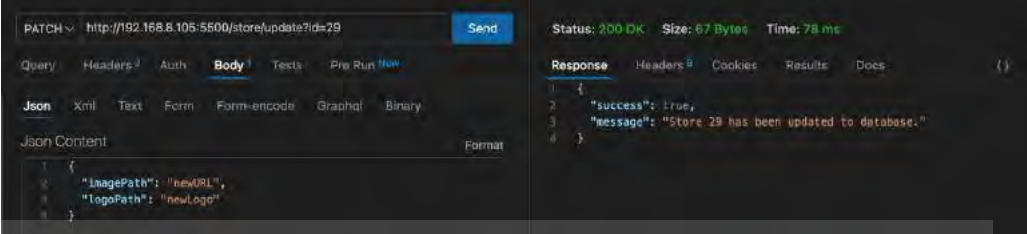
	query
HTTP method	GET
Description	รับ Query Parameters เป็นตัวอักษรหรือชื่อร้านที่ต้องการค้นหา โดย API จะทำการค้นหาข้อมูลชื่อร้านที่มีส่วนประกอบของตัวอักษรที่รับมาจาก Query Parameters ซึ่งค้นหาแบบ Insensitive Case และส่งข้อมูล ID ชื่อร้าน และรูปโลโก้ ที่ค้นพบทั้งหมดให้กับ Client
Sample Response	<p>ผลการค้นหาข้อมูลร้านค้าที่มีชื่อประกอบไปด้วยตัวอักษร ink จาก Database</p> <pre> {   "msg": "search results",   "results": [     {       "id": "13",       "storeName": "Ink Save",       "logoPath":       "https://firebasestorage.googleapis.com/v0/b/sahakit-       project.appspot.com/o/store%2Flogo%2F013.jpg?alt=media&amp;token=c8f65af1-       632d-4196-9069-18fa07561d0e"     },     {       "id": "16",       "storeName": "24 INK",       "logoPath":       "https://firebasestorage.googleapis.com/v0/b/sahakit-       project.appspot.com/o/store%2Flogo%2F016.jpg?alt=media&amp;token=4bb5b3b5-       8599-4e14-8674-80e9d3e72f67"     },     {       "id": "20",       "storeName": "Extra INK",       "logoPath": ""     }   ] } </pre>

#### 4.2.7. API อัปเดตข้อมูลร้านค้า

ตารางที่ 4.9 ตารางอธิบาย API อัปเดตข้อมูลร้านค้า

API ID	07						
API Name	อัปเดตข้อมูลร้านค้า						
Endpoint	/store/update:id						
Variable Request	<p>Query Parameters:</p> <p>storeId</p> <p>Body JSON:</p> <table border="1"> <thead> <tr> <th>Property</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>imagePath</td> <td>ที่อยู่ URL รูปภาพหน้าร้านค้า (require)</td> </tr> <tr> <td>logoPath</td> <td>ที่อยู่ URL รูปภาพโลโก้ร้านค้า (require)</td> </tr> </tbody> </table>	Property	Description	imagePath	ที่อยู่ URL รูปภาพหน้าร้านค้า (require)	logoPath	ที่อยู่ URL รูปภาพโลโก้ร้านค้า (require)
Property	Description						
imagePath	ที่อยู่ URL รูปภาพหน้าร้านค้า (require)						
logoPath	ที่อยู่ URL รูปภาพโลโก้ร้านค้า (require)						
HTTP method	PATCH						

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ควรคัดลอก หรือใช้ซ้ำโดยไม่ได้รับอนุญาต

Description	รับ Query Parameters เป็น storeId ที่ใช้ในการอ้างอิงร้านค้าที่ต้องการแก้ไข รับ Body เป็นข้อมูลที่ต้องการแก้ไข โดย API จะตรวจสอบความถูกต้องของข้อมูลเบื้องต้นและอัปเดตข้อมูลร้านค้าตามข้อมูลที่รับมาเข้าไปในฐานข้อมูลของระบบ จากนั้นส่ง response ผลลัพธ์การดำเนินการกลับให้ Client
Sample Response	 <pre> PATCH http://192.168.8.105:5600/store/update?id=29 Body {   "imagePath": "newURL",   "logoPath": "newLogo" } Response {   "success": true,   "message": "Store 29 has been updated to database." } </pre>

#### 4.2.8. API ลบข้อมูลร้านค้า

ตารางที่ 4.10 ตารางอธิบาย API ลบข้อมูลร้านค้า


API ID	08
API Name	ลบข้อมูลร้านค้า
Endpoint	/store/delete:id
Variable Request	Query Parameters: storeId
HTTP method	DELETE
Description	ลบข้อมูลร้านค้าใน Database โดยใช้ Query Parameters: storeId ที่รับมาเป็น id ในการอ้างอิงกับข้อมูลร้านค้าที่ต้องการลบ จากนั้นส่ง response ผลลัพธ์การดำเนินการกลับให้ Client
Sample Response	 <pre> DELETE http://192.168.8.105:5600/store/delete?id=29 Query Parameters id 29 Response {   "success": true,   "message": "Store 29 has been deleted from database." } </pre>

#### 4.2.9. API เพิ่มข้อมูลสินค้า

ตารางที่ 4.11 ตารางอธิบาย API เพิ่มข้อมูลสินค้า

API ID	09
API Name	เพิ่มข้อมูลสินค้า
Endpoint	/product/add
Variable Request	Body JSON:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

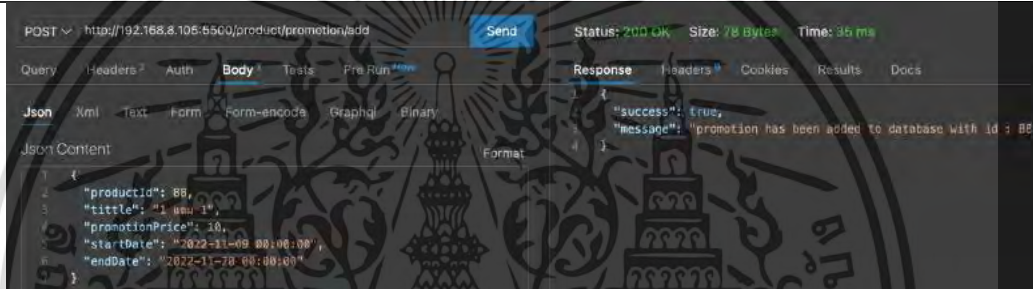
	Property	Description
	storeId	Id ร้านค้าที่เป็นเจ้าของข้อมูลสินค้า (require)
	productName	ชื่อสินค้า (require)
	imagePath	ที่อยู่ URL รูปภาพสินค้า (require)
	imageName	ชื่อรูปภาพสินค้า (require)
	price	ราคาของสินค้า (require)
	promotions	โปรโมชั่นของสินค้า (optional)
HTTP method	POST	
Description	เป็น API เพิ่มข้อมูลสินค้าลงในฐานข้อมูล โดยรับ Variable Request ผ่าน Body JSON ในรูปแบบ Array มีการตรวจสอบความครบถ้วนของข้อมูลเบื้องต้นก่อนอนุญาตให้เพิ่มข้อมูลลงใน ฐานข้อมูล และสามารถเพิ่มข้อมูลสินค้าหลายชิ้นได้ใน Request เดียว ข้อมูล Attribute ที่จำเป็นได้แก่ storeId, productName, imagePath, imageName, price เมื่อดำเนินการเสร็จสิ้น API จะส่ง response ผลลัพธ์การดำเนินการกลับให้ Client	
Sample Response	 <pre> {   "success": true,   "message": "Product has been added to database: Pen,Shelf" } </pre> <pre> {   "storeId": 1,   "products": [     {       "productName": "Pen",       "imagePath": "penImageURL",       "imageName": "penpen",       "price": 10     },     {       "productName": "Shelf",       "imagePath": "shelfImageURL",       "imageName": "shelf",       "price": 300,       "promotion": {         "title": "ลด 20%",         "promotionPrice": 240,         "startDate": "2022-11-09 00:00:00",         "endDate": "2022-11-20 00:00:00"       }     }   ] } </pre>	

#### 4.2.10. API เพิ่มข้อมูลโปรโมชั่นของสินค้า

ตารางที่ 4.12 ตารางอธิบาย API เพิ่มข้อมูลโปรโมชั่นของสินค้า

API ID	10
API Name	เพิ่มข้อมูลโปรโมชั่นของสินค้า
Endpoint	/product/promotion/add
Variable Request	Body JSON:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่มีการเผยแพร่ ทั้งสิ้น อีกทั้งยังมีสิทธิ์เปลี่ยนแปลงเนื้อหา และต้องรับผิดชอบต่อสิ่งของเอกสารทุกครั้งที่มีการแก้ไข

	Property	Description
	productId	Id สินค้าที่เป็นเจ้าของข้อมูลโปรโมชั่น (require)
	tittle	ชื่อโปรโมชั่น (require)
	promotionPrice	ราคาหลังจัดโปรโมชั่น (require)
	startDate	วันและเวลาเริ่มต้นของโปรโมชั่น (require)
	endDate	วันและเวลาสิ้นสุดของโปรโมชั่น (require)
HTTP method	POST	
Description	รับ Variable Request ผ่าน Body JSON โดย API จะเพิ่มข้อมูลโปรโมชั่นของสินค้าที่มีอยู่ในฐานข้อมูล โดยอ้างอิงจาก productId เป็น fk มีการตรวจสอบความครบถ้วนของข้อมูลเบื้องต้นก่อนอนุญาตให้เพิ่มข้อมูลโปรโมชั่นลงในฐานข้อมูล จากนั้นส่ง response ผลลัพธ์การดำเนินการกลับให้ Client	
Sample Response		

#### 4.2.11. API แสดงรายการสินค้าทั้งหมด

ตารางที่ 4.13 ตารางอธิบาย API แสดงรายการสินค้าทั้งหมด

API ID	11
API Name	แสดงรายการสินค้าทั้งหมด
Endpoint	/product/getall
Variable Request	-
HTTP method	GET
Description	เป็น API สำหรับ Query ข้อมูลทั้งหมดใน Table products โดยรวมข้อมูลโปรโมชั่นสินค้า และมี Limit ที่ 20 records

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample Response	<pre>[   {     "productId": "1",     "storeId": "1",     "productName": "Cutter",     "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/products-a6a0-cc248911fb79",     "imageName": "0001",     "price": 20,     "promotions": {       "id": "1",       "title": "ลด 20%",       "promotionPrice": 16,       "startDate": "2022-11-09T07:00:00.000Z",       "endDate": "2022-11-20T07:00:00.000Z",       "productId": "1"     }   },   {     "productId": "3",     "storeId": "1",     "productName": "Fan",     "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/products-6fb7ef41dccc9",     "imageName": "0003",     "price": 569,     "promotions": {       "id": "2",       "title": "ลด 10%",       "promotionPrice": 512,       "startDate": "2022-11-09T07:00:00.000Z",       "endDate": "2022-11-20T07:00:00.000Z",       "productId": "3"     }   },   {     "productId": "17",     "storeId": "6",     "productName": "Cutter",     "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit-project.appspot.com/o/products-a6a0-cc248911fb79",     "imageName": "0001",     "price": 20,     "promotions": {       "id": "1",       "title": "ลด 20%",       "promotionPrice": 16,       "startDate": "2022-11-09T07:00:00.000Z",       "endDate": "2022-11-20T07:00:00.000Z",       "productId": "1"     }   } ]</pre>
-----------------	---

#### 4.2.12. API แสดงรายละเอียดสินค้า

ตารางที่ 4.14 ตารางอธิบาย API แสดงรายละเอียดสินค้า

API ID	12
API Name	แสดงรายละเอียดสินค้า
Endpoint	/product/get:id
Variable Request	Query Parameters: productId
HTTP method	GET
Description	เป็น API สำหรับแสดงรายละเอียดสินค้าที่ต้องการ โดยรับ Query Parameters เป็น productId ใช้สำหรับการ Query รายละเอียดข้อมูลสินค้าที่ต้องการ โดยใช้ productId ในการอ้างอิงกับข้อมูลสินค้าในฐานข้อมูลที่ต้องการ รวมถึงข้อมูลโปรโมชั่นของสินค้า
Sample Response	ผลการดึงข้อมูลสินค้า id ที่ 1 จาก Database <pre>{   "id": "1",   "productName": "Cutter",   "price": 20,   "promotionPrice": 16,   "startDate": "2022-11-09T07:00:00.000Z",   "endDate": "2022-11-20T07:00:00.000Z",   "productId": "1" }</pre>

	<pre> "imagePath": "https://firebasestorage.googleapis.com/v0/b/sahakit- project.appspot.com/o/product%2Fcutter.jpg?alt=media&amp;token=513be474- be02-4179-a6a0-cc248911fb79", "imageName": "0001", "price": 20, "storeId": "1", "PromoProduct": {   "id": "1",   "title": "ลด 20%",   "promotionPrice": 16,   "startDate": "2022-11-09T07:00:00.000Z",   "endDate": "2022-11-20T07:00:00.000Z",   "productId": "1" } } </pre>
--	---

#### 4.2.13. API ค้นหาสินค้าด้วยชื่อ

ตารางที่ 4.15 ตารางอธิบาย API ค้นหาสินค้าด้วยชื่อ

API ID	13
API Name	ค้นหาสินค้าด้วยชื่อ
Endpoint	/product/search:query
Variable Request	Query Parameters: query
HTTP method	GET
Description	รับ Query Parameters เป็นตัวอักษรหรือชื่อสินค้าที่ต้องการค้นหา โดย API จะทำการค้นหา ข้อมูลชื่อสินค้าที่มีส่วนประกอบของตัวอักษรที่รับมาจาก Query Parameters ซึ่งค้นหาแบบ Insensitive Case และส่งข้อมูลสินค้า ชื่อร้านค้าที่มีสินค้านั้นรวมถึงโปรโมชั่นของสินค้า ที่ค้นพบ ทั้งหมดให้กับ Client
Sample Response	ผลการค้นหาข้อมูลสินค้าที่มีชื่อประกอบไปด้วยตัวอักษร printer จาก Database <pre> {   "msg": "search results",   "results": [     {       "id": "60",       "productName": "Printer",       "imagePath":         "https://firebasestorage.googleapis.com/v0/b/sahakit-         project.appspot.com/o/product%2FPrinter_4.jpeg?alt=media&amp;token=df4867c9-         8e73-4fbe-815a-56d2621b0ec8",       "imageName": "0060",       "price": 3700,       "storeId": "20",       "PromoProduct": null,       "store": {         "storeName": "Extra INK"       }     },     {       "id": "48",       "productName": "Printer",       "imagePath":         "https://firebasestorage.googleapis.com/v0/b/sahakit-         project.appspot.com/o/product%2FPrinter_3.jpg?alt=media&amp;token=cf216874-         2cb2-48f0-ba30-a852c2beb987", </pre>

เอกสารนี้เป็นเอกสาร  
ไม่ว่ากรณีใดๆ ทั้งสิ้น


	<pre> "imageName": "0048", "price": 6990, "storeId": "16", "PromoProduct": {   "id": "6",   "title": "ลด 10%",   "promotionPrice": 6290,   "startDate": "2022-11-09T07:00:00.000Z",   "endDate": "2023-01-10T07:00:00.000Z",   "productId": "48" }, "store": {   "storeName": "24 INK" } }, {   "id": "39",   "productName": "Printer",   "imagePath":   "https://firebasestorage.googleapis.com/v0/b/sahakit-   project.appspot.com/o/product%2FPrinter_2.webp?alt=media&amp;token=2a73a63f-   3ffe-4abe-bacd-14d697e58d4f",   "imageName": "0039",   "price": 6990,   "storeId": "13",   "PromoProduct": null,   "store": {     "storeName": "Ink Save"   } }, {   "id": "33",   "productName": "Printer",   "imagePath":   "https://firebasestorage.googleapis.com/v0/b/sahakit-   project.appspot.com/o/product%2FPrinter_1.webp?alt=media&amp;token=b75d88e9-   1914-40a3-9e7b-bf34a8dd4be2",   "imageName": "0033",   "price": 8890,   "storeId": "11",   "PromoProduct": null,   "store": {     "storeName": "Clinic Printer"   } } ] </pre>
--	---

#### 4.2.14. API อัปเดตข้อมูลร้านค้า

ตารางที่ 4.16 ตารางอธิบาย API อัปเดตข้อมูลร้านค้า

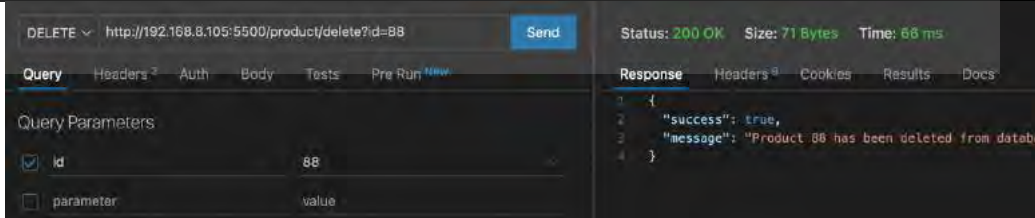
API ID	14
API Name	อัปเดตข้อมูลร้านค้า
Endpoint	/product/update:id
Variable	Query Parameters:
Request	productId Body JSON:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Property	Description
	productName	ชื่อสินค้า (require)
	price	ราคาของสินค้า (require)
HTTP method	PATCH	
Description	รับ Query Parameters เป็น productid ที่ใช้ในการอ้างอิงสินค้าที่ต้องการแก้ไข และรับ Body เป็นข้อมูลที่ต้องการแก้ไข โดย API จะตรวจสอบความถูกต้องของข้อมูลเบื้องต้นและอัปเดตข้อมูลสินค้าตามข้อมูลที่รับมาเข้าไปในฐานข้อมูลของระบบ จากนั้นส่ง response ผลลัพธ์การดำเนินการกลับให้ Client	
Sample Response		

#### 4.2.15. API ลบข้อมูลสินค้า

ตารางที่ 4.17 ตารางอธิบาย API ลบข้อมูลสินค้า

API ID	15
API Name	ลบข้อมูลสินค้า
Endpoint	/product/delete?id
Variable Request	Query Parameters: productid
HTTP method	DELETE
Description	ลบข้อมูลสินค้าใน Database โดยใช้ Query Parameters: productid ที่รับมาเป็น id ในการอ้างอิงกับข้อมูลสินค้าที่ต้องการลบ จากนั้นส่ง response ผลลัพธ์การดำเนินการกลับให้ Client
Sample Response	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3. Python Application

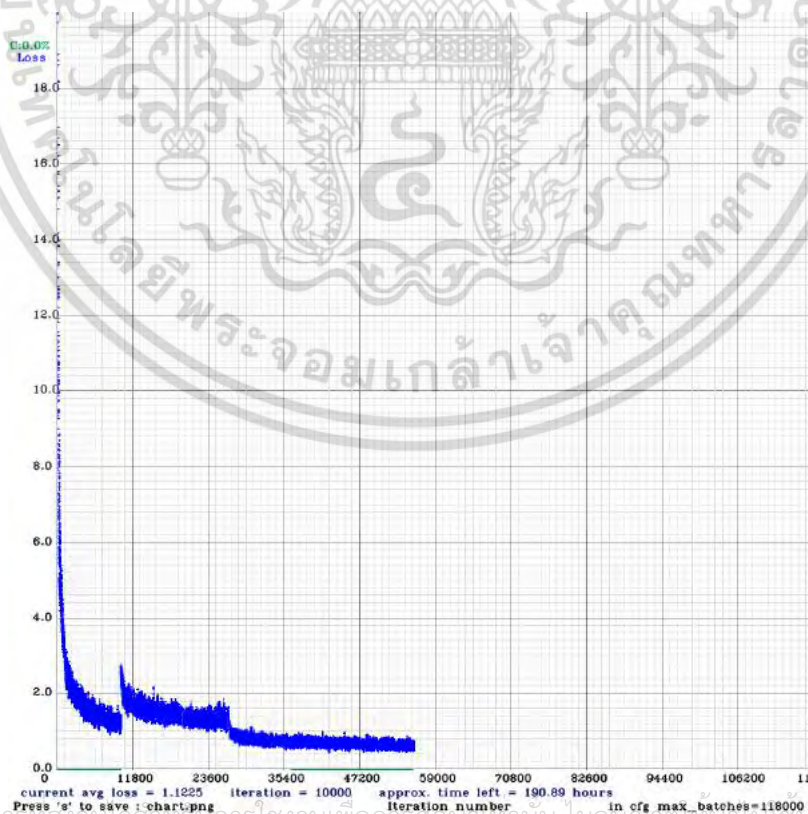
เป็นส่วนแอปพลิเคชัน Python ที่พัฒนาเพื่อใช้ในการติดต่อเรียกใช้งานการ์ด FPGA ในการประมวลผล Object Detection ผ่าน High-Level API ที่เป็น Library ใน Vitis-AI Development Platform โดยในหัวข้อนี้จะมีการอภิปรายผลการวิจัยตั้งแต่ผลการฝึกสอนโมเดล AI การแปลงโมเดล AI เป็น fixed-point model การ compile ให้เข้ากับการ์ดที่ต้องการใช้งาน ไปจนถึงการใช้งานโมเดลที่ผ่านการ compile แล้ว

#### 4.3.1. Object Detection Model

ในส่วนของการฝึกสอน model การตรวจจับวัตถุจะแบ่งเป็น 2 models เพื่อใช้ในการตรวจจับที่แตกต่างกัน โดยใช้ YOLOv4 Darknet Framework ทั้งในการฝึกสอนและการประเมินผลความแม่นยำทั้ง 2 models

##### 4.3.1.1. Model Product

เป็นโมเดล Object Detection ที่ฝึกสอนสำหรับตรวจจับรูปภาพสินค้า IT มีจำนวนคลาสสินค้าทั้งหมด 59 คลาส ฝึกสอนโดยใช้ Dataset ที่ผู้จัดทำงานวิจัยและคณะทำขึ้นมาเองทั้งหมด จำนวน 13,468 ภาพ แบ่งเป็น รูปที่ใช้ในการฝึกสอน (Train Set) 9,223 ภาพ รูปที่ใช้ในการตรวจสอบความแม่นยำ (Validation Set) 2,524 ภาพ และภาพที่ใช้ในการทดสอบ (Test Set) 1,711 ภาพ



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่นำไปเผยแพร่ประโยชน์ด้านการค้า  
ไม่ว่ากรณีรูปที่ 4.24 Plot Graph ที่ได้จากการฝึกสอน Model สินค้า โดย Plot จากค่า AVG loss นำไปใช้

ผลลัพธ์ค่าน้ำหนักโมเดลที่ได้ที่จากการฝึกสอนโดยคัดเลือกจากการประเมินผลความแม่นยำ หรือค่า Mean Average Precision (mAP) 3 อันดับแรกที่สูงที่สุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำนำหน้า Model สินค้าที่ 1 ผ่านการฝึกสอน 26800 รอบ

คำนวณ mAP และทดลองตรวจจับรูปสินค้า

```

Loading weights from cfg/yolov4_product/snapshots/yolov4-product_v2_26800.weights...
seen 64, trained: 1715 K-images (26 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

calculation mAP (mean average precision)...
Detection layer: 139 - type = 28
Detection layer: 150 - type = 28
Detection layer: 161 - type = 28
4XRT build version: 2.11.648
Build hash: 38a348510a76968a67d988128c3368f554e7b97b
Build date: 2021-07-02 15:48:32
Git branch: 2021.1
HOST: Alex-PowerEdge-R740
EXE: /home/alex/Work_Name_2022/Vitis-AI/working/darknet/darknet
2548
detections_count = 14770, unique_truth_count = 3962
class_id = 0, name = asus_notebook, ap = 70.07% (TP = 29, FP = 5)
class_id = 1, name = banknote, ap = 95.78% (TP = 62, FP = 11)
class_id = 2, name = barchair, ap = 86.51% (TP = 162, FP = 12)
class_id = 3, name = blackhead_remover, ap = 5.41% (TP = 3, FP = 3)
class_id = 4, name = bookbank, ap = 75.29% (TP = 28, FP = 5)
class_id = 5, name = bottle, ap = 62.59% (TP = 65, FP = 16)
class_id = 6, name = bread, ap = 76.66% (TP = 29, FP = 7)
class_id = 7, name = cake, ap = 73.89% (TP = 42, FP = 15)
class_id = 8, name = chair, ap = 58.06% (TP = 18, FP = 6)
class_id = 9, name = coffee, ap = 56.98% (TP = 29, FP = 17)
class_id = 10, name = coin operated games, ap = 76.63% (TP = 16, FP = 9)
class_id = 11, name = coins, ap = 89.84% (TP = 63, FP = 11)
class_id = 12, name = cutter, ap = 59.87% (TP = 79, FP = 38)
class_id = 13, name = dahua cctv, ap = 66.53% (TP = 28, FP = 3)
class_id = 14, name = doll, ap = 24.73% (TP = 8, FP = 21)
class_id = 15, name = earpods, ap = 59.90% (TP = 27, FP = 23)
class_id = 16, name = elbow, ap = 31.84% (TP = 26, FP = 38)
class_id = 17, name = external hdd, ap = 83.47% (TP = 36, FP = 5)
class_id = 18, name = fan, ap = 83.97% (TP = 101, FP = 24)
class_id = 19, name = fish drive, ap = 59.67% (TP = 14, FP = 4)
class_id = 20, name = folio case, ap = 26.37% (TP = 8, FP = 17)
class_id = 21, name = food container, ap = 36.79% (TP = 23, FP = 9)
class_id = 22, name = game machine, ap = 64.99% (TP = 99, FP = 74)
class_id = 23, name = hand cream, ap = 90.34% (TP = 29, FP = 16)
class_id = 24, name = hiview cctv, ap = 96.24% (TP = 77, FP = 6)
class_id = 25, name = ink cartridge, ap = 52.69% (TP = 6, FP = 4)
class_id = 26, name = ink toner, ap = 70.06% (TP = 11, FP = 1)
class_id = 27, name = ipad pro11, ap = 94.23% (TP = 273, FP = 23)
class_id = 28, name = iphone 13mini, ap = 91.53% (TP = 54, FP = 1)
class_id = 29, name = iphone 14promax, ap = 96.02% (TP = 51, FP = 4)
class_id = 30, name = knee, ap = 31.29% (TP = 19, FP = 32)
class_id = 31, name = lan cable, ap = 7.93% (TP = 1, FP = 1)
class_id = 32, name = lightning cable, ap = 69.96% (TP = 21, FP = 4)
class_id = 33, name = macbook13, ap = 92.98% (TP = 17, FP = 2)
class_id = 34, name = macbook16, ap = 90.58% (TP = 43, FP = 3)
class_id = 35, name = mask, ap = 56.97% (TP = 61, FP = 23)
class_id = 36, name = microsoft.office, ap = 22.92% (TP = 3, FP = 2)
class_id = 37, name = microusb cable, ap = 0.00% (TP = 0, FP = 0)
class_id = 38, name = mobile case, ap = 39.85% (TP = 75, FP = 58)
class_id = 39, name = mobile film, ap = 51.84% (TP = 16, FP = 13)
class_id = 40, name = monitor, ap = 62.78% (TP = 36, FP = 8)
class_id = 41, name = mouse, ap = 82.77% (TP = 42, FP = 6)
class_id = 42, name = pocketwifi, ap = 41.13% (TP = 29, FP = 28)
class_id = 43, name = power bank, ap = 43.51% (TP = 13, FP = 6)
class_id = 44, name = power cord, ap = 58.21% (TP = 27, FP = 7)
class_id = 45, name = printer, ap = 73.55% (TP = 18, FP = 1)
class_id = 46, name = ram_ddr3, ap = 85.67% (TP = 23, FP = 3)
class_id = 47, name = range_extender, ap = 30.50% (TP = 16, FP = 7)
class_id = 48, name = scissors, ap = 59.64% (TP = 66, FP = 42)
class_id = 49, name = sony cctv, ap = 71.23% (TP = 17, FP = 5)
class_id = 50, name = speaker, ap = 69.54% (TP = 118, FP = 33)
class_id = 51, name = table, ap = 83.21% (TP = 37, FP = 7)
class_id = 52, name = tablet case, ap = 73.06% (TP = 48, FP = 15)
class_id = 53, name = tissue, ap = 37.63% (TP = 17, FP = 3)
class_id = 54, name = true_box, ap = 71.11% (TP = 28, FP = 9)
class_id = 55, name = twisted_ankle, ap = 75.56% (TP = 26, FP = 9)
class_id = 56, name = type_c_cable, ap = 18.39% (TP = 2, FP = 2)
class_id = 57, name = windows10, ap = 47.37% (TP = 5, FP = 2)
class_id = 58, name = zfold4, ap = 91.96% (TP = 41, FP = 3)

for conf_thresh = 0.25, precision = 0.77, recall = 0.59, F1-score = 0.67
for conf_thresh = 0.25, TP = 2335, FP = 714, FN = 1627, average IoU = 57.40 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.599240, or 59.92 %
Total Detection Time: 74 Seconds

```

รูปที่ 4.25 รูปการประเมินผลค่าความแม่นยำคำนำหน้า Model สินค้าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.18 ตารางรูปผลลัพธ์การตรวจจับสินค้าด้วยค่าน้ำหนัก Model สินค้าที่ 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำนำหน้า Model สินค้าที่ 2 ผ่านการฝึกสอน 30000 รอบ

คำนวณ mAP และทดลองตรวจจับรูปสินค้า

```

Loading weights from cfg/yolov4_product/snapshots/yolov4-product_v2_30000.weights...
seen 64, trained: 1920 K-images (30 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

calculation mAP (mean average precision)...
Detection layer: 139 - type = 28
Detection layer: 150 - type = 28
Detection layer: 161 - type = 28
4XRT build version: 2.11.648
Build hash: 38a348510a76968a67d988128c3368f554e7b97b
Build date: 2021-07-02 15:48:32
Git branch: 2021.1
HOST: Alex-PowerEdge-R740
EXE: /home/alex/Work_Name_2022/Vitis-AI/working/darknet/darknet
2548

detections_count = 7269, unique_truth_count = 3962
class_id = 0, name = asus_notebook, ap = 79.85% (TP = 22, FP = 4)
class_id = 1, name = banknote, ap = 89.53% (TP = 47, FP = 2)
class_id = 2, name = barchair, ap = 87.39% (TP = 170, FP = 12)
class_id = 3, name = blackhead_remover, ap = 2.48% (TP = 2, FP = 9)
class_id = 4, name = bookbank, ap = 28.09% (TP = 10, FP = 15)
class_id = 5, name = bottle, ap = 59.70% (TP = 54, FP = 10)
class_id = 6, name = bread, ap = 21.51% (TP = 19, FP = 4)
class_id = 7, name = cake, ap = 87.64% (TP = 45, FP = 3)
class_id = 8, name = chair, ap = 82.55% (TP = 21, FP = 2)
class_id = 9, name = coffee, ap = 43.69% (TP = 23, FP = 8)
class_id = 10, name = coin operated games, ap = 75.85% (TP = 17, FP = 7)
class_id = 11, name = coins, ap = 92.99% (TP = 67, FP = 4)
class_id = 12, name = cutter, ap = 59.60% (TP = 74, FP = 15)
class_id = 13, name = dahua CCTV, ap = 65.09% (TP = 20, FP = 6)
class_id = 14, name = doll, ap = 22.79% (TP = 9, FP = 10)
class_id = 15, name = earpods, ap = 83.40% (TP = 36, FP = 6)
class_id = 16, name = elbow, ap = 33.59% (TP = 26, FP = 22)
class_id = 17, name = external_hdd, ap = 82.86% (TP = 35, FP = 4)
class_id = 18, name = fan, ap = 96.87% (TP = 100, FP = 7)
class_id = 19, name = fisah_drive, ap = 79.81% (TP = 17, FP = 0)
class_id = 20, name = folio_case, ap = 35.36% (TP = 9, FP = 7)
class_id = 21, name = food_container, ap = 38.50% (TP = 20, FP = 5)
class_id = 22, name = game_machina, ap = 78.09% (TP = 96, FP = 31)
class_id = 23, name = hand_cream, ap = 36.70% (TP = 26, FP = 15)
class_id = 24, name = hiview CCTV, ap = 99.64% (TP = 81, FP = 1)
class_id = 25, name = ink_cartridge, ap = 35.99% (TP = 5, FP = 4)
class_id = 26, name = ink_toner, ap = 68.92% (TP = 11, FP = 2)
class_id = 27, name = ipad_pro11, ap = 92.04% (TP = 258, FP = 10)
class_id = 28, name = iphone_13mini, ap = 90.00% (TP = 54, FP = 0)
class_id = 29, name = iphone_14promax, ap = 96.30% (TP = 52, FP = 1)
class_id = 30, name = knee, ap = 26.21% (TP = 19, FP = 17)
class_id = 31, name = lan_cable, ap = 12.15% (TP = 2, FP = 3)
class_id = 32, name = lightning_cable, ap = 61.02% (TP = 24, FP = 9)
class_id = 33, name = macbook13, ap = 100.00% (TP = 24, FP = 0)
class_id = 34, name = macbook15, ap = 92.00% (TP = 44, FP = 0)
class_id = 35, name = mask, ap = 50.81% (TP = 36, FP = 8)
class_id = 36, name = microsoft_office, ap = 16.67% (TP = 2, FP = 3)
class_id = 37, name = microusb_cable, ap = 0.00% (TP = 0, FP = 0)
class_id = 38, name = mobile_case, ap = 48.03% (TP = 63, FP = 36)
class_id = 39, name = mobile_film, ap = 68.62% (TP = 22, FP = 5)
class_id = 40, name = monitor, ap = 64.55% (TP = 36, FP = 9)
class_id = 41, name = mouse, ap = 85.28% (TP = 47, FP = 3)
class_id = 42, name = pocketwifi, ap = 24.91% (TP = 20, FP = 10)
class_id = 43, name = power_bank, ap = 50.26% (TP = 20, FP = 4)
class_id = 44, name = power_cord, ap = 71.89% (TP = 46, FP = 12)
class_id = 45, name = printer, ap = 67.69% (TP = 20, FP = 4)
class_id = 46, name = ram_ddr3, ap = 92.05% (TP = 25, FP = 1)
class_id = 47, name = range_extender, ap = 58.13% (TP = 28, FP = 8)
class_id = 48, name = scissors, ap = 63.53% (TP = 51, FP = 25)
class_id = 49, name = sony CCTV, ap = 76.11% (TP = 16, FP = 6)
class_id = 50, name = speaker, ap = 61.79% (TP = 97, FP = 25)
class_id = 51, name = table, ap = 87.52% (TP = 38, FP = 7)
class_id = 52, name = tablet_case, ap = 75.39% (TP = 49, FP = 17)
class_id = 53, name = tissue, ap = 31.72% (TP = 17, FP = 1)
class_id = 54, name = true_box, ap = 57.97% (TP = 15, FP = 1)
class_id = 55, name = twisted_ankle, ap = 66.12% (TP = 26, FP = 8)
class_id = 56, name = type_c_cable, ap = 21.85% (TP = 5, FP = 6)
class_id = 57, name = windows10, ap = 51.11% (TP = 7, FP = 2)
class_id = 58, name = zfold4, ap = 92.53% (TP = 42, FP = 2)

for conf_thresh = 0.25, precision = 0.83, recall = 0.57, F1-score = 0.68
for conf_thresh = 0.25, TP = 2278, FP = 457, FN = 1684, average IoU = 64.91 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.612367, or 61.24 %
Total Detection Time: 76 Seconds

```

รูปที่ 4.26 รูปการประเมินผลค่าความแม่นยำคำนำหน้า Model สินค้าที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.19 ตารางรูปผลลัพธ์การตรวจจับสินค้าด้วยค่าน้ำหนัก Model สินค้าที่ 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำนำหน้า Model สินค้าที่ 3 ผ่านการฝึกสอน 50000 รอบ

คำนวณ mAP และทดลองตรวจจับรูปสินค้า

```

Loading weights from cfg/yolov4_product/snapshots/yolov4-product_v2_50000.weights...
seen 64, trained: 3200 K-images (50 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

calculation mAP (mean average precision)...
Detection layer: 139 - type = 28
Detection layer: 150 - type = 28
Detection layer: 161 - type = 28
4XRT build version: 2.11.648
Build hash: 38a348510a76968a67d988128c3368f554e7b97b
Build date: 2021-07-02 15:48:32
Git branch: 2021.1
HOST: Alex-PowerEdge-R740
EXE: /home/alex/Work_Name_2022/Vitis-AI/working/darknet/darknet
2548
detections_count = 6108, unique_truth_count = 3962
class_id = 0, name = asus_notebook, ap = 77.05% (TP = 19, FP = 3)
class_id = 1, name = banknote, ap = 78.38% (TP = 41, FP = 3)
class_id = 2, name = barchair, ap = 90.90% (TP = 174, FP = 19)
class_id = 3, name = blackhead_remover, ap = 2.14% (TP = 1, FP = 6)
class_id = 4, name = bookbank, ap = 20.09% (TP = 12, FP = 16)
class_id = 5, name = bottle, ap = 56.74% (TP = 58, FP = 9)
class_id = 6, name = bread, ap = 17.95% (TP = 29, FP = 7)
class_id = 7, name = cake, ap = 81.96% (TP = 43, FP = 5)
class_id = 8, name = chair, ap = 83.64% (TP = 29, FP = 1)
class_id = 9, name = coffee, ap = 47.86% (TP = 23, FP = 10)
class_id = 10, name = coin_operated_games, ap = 66.34% (TP = 11, FP = 4)
class_id = 11, name = coins, ap = 93.69% (TP = 65, FP = 2)
class_id = 12, name = cutter, ap = 62.26% (TP = 73, FP = 14)
class_id = 13, name = dahua_cctv, ap = 64.32% (TP = 21, FP = 7)
class_id = 14, name = doll, ap = 22.22% (TP = 9, FP = 21)
class_id = 15, name = earpods, ap = 83.67% (TP = 36, FP = 7)
class_id = 16, name = elbow, ap = 29.63% (TP = 27, FP = 29)
class_id = 17, name = external_hdd, ap = 80.33% (TP = 36, FP = 6)
class_id = 18, name = fan, ap = 94.85% (TP = 96, FP = 6)
class_id = 19, name = fisah_drive, ap = 79.72% (TP = 16, FP = 0)
class_id = 20, name = folio_case, ap = 18.81% (TP = 4, FP = 7)
class_id = 21, name = food_container, ap = 36.96% (TP = 20, FP = 5)
class_id = 22, name = game_machine, ap = 61.44% (TP = 78, FP = 28)
class_id = 23, name = hand_cream, ap = 58.42% (TP = 32, FP = 9)
class_id = 24, name = hiview_cctv, ap = 98.10% (TP = 84, FP = 3)
class_id = 25, name = ink_cartridge, ap = 39.86% (TP = 5, FP = 4)
class_id = 26, name = ink_toner, ap = 70.70% (TP = 11, FP = 2)
class_id = 27, name = ipad_pro11, ap = 95.36% (TP = 286, FP = 4)
class_id = 28, name = iphone_13mini, ap = 89.97% (TP = 53, FP = 1)
class_id = 29, name = iphone_14promax, ap = 96.30% (TP = 52, FP = 2)
class_id = 30, name = knee, ap = 39.98% (TP = 29, FP = 12)
class_id = 31, name = lan_cable, ap = 12.71% (TP = 3, FP = 3)
class_id = 32, name = lightning_cable, ap = 70.82% (TP = 28, FP = 6)
class_id = 33, name = macbook13, ap = 100.00% (TP = 24, FP = 0)
class_id = 34, name = macbook15, ap = 91.57% (TP = 45, FP = 2)
class_id = 35, name = mask, ap = 60.63% (TP = 62, FP = 8)
class_id = 36, name = microsoft_office, ap = 16.67% (TP = 2, FP = 1)
class_id = 37, name = microusb_cable, ap = 0.00% (TP = 0, FP = 0)
class_id = 38, name = mobile_case, ap = 45.24% (TP = 57, FP = 32)
class_id = 39, name = mobile_film, ap = 73.66% (TP = 22, FP = 3)
class_id = 40, name = monitor, ap = 59.89% (TP = 31, FP = 4)
class_id = 41, name = mouse, ap = 83.97% (TP = 41, FP = 1)
class_id = 42, name = pocketwifi, ap = 37.28% (TP = 21, FP = 12)
class_id = 43, name = power_bank, ap = 47.15% (TP = 26, FP = 3)
class_id = 44, name = power_cord, ap = 70.38% (TP = 47, FP = 14)
class_id = 45, name = printer, ap = 70.49% (TP = 21, FP = 6)
class_id = 46, name = ram_ddr3, ap = 92.74% (TP = 26, FP = 1)
class_id = 47, name = range_extender, ap = 64.96% (TP = 33, FP = 9)
class_id = 48, name = scissors, ap = 52.32% (TP = 39, FP = 19)
class_id = 49, name = sony_cctv, ap = 76.06% (TP = 17, FP = 8)
class_id = 50, name = speaker, ap = 53.25% (TP = 98, FP = 22)
class_id = 51, name = table, ap = 90.36% (TP = 39, FP = 6)
class_id = 52, name = tablet_case, ap = 72.68% (TP = 49, FP = 15)
class_id = 53, name = tissue, ap = 32.06% (TP = 18, FP = 4)
class_id = 54, name = true_box, ap = 63.19% (TP = 18, FP = 2)
class_id = 55, name = twisted_ankle, ap = 67.53% (TP = 26, FP = 8)
class_id = 56, name = type_c_cable, ap = 17.87% (TP = 5, FP = 5)
class_id = 57, name = windows10, ap = 46.61% (TP = 6, FP = 8)
class_id = 58, name = zfold4, ap = 95.10% (TP = 43, FP = 2)

for conf_thresh = 0.25, precision = 0.84, recall = 0.58, F1-score = 0.69
for conf_thresh = 0.25, TP = 2291, FP = 421, FN = 1671, average IoU = 67.35 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.609457, or 60.95 %
Total Detection Time: 87 Seconds
  
```

รูปที่ 4.27 รูปการประเมินผลค่าความแม่นยำคำนำหน้า Model สินค้าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.20 ตารางรูปผลลัพธ์การตรวจจับสินค้าด้วยค่าน้ำหนัก Model สินค้าที่ 3



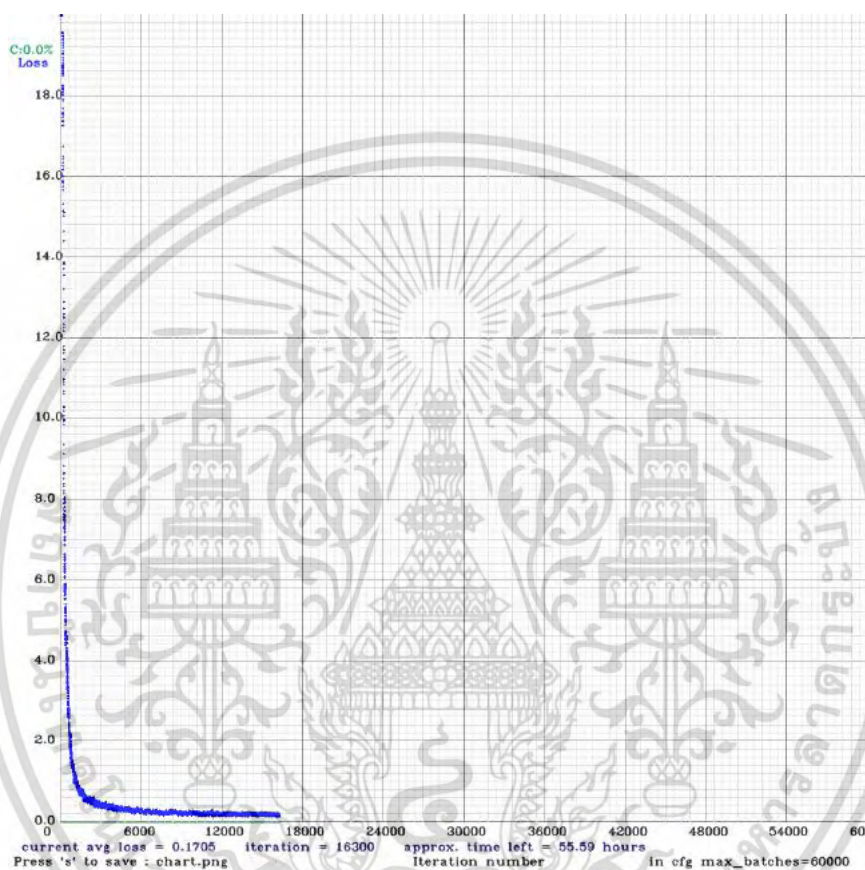
จะสังเกตเห็นได้ว่า จากรูปผลลัพธ์การประเมินผลความแม่นยำทั้ง 3 Model ค่าน้ำหนัก Model ที่มีค่า Mean Average Precision (mAP) มากที่สุดคือ ค่าน้ำหนัก Model ที่ 2 ซึ่งมีค่า mAP อยู่ที่ 61.24 โดยใช้จำนวนรอบในการฝึกสอนทั้งหมด 30,000 รอบ

จึงทำให้ผู้จัดทำงานวิจัยได้เลือกค่าน้ำหนัก Model ที่ 2 ที่มีค่า mAP มากที่สุดสำหรับการตรวจจับสินค้าในระบบนี้ และนำไปดำเนินการแปลง Model ให้สามารถทำงานบน DPU ได้ในขั้นตอนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.3.1.2. Model Store

เป็นโมเดล Object Detection ที่ฝึกสอนสำหรับตรวจจับรูปภาพหน้าร้านค้า มีจำนวนคลาสหน้าร้านค้าทั้งหมด 30 คลาส ฝึกสอนโดยใช้ Dataset ที่ผู้จัดทำงานวิจัยและคณะทำขึ้นมาเองทั้งหมดจำนวน 9,240 ภาพ แบ่งเป็น รูปที่ใช้ในการฝึกสอน (Train Set) 6,460 ภาพ รูปที่ใช้ในการตรวจสอบความแม่นยำ (Validation Set) 1,841 ภาพ และภาพที่ใช้ในการทดสอบ (Test Set) 939 ภาพ



รูปที่ 4.28 Plot Graph ที่ได้จากการฝึกสอน Model ร้านค้า โดย Plot จากค่า AVG loss

ผลลัพธ์ค่าน้ำหนักโมเดลที่ได้ที่จากการฝึกสอนโดยคัดเลือกจากการประเมินผลความแม่นยำ หรือค่า Mean Average Precision (mAP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Loading weights from cfg/yolov4_store/snapshots/yolov4-store_10000.weights...
seen 64, trained: 640 K-images (10 Kilo-batches_64)
Done! Loaded 162 layers from weights-file

calculation mAP (mean average precision)...
Detection layer: 139 - type = 28
Detection layer: 150 - type = 28
Detection layer: 161 - type = 28
4XRT build version: 2.11.648
Build hash: 38a348510a76068a67d988128c3368f554e7b97b
Build date: 2021-07-02 15:48:32
Git branch: 2021.1
HOST: Alex-PowerEdge-R740
EXE: /home/alex/Work_Name_2022/Vitis-AI/working/darknet/darknet
1844
detections_count = 3894, unique_truth_count = 2352
class_id = 0, name = 10_k2phone, ap = 98.98% (TP = 82, FP = 2)
class_id = 1, name = 11_clinicprinter, ap = 100.00% (TP = 60, FP = 0)
class_id = 2, name = 12_wk, ap = 36.52% (TP = 13, FP = 17)
class_id = 3, name = 13_inksave, ap = 92.88% (TP = 46, FP = 5)
class_id = 4, name = 14_powerit, ap = 88.39% (TP = 39, FP = 3)
class_id = 5, name = 15_kp, ap = 90.24% (TP = 37, FP = 0)
class_id = 6, name = 16_24link, ap = 92.40% (TP = 39, FP = 2)
class_id = 7, name = 17_mezzo, ap = 86.72% (TP = 136, FP = 21)
class_id = 8, name = 18_mz, ap = 73.38% (TP = 97, FP = 25)
class_id = 9, name = 19_nsp, ap = 90.40% (TP = 93, FP = 0)
class_id = 10, name = 1_thankyou, ap = 99.58% (TP = 84, FP = 0)
class_id = 11, name = 20_extraink, ap = 100.00% (TP = 76, FP = 0)
class_id = 12, name = 21_cliniccom, ap = 100.00% (TP = 44, FP = 0)
class_id = 13, name = 22_yongxin, ap = 89.13% (TP = 72, FP = 2)
class_id = 14, name = 23_echo, ap = 100.00% (TP = 43, FP = 0)
class_id = 15, name = 24_vida, ap = 96.53% (TP = 57, FP = 1)
class_id = 16, name = 25_iccomputer, ap = 78.05% (TP = 30, FP = 5)
class_id = 17, name = 26_itcity, ap = 98.93% (TP = 147, FP = 1)
class_id = 18, name = 2701_horit, ap = 100.00% (TP = 13, FP = 0)
class_id = 19, name = 2702_horit, ap = 96.11% (TP = 76, FP = 0)
class_id = 20, name = 2703_horit, ap = 97.60% (TP = 55, FP = 4)
class_id = 21, name = 2_mrdiy, ap = 100.00% (TP = 53, FP = 0)
class_id = 22, name = 3_zakin, ap = 100.00% (TP = 78, FP = 0)
class_id = 23, name = 4_ghbank, ap = 80.35% (TP = 58, FP = 1)
class_id = 24, name = 5_ngmobile, ap = 26.15% (TP = 12, FP = 41)
class_id = 25, name = 6_asus, ap = 76.44% (TP = 27, FP = 0)
class_id = 26, name = 7_jib, ap = 93.93% (TP = 147, FP = 3)
class_id = 27, name = 8_true, ap = 93.59% (TP = 80, FP = 20)
class_id = 28, name = 91_banana, ap = 99.49% (TP = 82, FP = 9)
class_id = 29, name = 92_banana, ap = 100.00% (TP = 74, FP = 0)

for conf_thresh = 0.25, precision = 0.92, recall = 0.83, F1-score = 0.87
for conf_thresh = 0.25, TP = 1950, FP = 162, FN = 402, average IoU = 73.90 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.891930, or 89.19 %
Total Detection Time: 47 Seconds

```

รูปที่ 4.29 รูปการประเมินผลค่าความแม่นยำค่าน้ำหนัก Model ร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า MR.D.I.Y.



รูปที่ 4.31 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า Mezzo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า VID A



รูปที่ 4.33 ผลลัพธ์การตรวจจับโลโก้หน้าร้านค้า BANANA

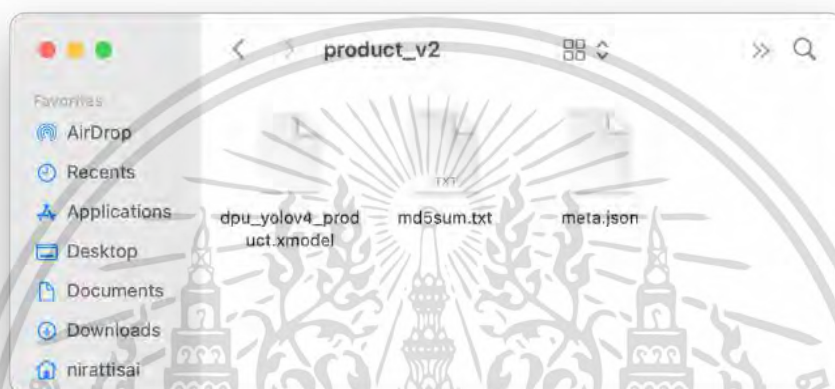
จากรูปผลลัพธ์การประเมินผลความแม่นยำ ค่าน้ำหนักของ Model ร้านค้ามีค่า mAP อยู่ที่ 89.19 ซึ่งให้ผลลัพธ์ในการตรวจจับรูปภาพร้านค้าที่ดี โดยใช้จำนวนรอบในการฝึกสอนทั้งหมด 10,000 รอบ

จึงทำให้ผู้จัดทำงานวิจัย หยุดการฝึกสอนไว้ที่ 10,000 รอบ และนำ Model สิ้นค่านั้นไปดำเนินการแลกเปลี่ยน Model ให้สามารถทำงานบน DPU ได้ในขั้นต่อไป

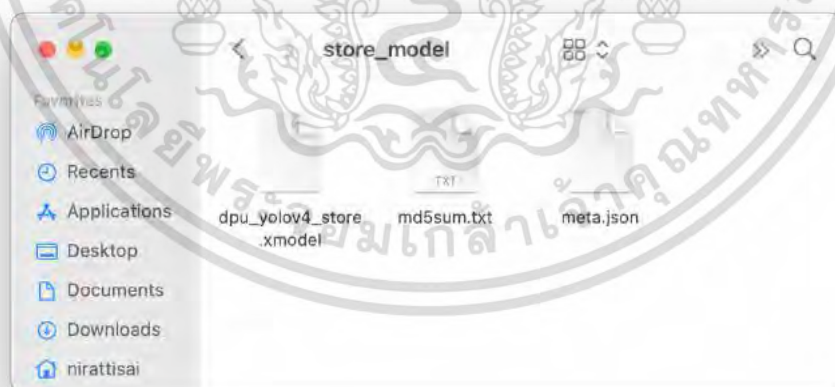
## 4.3.2. FPGA Process

### 4.3.2.1. Xmodel

เมื่อคัดเลือก Model ที่มีผลลัพธ์ที่พึงพอใจได้แล้ว ขั้นตอนต่อไปคือการแปลง Model ให้สามารถทำงานบน DPU ของ Xilinx ได้ โดยใช้ Vitis-AI Development Platform ในการแปลง Model เมื่อแปลง Model เรียบร้อยแล้วจะได้ไฟล์ที่มีนามสกุลไฟล์ .xmodel และสามารถนำไปใช้งานด้วยโค้ดไพธอนได้



รูปที่ 4.34 ไฟล์ xmodel ของ Model สินค้า



รูปที่ 4.35 ไฟล์ xmodel ของ Model ร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.3.2.2. DPU inference Speed test

ทดสอบความเร็วการประมวลผลการตรวจจับรูปภาพด้วย DPU ผ่านโมเดล YOLOv4 ที่เทรนด้วยชุดข้อมูล MS COCO 2017 และมีขนาด input size ขนาด 512x512 โดยนำภาพที่ไม่ซ้ำกัน 360 ภาพให้ โมเดลที่ผ่านการแปลงให้สามารถทำงาน กับได้แล้ว DPU ประมวลผลและจับเวลาตั้งแต่เริ่มประมวลผลภาพแรกจนถึงภาพสุดท้าย ผลลัพธ์ที่ได้เป็นดังนี้

```
(vitis-ai-tensorflow) Vitis-AI /workspace/Development/yolov4_speedtest > bash run_speedtest.sh 1
create dpu runner
img processing..
thread: 1
cnt: 360
FPS=42.58, total frames = 360.00 , time=8.454655 seconds
```

รูปที่ 4.36 ผลลัพธ์การทดสอบความเร็วการประมวลผลด้วย DPU

ผลลัพธ์ที่ได้มีความเร็วในการประมวลผลที่น่าพึงพอใจโดยใช้เวลาในการประมวลผลทั้งหมด 8.5 วินาทีโดยประมาณ เฉลี่ยแล้วมีความสามารถในการประมวลรูปภาพ 42.5 ภาพ ต่อหนึ่งวินาที ซึ่งในการทดลองนี้ใช้ Thread เพียง 1 Thread เท่านั้น หากทำงานแบบ Multiple thread จะสามารถทำงานได้เร็วขึ้นกว่าเดิม 83.33 % โดยเทียบจากข้อมูลประสิทธิภาพการทำงานของ Alveo U50LV ของ Xilinx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.3. Python Application

#### 4.3.3.1. App.py

เป็น Application หลักที่ทำหน้าที่ในการคอยตรวจเช็ครูปภาพจากแหล่ง Folder ที่กำหนด โดยจะมีการถ่วงเวลาไว้ชะลอรูปภาพจาก Backend เนื่องจากการประมวลผลด้วย DPU มีความสามารถในการประมวลผลพร้อมกันได้สูงสุด 5 ภาพในหนึ่งรอบจึงจำเป็นต้องมีการชะลอเพื่อการทำงานแบบเต็มประสิทธิภาพของ DPU โดย App.py จะมีการคัดเลือกรูปภาพด้วย โดยเรียงตามเวลาจากเก่าที่สุดถึงใหม่ที่สุด ด้วย Function การเรียงรูปภาพด้วยเวลา ในรูปที่ 4.37 เมื่อ App.py ตรวจพบรูปภาพใน Folder ประเภทการตรวจจับที่กำหนด App.py จะส่งรูปภาพทั้งหมดที่ผ่านการคัดเลือกไม่เกิน 5 ภาพให้ dpu\_runner.py ประมวลผลในขั้นตอนต่อไป ในรูปที่ 4.38 เมื่อประมวลเรียบร้อยแล้ว App.py จะทำการลบไฟล์รูปภาพที่ผ่านการประมวลผลแล้วเพื่อไม่ให้มีการเก็บข้อมูลรูปภาพใดๆ ของ Client และเพื่อไม่ให้เปลืองพื้นที่ Storage ของ Server ในรูปที่ 4.39

```

32 # Return source folder files but from old to last
33 def get_time_sorted_files(type):
34     time_sorted_files = []
35     folder = folders[type]
36     for file_name in fileutils.get_image_names(folder):
37         if file_name != 'processed' and file_name != 'Thumbs.db' and file_name.find('.lock') is -1:
38             try:
39                 time_sorted_files.append(
40                     File.File(
41                         type,
42                         folder + file_name,
43                         file_name,
44                         None,
45                         None,
46                         None,
47                         None,
48                         None,
49                         None,
50                         None,
51                         fileutils.get_file_mtime(folder + file_name)
52                     )
53             )
54         except Exception as e:
55             print(e)
56             print(file_name + ' file is already taken.')
57
58     time_sorted_files.sort(key=lambda x: x.getmtime, reverse=False)
59     return time_sorted_files

```

รูปที่ 4.37 Function การเรียงรูปภาพด้วยเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

88 # Create image objects
89 for sorted_file in sorted_files:
90
91     # File locking check and locking
92     if not os.path.isfile(sorted_file.file_path + '.lock'):
93         try:
94             # Limited amount of files
95             taken_files_count = taken_files_count + 1
96             if taken_files_count > max_files_to_take:
97                 break
98
99             # Lock file and process
100             open(sorted_file.file_path + '.lock', 'a').close()
101             # Append for processing
102             gm_time = fileutils.get_file_create_time(sorted_file.file_path)
103             file = File.File(
104                 sorted_file.type,
105                 sorted_file.file_path,
106                 sorted_file.file_name,
107                 fileutils.get_file_extension(sorted_file.file_path),
108                 fileutils.get_file_create_year(gm_time),
109                 fileutils.get_file_create_month(gm_time),
110                 fileutils.get_file_create_day(gm_time),
111                 fileutils.get_file_create_hour(gm_time, time_offset_hours),
112                 fileutils.get_file_create_minute(gm_time),
113                 fileutils.get_file_create_second(gm_time),
114                 None
115             )
116             image_file_objects.append(file)
117         except FileNotFoundError as e:
118             print(e)
119             pass
120     if(taken_files_count > 0):
121         if(type == 0):
122             dpu_runner.predict(image_file_objects, os.getcwd() + 'store_model/dpu_yolov4_store.xmodel',
123                               os.getcwd() + '/yolov4/configs/store.txt', 416, 0.5, os.getcwd() + '/processed/store/')
124         elif (type == 1):
125             dpu_runner.predict(image_file_objects, os.getcwd() + 'product_model/dpu_yolov4_product.xmodel',
126                               os.getcwd() + '/yolov4/configs/product.txt', 416, 0.5, os.getcwd() + '/processed/product/')
127

```

รูปที่ 4.38 Function เรียกใช้งาน dpu\_runner

```

128 # Post Process
129 for image_object in image_file_objects:
130     try:
131         # Also runs all sub processes
132         print(image_object.file_name)
133         # object_detection.analyze_image(
134         #     image_object,
135         #     move_to_processed,
136         # )
137     except AttributeError as a:
138         print('Invalid image file or storage may be run out', a)
139         # try remove the file
140         os.remove(image_object.file_path)
141     except Exception as e:
142         print(e)
143     finally:
144         # Finally remove lock file so those don't pile up
145         remove_file_lock(image_object.file_path + '.lock')
146

```

เอก

รศ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เปิดเผยเนื้อหาและต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.3.3.2. dpu\_runner.py

เป็น Application ที่ใช้ติดต่อการเครื่องมือของ Xilinx สำหรับส่วนประสานงานการทำงานกับ DPU โดย dpu\_runner.py จะทำหน้าที่ สร้าง Runner หรือ runtime API เพื่อใช้สำหรับการประมวลผลด้วย DPU โดยจะต้องส่ง Model ที่ผ่านการแปลงให้เข้ากับ DPU แล้วเข้าไปใช้งาน และเตรียมรูปภาพเบื้องต้นให้รูปภาพอยู่ในขนาดที่ input size ของ Model ต้องการ เมื่อสร้าง Runner และเตรียมรูปภาพเรียบร้อยแล้ว dpu\_runner.py ก็จะมีรูปภาพที่ผ่านการเตรียมพร้อมแล้วไปให้ DPU ประมวลผล และนำผลลัพธ์ที่ได้มาถอดรหัสตาม Algorithm ของ YOLOv4 architecture จากนั้นนำผลลัพธ์ที่ถอดรหัสแล้วมาบันทึกไว้ใน TXT ไฟล์ ในโฟลเดอร์ตามประเภทการตรวจจับวัตถุโดยสร้างชื่อไฟล์ตาม ID ที่ได้รับมาตามชื่อรูปภาพ

```

153 def predict(image_objects, model_path, class_path, input_shape, conf_threshold, output_path):
154
155     class_names = get_classes(class_path)
156     anchors_path = os.getcwd() + '/yolov4/configs/yolov4-leaky_anchors.txt'
157     anchors = get_anchors(anchors_path)
158     assert len(anchors) > 0, 'no anchors'
159     model_input_shape = (input_shape, input_shape)
160     img_pre = []
161     img_ori = []
162
163     g = xir.Graph.deserialize(model_path)
164     subgraphs = get_child_subgraph_dpu(g)
165     assert len(subgraphs) == 1 # only one DPU kernel
166
167     """ create dpu runner """
168     runner = vart.Runner.create_runner(subgraphs[0], "run")
169
170     """ pre-process image list to be run """
171     for image_object in image_objects:
172         print(image_object.file_path)
173         image = cv2.imread(image_object.file_path)
174         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
175         img_ori.append({'image': image, 'file_path': image_object.file_path,
176                       'file_name': image_object.file_name})
177         img_pre.append(preprocess_image(image, model_input_shape))
178
179     """ run with batch """
180     cnt = len(image_objects)
181     runInference(runner, img_pre, img_ori, cnt, anchors,
182                 model_input_shape, class_names, conf_threshold, output_path)
183
184     del runner

```

รูปที่ 4.40 Function predict

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

37 def preprocess_image(image, model_input_shape, is_letterbox=False):
38     """
39     Prepare model input image data with letterbox
40     resize, normalize and dim expansion
41
42     # Arguments
43     image: origin input image
44     PIL Image object containing image data
45     model_input_shape: model input image shape
46     tuple of format (height, width).
47
48     # Returns
49     image_data: numpy array of image data for model input.
50     """
51     image_data = cv2.resize(image, model_input_shape[::-1], cv2.INTER_AREA)
52     image_data = np.expand_dims(image_data, 0)
53     image_data = image_data.astype('float32') / 255.
54     return image_data

```

รูปที่ 4.41 Function Pre-Process Image

```

72 while count < cnt:
73     runSize = input_ndim[0] # batch size 5
74     """prepare batch input/output """
75     inputData = [np.empty(input_ndim, dtype='f', order="C")]
76     outputs = [np.empty(output.dims, dtype='f')]
77     for output in outputTensors]
78
79     """init input image to input buffer """
80     for j in range(runSize):
81         imageRun = inputData[0]
82         imageRun[j, ...] = img_pre[(count + j) % n_of_images]
83
84     """run with batch """
85     job_id = runner.execute_async(inputData, outputs)
86     runner.wait(job_id)
87

```

รูปที่ 4.42 Function Run Inference (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

88     """decoding ouput """
89     for j in range(cnt):
90         # reorder output
91         yolo_outputs = [np.array([outputs[2][j, ...]]), np.array(
92             [outputs[1][j, ...]]), np.array([outputs[0][j, ...])]
93         id = img_ori[j]['file_name'].split('.')
94         result_file = open(os.path.join(output_path, id[0] + '.txt'), 'w')
95
96         image_shape = img_ori[j]['image'].shape[:2]
97         image_draw = img_ori[j]['image']
98         print(image_shape)
99
100        pred_boxes, pred_classes, pred_scores = yolo3_postprocess_np(yolo_outputs, image_shape, anchors,
101            len(class_names), model_input_shape, max_boxes=10, confidence=score_threshold,
102            elim_grid_sense=elim_grid_sense)
103
104        result_array = []
105        for box, cls, score in zip(pred_boxes, pred_classes, pred_scores):
106            result_array.append({"class": cls, "score": score})
107
108        sorted_score = sorted(
109            result_array, key=lambda d: d['score'], reverse=True)
110        for result in sorted_score:
111            result_file.write("%d,%f" % (result['class'], result['score']))
112            result_file.write('\n')
113        result_file.close()
114        os.remove(img_ori[j]['file_path'])
115
116    count = count + runSize

```

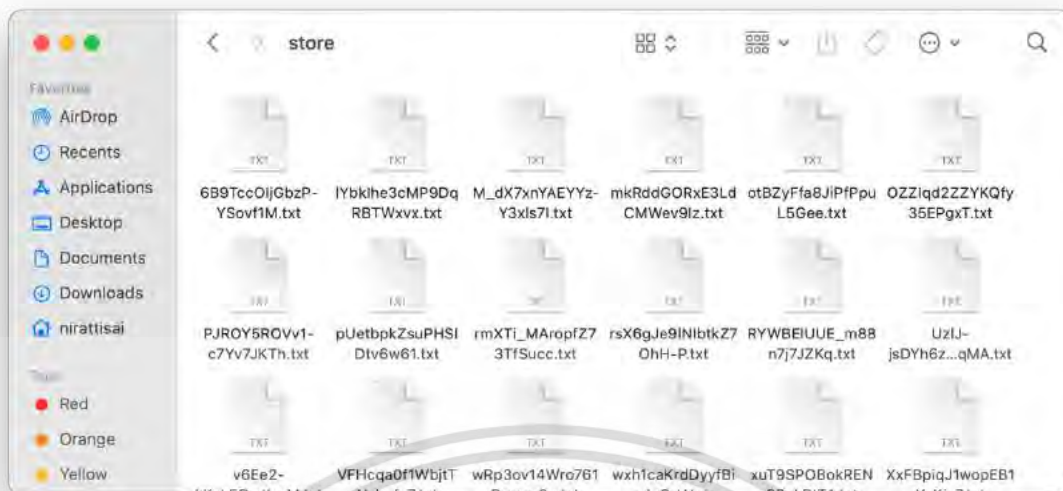
รูปที่ 4.43 Function Run Inference (2)

#### 4.3.3.3. ตัวอย่างผลลัพธ์การทำงานของ Python Application



รูปที่ 4.44 ตัวอย่างผลลัพธ์การประมวลผลด้วย Model ตรวจจับ Product

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.45 ตัวอย่างผลลัพธ์การประมวลผลด้วย Model ตรวจจับ Store

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4. ผลการดำเนินงานการทดสอบ Test Case

หลังจากดำเนินการพัฒนาระบบต้นแบบ Client-Server เสร็จสิ้นตาม Requirement ของระบบต้นแบบแล้ว ขั้นตอนต่อไปคือการทดสอบระบบโดยจะแบ่งการทดสอบออกเป็น 2 ส่วน คือ 1.ทดสอบระบบต้นแบบ Client ที่พัฒนาเป็น Mobile Application และ 2.ทดสอบระบบต้นแบบ Server ที่พัฒนาเป็น RESTful API

โดยการทดสอบจะจัดทำเป็นตารางสำหรับการทดสอบระบบ Mobile Application โดยจะระบุ ID ของ Test Case ชื่อ Test Case ขั้นตอนการทดสอบ ผลลัพธ์การทดสอบที่คาดหวัง ผลลัพธ์การทดสอบที่แท้จริง และสถานะการทดสอบตามลำดับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4.1. Test Case Mobile Application

ตารางที่ 4.21 Test Case Mobile Application

TCID	Test Case	Test Steps	Expected Result	Actual Result	Status
M01	ค้นหาสินค้าด้วยรูปภาพ	1. เข้าเมนู Search Product with Image 2. ถ่ายรูปสินค้าที่มีอยู่ในระบบ	แสดงรายการสินค้าที่ค้นพบ	แสดงรายการสินค้าที่ค้นพบ	Pass
M02	ดูรายละเอียดข้อมูลร้านค้า	1. ค้นหาสินค้าหรือร้านค้า 2. กดเลือกรายการสินค้าหรือร้านค้าที่สนใจ	แสดงรายละเอียดของร้านค้าที่ต้องการ ประกอบด้วย โลโก้ร้านค้า ชื่อร้าน รูปหน้าร้าน โปรโมชั่นของร้านค้า รายการสินค้าในร้านนั้น	แสดงรายละเอียดของร้านค้าที่ต้องการ ประกอบด้วย โลโก้ร้านค้า ชื่อร้าน รูปหน้าร้าน โปรโมชั่นของร้านค้า รายการสินค้าในร้านนั้น	Pass
M03	แสดงตำแหน่งร้านค้า	1. เข้าหน้าดูรายละเอียดร้านค้า 2. กดปุ่ม Navigation	แสดงแผนที่ในห้างสรรพสินค้าพร้อมหมุดที่อยู่ร้านค้าที่ต้องการ	แสดงแผนที่ในห้างสรรพสินค้าพร้อมหมุดที่อยู่ร้านค้าที่ต้องการ	Pass
M04	แสดงเส้นทางไปยังร้านค้า	1. เข้าหน้าดูรายละเอียดร้านค้า 2. กดปุ่ม Navigation 3. ถ่ายรูปหน้าร้านค้าตำแหน่งที่อยู่ปัจจุบัน	แสดงแผนที่ในห้างสรรพสินค้าพร้อมเส้นทางจากตำแหน่งปัจจุบันไปยังร้านค้าที่ต้องการ	แสดงแผนที่ในห้างสรรพสินค้าพร้อมเส้นทางจากตำแหน่งปัจจุบันไปยังร้านค้าที่ต้องการ	Pass
M05	ค้นหาร้านค้าในระบบ	1. เข้าเมนู Search Store 2. กรอกตัวอักษรชื่อร้านที่ต้องการค้นหา 3. กด Search	แสดงรายการร้านค้าที่เกี่ยวข้องในรายการประกอบด้วย โลโก้ร้าน และ ชื่อร้านค้า	แสดงรายการร้านค้าที่เกี่ยวข้องในรายการประกอบด้วย โลโก้ร้าน และ ชื่อร้านค้า	Pass

M06	ดูข้อมูลข่าวสาร และข้อมูลโปรโมชั่น ห้างสรรพสินค้า	1.เข้าหน้าหลัก	แสดงภาพข่าวสารและโปรโมชั่น ของห้างสรรพสินค้า	แสดงภาพข่าวสารและโปรโมชั่นของ ห้างสรรพสินค้า	Pass
M07	ค้นหาสินค้าด้วยตัวอักษร	1. เข้าเมนู Search Product 2. กรอกตัวอักษรชื่อสินค้าที่ต้องการ ค้นหา 3.กด Search	แสดงรายการสินค้าที่เกี่ยวข้อง ประกอบไปด้วย thumbnail ชื่อ สินค้า ราคา โปรโมชั่นของสินค้า และชื่อร้านค้าที่มีสินค้านั้น	แสดงรายการสินค้าที่เกี่ยวข้อง ประกอบไปด้วย thumbnail ชื่อสินค้า ราคา โปรโมชั่นของสินค้า และชื่อ ร้านค้าที่มีสินค้านั้น	Pass
M08	แจ้งข้อผิดพลาดให้ผู้ใช้งานทราบ	1. เข้าเมนู Search Product with Image 2. ถ่ายรูปสินค้าที่ไม่มีอยู่ในระบบ	แสดงข้อผิดพลาดของระบบและ แจ้งวิธีดำเนินการแก้ไข	แสดงข้อผิดพลาดของระบบและแจ้ง วิธีดำเนินการแก้ไข	Pass

#### 4.4.2. Test Case RESTful API

ตารางที่ 4.22 Test Case RESTful API

TCID	Test Case	Test Steps	Expected Result	Actual Result	Status
A01	รับข้อมูลรูปภาพจาก Client และ อ่านผลลัพธ์การตรวจจับวัตถุของ ภาพนั้น	1. ส่งข้อมูลรูปภาพในรูปแบบ Base 64 พร้อมกับประเภทของรูปภาพ 2. รอผลลัพธ์จาก Python Application 3. อ่านผลลัพธ์	ไฟล์รูปภาพในเครื่อง Server และ ตั้งชื่อภาพเป็น ID รอให้ Python Application ประมวลผลและสร้าง ผลลัพธ์เป็น TXT ไฟล์ที่มีชื่อไฟล์ เป็น ID เดียวกันอ่านผลลัพธ์ของ การตรวจจับ และนำผลลัพธ์ไป ค้นหาข้อมูลในฐานข้อมูลส่งกลับให้ Client	ไฟล์รูปภาพในเครื่อง Server และตั้ง ชื่อภาพเป็น ID รอให้ Python Application ประมวลผลและสร้าง ผลลัพธ์เป็น TXT ไฟล์ที่มีชื่อไฟล์เป็น ID เดียวกันอ่านผลลัพธ์ของการ ตรวจจับ และนำผลลัพธ์ไปค้นหาข้อมูล ในฐานข้อมูลส่งกลับให้ Client	Pass

A02	แจ้งข้อผิดพลาดการตรวจจับวัตถุ	<ol style="list-style-type: none"> <li>1. ส่งข้อมูลรูปภาพในรูปแบบ Base 64 พร้อมกับประเภทของรูปภาพ</li> <li>2. รวบรวมผลลัพธ์จาก Python Application</li> <li>3. อ่านผลลัพธ์</li> </ol>	ไฟล์รูปภาพในเครื่อง Server และตั้งชื่อภาพเป็น ID รอให้ Python Application ประมวลผลและสร้างผลลัพธ์เป็น TXT ไฟล์ที่มีชื่อไฟล์เป็น ID เดียวกันอ่านผลลัพธ์ของการตรวจจับที่ไม่สามารถตรวจจับได้ ส่งกลับข้อผิดพลาดการตรวจจับวัตถุให้ Client	ไฟล์รูปภาพในเครื่อง Server และตั้งชื่อภาพเป็น ID รอให้ Python Application ประมวลผลและสร้างผลลัพธ์เป็น TXT ไฟล์ที่มีชื่อไฟล์เป็น ID เดียวกันอ่านผลลัพธ์ของการตรวจจับที่ไม่สามารถตรวจจับได้ ส่งกลับข้อผิดพลาดการตรวจจับวัตถุให้ Client	Pass
A03	แจ้งข้อผิดพลาดการทำงานของ Python Application	<ol style="list-style-type: none"> <li>1. ส่งข้อมูลรูปภาพในรูปแบบ Base 64 พร้อมกับประเภทของรูปภาพ</li> <li>2. รวบรวมผลลัพธ์จาก Python Application</li> <li>3. อ่านผลลัพธ์</li> </ol>	ไฟล์รูปภาพในเครื่อง Server และตั้งชื่อภาพเป็น ID และรอ Python Application ประมวลผลเกินขอบเขตเวลาที่กำหนดไว้ ส่งกลับข้อผิดพลาดการทำงานให้ Client	ไฟล์รูปภาพในเครื่อง Server และตั้งชื่อภาพเป็น ID และรอ Python Application ประมวลผลเกินขอบเขตเวลาที่กำหนดไว้ ส่งกลับข้อผิดพลาดการทำงานให้ Client	Pass
A04	เพิ่มข้อมูลร้านค้า	1. ส่งข้อมูลร้านค้าที่จำเป็นในรูปแบบ JSON ประกอบไปด้วย ID ร้านค้า ชื่อร้านค้า รูปภาพหน้าร้านค้า และโลโก้ร้านค้า	สร้างข้อมูลร้านค้าใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	สร้างข้อมูลร้านค้าใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	Pass
A05	เพิ่มข้อมูลโปรโมชั่นของร้านค้า	1. ส่งข้อมูลโปรโมชั่นของร้านค้าที่จำเป็นในรูปแบบ JSON ประกอบไปด้วย ID ร้านค้า ชื่อโปรโมชั่น รายละเอียดโปรโมชั่น เวลาเริ่มต้นโปรโมชั่น และเวลาสิ้นสุดโปรโมชั่น	สร้างข้อมูลโปรโมชั่นใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	สร้างข้อมูลโปรโมชั่นใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	Pass
A06	เรียกดูข้อมูลร้านค้าทั้งหมดใน	1. ส่งคำขอเรียกดูข้อมูลร้านค้าทั้งหมด	แสดงรายการข้อมูลร้านค้าทั้งหมดที่มีอยู่ในฐานข้อมูล	แสดงรายการข้อมูลร้านค้าทั้งหมดที่มีอยู่ในฐานข้อมูล	Pass

A07	เรียกดูรายละเอียดร้านค้า	1. ส่งคำขอเรียกดูรายละเอียดข้อมูลของร้านค้านั้นโดยอ้างอิงด้วย ID	แสดงข้อมูลรายละเอียดข้อมูลร้านค้าที่ถูกอ้างอิงด้วย ID	แสดงข้อมูลรายละเอียดข้อมูลร้านค้าที่ถูกอ้างอิงด้วย ID	Pass
A08	ค้นหาร้านค้าด้วยชื่อ	1. ส่งคำขอค้นหาข้อมูลของร้านค้าด้วยชื่อร้านค้า	แสดงรายการข้อมูลร้านค้าที่เกี่ยวข้องทั้งหมด	แสดงรายการข้อมูลร้านค้าที่เกี่ยวข้องทั้งหมด	Pass
A09	อัปเดตข้อมูลร้านค้า	1. ส่งคำขออัปเดตข้อมูลร้านค้าโดยอ้างอิง ID ของร้านค้า และส่งข้อมูลที่ต้องการอัปเดตในรูปแบบ JSON	อัปเดตข้อมูลร้านค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การอัปเดตข้อมูลร้านค้าในฐานข้อมูล	อัปเดตข้อมูลร้านค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การอัปเดตข้อมูลร้านค้าในฐานข้อมูล	Pass
A10	ลบข้อมูลร้านค้า	1. ส่งคำขอลบข้อมูลร้านค้าโดยอ้างอิง ID ของร้านค้า	ลบข้อมูลร้านค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การลบข้อมูลร้านค้าในฐานข้อมูล	ลบข้อมูลร้านค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การลบข้อมูลร้านค้าในฐานข้อมูล	Pass
A11	เพิ่มข้อมูลสินค้า	1. ส่งข้อมูลสินค้าที่จำเป็นในรูปแบบ JSON ประกอบไปด้วย ID ร้านค้าที่เป็นเจ้าของร้านค้า ชื่อสินค้า รูปภาพสินค้า และราคา	สร้างข้อมูลสินค้าใหม่ในฐานข้อมูลและตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	สร้างข้อมูลสินค้าใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	Pass
A12	เพิ่มข้อมูลโปรโมชั่นของสินค้า	1. ส่งข้อมูลโปรโมชั่นของสินค้าที่จำเป็นในรูปแบบ JSON ประกอบไปด้วย ID สินค้า ชื่อโปรโมชั่น ราคาโปรโมชั่น เวลาเริ่มต้นโปรโมชั่น และเวลาสิ้นสุดโปรโมชั่น	สร้างข้อมูลโปรโมชั่นใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	สร้างข้อมูลโปรโมชั่นใหม่ในฐานข้อมูล และตอบกลับผลลัพธ์การเพิ่มข้อมูลลงฐานข้อมูล	Pass
A13	เรียกดูข้อมูลสินค้าทั้งหมดใน	1. ส่งคำขอเรียกดูข้อมูลสินค้าทั้งหมด	แสดงรายการข้อมูลสินค้าทั้งหมดที่มีอยู่ในฐานข้อมูลโดยจำกัดที่ 20 รายการ	แสดงรายการข้อมูลสินค้าทั้งหมดที่มีอยู่ในฐานข้อมูลโดยจำกัดที่ 20 รายการ	Pass
A14	เรียกดูรายละเอียดสินค้า	1. ส่งคำขอเรียกดูรายละเอียดข้อมูลของสินค้านั้นโดยอ้างอิงด้วย ID	แสดงข้อมูลรายละเอียดข้อมูลสินค้าที่ถูกอ้างอิงด้วย ID	แสดงข้อมูลรายละเอียดข้อมูลสินค้าที่ถูกอ้างอิงด้วย ID	Pass

A15	ค้นหาสินค้าด้วยชื่อ	1. ส่งคำขอค้นหาข้อมูลของสินค้าด้วยชื่อสินค้า	แสดงรายการข้อมูลสินค้าที่เกี่ยวข้องทั้งหมด	แสดงรายการข้อมูลสินค้าที่เกี่ยวข้องทั้งหมด	Pass
A16	อัปเดตข้อมูลสินค้า	1. ส่งคำขออัปเดตข้อมูลสินค้าโดยอ้างอิง ID ของสินค้า และส่งข้อมูลที่ต้องการอัปเดตในรูปแบบ JSON	อัปเดตข้อมูลสินค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การอัปเดตข้อมูลสินค้าในฐานข้อมูล	อัปเดตข้อมูลสินค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การอัปเดตข้อมูลสินค้าในฐานข้อมูล	Pass
A17	ลบข้อมูลสินค้า	1. ส่งคำขอลบข้อมูลสินค้าโดยอ้างอิง ID ของสินค้า	ลบข้อมูลสินค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การลบข้อมูลสินค้าในฐานข้อมูล	ลบข้อมูลสินค้าที่อ้างอิงในฐานข้อมูลและตอบกลับผลลัพธ์การลบข้อมูลสินค้าในฐานข้อมูล	Pass

#### 4.4.3. Test Case โมเดลตรวจจับสินค้า หน้าร้านค้า และ Python Application

ตารางที่ 4.23 Test Case โมเดลตรวจจับสินค้า หน้าร้านค้า และ Python Application

TCID	Test Case	Test Steps	Expected Result	Actual Result	Status
AI01	พัฒนาโมเดลตรวจจับสินค้า	1. ประเมินผลโมเดลสินค้าด้วย Darknet Framework 2. แปลงโมเดลเป็น Fixed-point 3. ใช้งานบนการ์ด FPGA	มีจำนวนคลาสที่สามารถตรวจจับรูปสินค้ามากกว่า 50 คลาส มีค่าเฉลี่ยความแม่นยำสูง (mAP) กว่า 0.5 ขึ้นไป สามารถใช้งานบนการ์ด FPGA ได้	มีจำนวนคลาสที่สามารถตรวจจับรูปสินค้า 59 คลาส มีค่าเฉลี่ยความแม่นยำ (mAP) อยู่ที่ 0.612 สามารถใช้งานบนการ์ด FPGA ได้	Pass
AI02	พัฒนาโมเดลตรวจจับร้านค้า	1. ประเมินผลโมเดลสินค้าด้วย Darknet Framework 2. แปลงโมเดลเป็น Fixed-point 3. ใช้งานบนการ์ด FPGA	มีจำนวนคลาสที่สามารถตรวจจับรูปหน้าร้านค้ามากกว่า 20 คลาส มีค่าเฉลี่ยความแม่นยำสูง (mAP) กว่า 0.5 ขึ้นไป	มีจำนวนคลาสที่สามารถตรวจจับรูปหน้าร้านค้า 30 คลาส มีค่าเฉลี่ยความแม่นยำ (mAP) อยู่ที่ 0.8913	Pass

			สามารถใช้งานบนการ์ด FPGA ได้	สามารถใช้งานบนการ์ด FPGA ได้	
AI03	โหลดโครงสร้างโมเดลเข้าการ์ด FPGA	1. นำโมเดลสินค้าหรือร้านค้าเก็บใส่ไว้ใน โพลเดอร์ที่กำหนด 2. เช็คว่าจำนวน subgraphs ของโมเดล 3. สร้าง runner ด้วยคำสั่งที่ Vitis-AI เตรียมมาให้	ได้ object runner ที่มี subgraphs เพียง 1 subgraphs	ได้ object runner ที่มี subgraphs เพียง 1 subgraphs	Pass
AI04	เตรียมรูปภาพก่อนประมวลผล	1. นำรูปภาพเก็บใส่ไว้ในโพลเดอร์ที่ กำหนด 2. อ่านรูปภาพผ่าน cv2.imread 3. นำรูปเข้าฟังก์ชันเตรียมรูปภาพ	ได้ numpy array ของข้อมูล รูปภาพขนาดที่มีขนาดที่กำหนด และผ่านการ normalize	ได้ numpy array ของข้อมูลรูปภาพ ขนาดที่มีขนาดที่กำหนดและผ่านการ normalize	Pass
AI05	นำรูปภาพที่เตรียมแล้วเข้า ประมวลผลการตรวจจับสินค้าหรือ หน้าร้านค้า	1. สร้าง array สำหรับเก็บ numpy array ของข้อมูลรูปภาพที่เตรียมแล้ว ขนาด 5 ช่อง 2. นำเข้า array สำหรับเก็บ numpy array รูปภาพให้โมเดลประมวลผล 5 ช่องพร้อมกัน	สามารถนำเข้ารูปภาพได้พร้อมกัน มากที่สุด 5 รูป	สามารถนำเข้ารูปภาพได้พร้อมกันมาก สุด 5 รูป	Pass
AI06	ถอดรหัสผลลัพธ์ที่ได้จากการ ประมวลผลการตรวจจับสินค้าหรือ หน้าร้านค้า	1. นำเข้า array สำหรับเก็บ numpy array รูปภาพให้โมเดลประมวลผล	ได้ผลลัพธ์ตำแหน่งกรอบวัตถุสินค้า หรือหน้าร้านค้าที่ตรวจเจอ ชื่อ คลาส และค่าความมั่นใจ	ได้ผลลัพธ์ตำแหน่งกรอบวัตถุสินค้าหรือ หน้าร้านค้าที่ตรวจเจอ ชื่อคลาส และ ค่าความมั่นใจ	Pass
AI07	สร้างไฟล์ผลลัพธ์การตรวจจับ เป็น txt ไฟล์	1. นำเข้า array สำหรับเก็บ numpy array รูปภาพให้โมเดลประมวลผล	ได้ไฟล์ผลลัพธ์การตรวจจับเป็น txt ไฟล์ ในแต่ละบรรทัดประกอบไป ด้วยชื่อคลาสสินค้าหรือหน้าร้านค้า ที่ตรวจเจอ และค่าความมั่นใจ	ได้ไฟล์ผลลัพธ์การตรวจจับเป็น txt ไฟล์ ในแต่ละบรรทัดประกอบไปด้วย ชื่อคลาสสินค้าหรือหน้าร้านค้าที่ตรวจ เจอ และค่าความมั่นใจ	Pass

#### 4.5. Traceability matrix

Traceability Matrix เป็นเอกสารหรือเครื่องมือที่ใช้ในการติดตามความสัมพันธ์ระหว่างรายการต่างๆ ในโครงการ เช่น Requirements, Design และ Test case เพื่อให้แน่ใจว่ามีการปฏิบัติตามความต้องการและมีการทดสอบแล้วว่าถูกต้อง

ตารางที่ 4.24 ตาราง Traceability Matrix

Functional ID	GUI ID	API ID	Test ID	หมายเหตุ
FR01	รูปที่ 4.3	-	M06	ดูข้อมูล ห้างสรรพสินค้า
FR02, FR04, FR12, FR13, FR14, FR17, FR18, FR25, FR27, FR28, FR29, FR30, FR31, FR32	รูปที่ 4.2	01	M01, A01, AI01, AI03, AI04, AI05, AI06, AI07	ค้นหาสินค้าด้วยภาพ
FR07	รูปที่ 4.9	05	M02, A07	ดูข้อมูลร้านค้า
FR09, FR10, FR12, FR13, FR14, FR17, FR19, FR26, FR27, FR28, FR29, FR30, FR31, FR32	รูปที่ 4.18	≈	M04, AI02, AI03, AI04, AI05, AI06, AI07	ระบุตำแหน่งร้านค้า ต้นทาง
FR08, FR09, FR10	รูปที่ 4.16, รูปที่ 4.17, รูปที่ 4.18	-	M03, M04,	การทำงานร่วมกับ ระบบ Navigation Path
FR11, FR15, FR16, FR20	รูปที่ 4.21, รูปที่ 4.22	01	M08, A02, A03, AI07	แจ้งข้อผิดพลาด
FR05, FR06	รูปที่ 4.19	06,	M05, A15	ค้นหาข้อมูลร้านค้า จากชื่อ
FR03, FR04,	รูปที่ 4.20	13	M07, A08	ค้นหาข้อมูลสินค้า จากชื่อ
FR21, FR22, FR23, FR24	-	02, 03, 07, 08, 09, 10, 14, 15	A04, A05, A09, A10, A11, A12, A16, A17	จัดการข้อมูลใน ฐานข้อมูล

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอ

หัวข้อสหกิจศึกษานี้ถูกจัดทำเพื่อศึกษากระบวนการดำเนินงานวิจัย และปฏิบัติการพัฒนาระบบต้นแบบโครงการ Visual Search ซึ่งดำเนินการภายใต้การกำกับดูแลของ บริษัท โทรคมนาคมแห่งชาติ จำกัด (มหาชน) ส่วนปฏิบัติการงานวิจัยดิจิทัล ศูนย์วิจัยและพัฒนา Brownien Lab โดยโครงการงานวิจัยนี้มุ่งเน้นไปที่การพัฒนาระบบต้นแบบสำหรับการใช้งานเทคโนโลยี DPU ของ Xilinx ซึ่งคณะผู้จัดทำสามารถสรุปผลการดำเนินงานวิจัยได้ดังนี้

#### 5.1. สรุปผลการดำเนินงาน

การพัฒนาระบบต้นแบบสำหรับการใช้งานเทคโนโลยี The Xilinx® Deep Learning Processor Unit (DPU) ในหัวข้อโครงการ Visual Search สามารถสรุปได้เป็น 3 ส่วนหลัก ๆ ได้แก่ 1) ส่วนของ Client เป็นระบบเบื้องหน้า (Front-end) ที่จัดทำในรูปแบบแอปพลิเคชันมือถือ 2) ส่วนของ Server เป็นระบบเบื้องหลัง (Back-end) ที่จัดทำในรูปแบบ RESTful APIs และ 3) ส่วน Python Application โดยสิ่งได้เรียนรู้จากการดำเนินงานหัวข้อสหกิจศึกษานี้มีดังนี้

1. ความรู้ในเรื่องชิป Field-Programmable Gate Array (FPGA)
2. หลักการทำงานของเทคโนโลยี The Xilinx® Deep Learning Processor Unit ประเภทต่าง ๆ ทั้งแบบที่เป็นชิป on board และชิปการ์ด
3. การติดตั้งและใช้งานเทคโนโลยี The Xilinx® Deep Learning Processor Unit ผ่านการลงมือปฏิบัติจริงบนเครื่อง Server ที่ศูนย์วิจัยและพัฒนา Brownien Lab
4. ความรู้โครงสร้างโมเดลปัญญาประดิษฐ์ประเภทการประมวลผลภาพ
5. การพัฒนาและปรับแต่งโครงสร้างโมเดลปัญญาประดิษฐ์ประเภทการประมวลผลภาพ ด้วย Darknet framework, TensorFlow และ Pytorch
6. การแปลงและปรับแต่งโมเดลปัญญาประดิษฐ์ให้มีขนาดเล็กลงสามารถใช้งานกับเทคโนโลยี The Xilinx® Deep Learning Processor Unit สำหรับเร่งการประมวลผล
7. การใช้งาน Vitis-AI Development Platform
8. การสร้างสภาพแวดล้อมจำลองด้วย Docker สำหรับการใช้งานเทคโนโลยี The Xilinx® Deep Learning Processor Unit
9. การพัฒนา Python Application ในการเรียกใช้งานเครื่องมือ High-Level API สำหรับการใช้งานเทคโนโลยี The Xilinx® Deep Learning Processor Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า งานเทคโนโลยี The Xilinx® Deep Learning Processor Unit  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. กระบวนการออกแบบและพัฒนาระบบต้นแบบ Client-Server และส่วนประสานการทำงาน รวมกันกับ Python Application
11. การพัฒนาแอปพลิเคชันมือถือด้วย React Native
12. การพัฒนา RESTful API ในการติดต่อสื่อสารระหว่าง Client-Server
13. การพัฒนา Database ด้วย PostgreSQL สำหรับเก็บข้อมูลในระบบต้นแบบ
14. การใช้งาน Git Version Control ในการทำงานร่วมกับเพื่อนร่วมทีม

## 5.2. ปัญหาที่พบเจอ

เนื่องจากเทคโนโลยี The Xilinx® Deep Learning Processor Unit เป็นเทคโนโลยีที่เกิดขึ้นใหม่ และยังไม่เสถียร และทำให้การดำเนินงานวิจัยนี้คณะผู้จัดทำได้ประสบปัญหาดังต่อไปนี้

1. การใช้งานและการพัฒนาระบบต้นแบบสำหรับการใช้งานเทคโนโลยี The Xilinx® Deep Learning Processor Unit เป็นไปอย่างยากลำบาก เพราะมีการปรับเปลี่ยนคำสั่งต่าง ๆ อย่าง ต่อเนื่อง ทำให้ต้องอัปเดตข้อมูลในการดำเนินการใหม่
2. Community การใช้งาน DPU มีน้อยทำให้แก้ไขปัญหา error ของตัวการ์ดได้ยาก
3. โมเดล AI ที่เทคโนโลยี DPU รองรับยังมีไม่มากนัก ทำให้การพัฒนาโมเดล AI จำเป็นจะต้องใช้ โครงสร้างโมเดลเวอร์ชันเก่าที่ไม่ทันสมัย
4. มีการขาดตอนของโมเดล เนื่องจากตัวดำเนินการยังไม่ Support จึงจำเป็นต้องปรับแต่ง โครงสร้างโมเดลทำให้ประสิทธิภาพการตรวจจับลดลงเล็กน้อย
5. ขาดสิทธิในการเก็บรวบรวมข้อมูลในห้างสรรพสินค้า
6. ไม่มีชุดข้อมูลที่มากพอสำหรับการฝึกสอนโมเดลให้มีประสิทธิภาพ และเนื่องจากทางคณะผู้จัดทำ จะต้องเก็บรวบรวมข้อมูลด้วยตัวเองทั้งหมด และ Labeling ข้อมูลในเวลาที่จำกัด

## 5.3. ข้อเสนอแนะ

ในการต่อยอดและพัฒนาระบบต้นแบบให้เป็นระบบที่สามารถใช้งานจริงในภาคธุรกิจได้นั้น คณะผู้จัดทำสามารถสรุปข้อเสนอแนะเป็นหัวข้อได้ดังนี้

1. มีการปรับปรุงและพัฒนาระบบอยู่อย่างสม่ำเสมอ ให้ทันต่อการเปลี่ยนแปลงชุดคำสั่งการใช้งาน เทคโนโลยี The Xilinx® Deep Learning Processor Unit เพื่อการใช้งานเทคโนโลยีอย่างมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เพิ่มข้อมูลฝึกสอนให้โมเดลปัญญาประดิษฐ์ที่ใช้ในการตรวจจับสินค้าหรือหน้าร้านค้า เพื่อให้มีความแม่นยำมากขึ้นในการใช้งานจริง
3. เน้นการพัฒนาระบบไปที่การตรวจจับหน้าร้านค้าและระบบ Indoor-Navigation เนื่องจากหน้าร้านค้าในห้างสรรพสินค้ามีการปรับเปลี่ยนบ่อยครั้งน้อยจึงทำให้โมเดลปัญญาประดิษฐ์ที่ใช้ในการตรวจจับหน้าร้านค้ามีความคงทน และความแม่นยำสูง อีกทั้งระบบ Indoor-Navigation ยังเป็นความสามารถที่ตอบโจทย์ในด้านธุรกิจได้มาก และมีมูลค่าสูง
4. การพัฒนาระบบใหญ่ควรมีแบบแผนที่แน่ชัดเพื่อให้ง่ายต่อการดำเนินงาน และรองรับการเปลี่ยนแปลงที่จะเกิดขึ้น
5. มีการพัฒนาระบบ Web service และระบบเบื้องหน้า (Front-end) สำหรับการจัดการข้อมูลในฐานข้อมูลที่สะดวกขึ้นในอนาคต
6. ออกแบบฐานข้อมูลใหม่ เนื่องจากฐานข้อมูลในระบบนี้ยังมีข้อผิดพลาดเช่น ราคาสินค้าที่เก็บเป็นเลขจำนวนเต็ม
7. เปลี่ยนการลบข้อมูลเป็นการเลิกใช้งานข้อมูล ออกแบบการลบข้อมูลในฐานข้อมูลใหม่ควรจะสร้างเป็น Attribute สถานะของข้อมูล เพื่อไม่ให้เกิดการลบ record ไปกระทบกับส่วนอื่น ๆ ที่มีความสัมพันธ์กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เอกสารอ้างอิง

Surapong Kanoktipsatharporn. (27 สิงหาคม 2022). *Object Detection คืออะไร*. เข้าถึงได้จาก BUA Labs: <https://www.bualabs.com/archives/3453/what-is-object-detection-tutorial-tensorflow-js-build-object-detection-machine-learning-coco-ssd-tfjs-ep-8/>

codebee. (27 สิงหาคม 2022). *React Native คืออะไร ? เครื่องมือพัฒนา Mobile Apps แบบ Cross-Platform*. เข้าถึงได้จาก codebee: <https://www.codebee.co.th/labs/react-native-คืออะไร/>

วิกิพีเดีย สารานุกรมเสรี. (22 สิงหาคม 2022). *โทรศัพท์เคลื่อนที่*. เข้าถึงได้จาก วิกิพีเดีย: <https://th.wikipedia.org/wiki/โทรศัพท์เคลื่อนที่>

Beam Warasanan. (27 สิงหาคม 2022). *Objective Detection เทคโนโลยีตรวจจับวัตถุ*. เข้าถึงได้จาก SENALABS: <https://www.sennalabs.com/th/blogs/Objective-Detection>

Thum Namprom. (27 สิงหาคม 2022). *Image Recognition Technology*. เข้าถึงได้จาก Propea: <https://properea.com/image-recognitions-technology-17-06-2021/>

Divya Sheel. (27 สิงหาคม 2022). *Deep Learning คืออะไร?* เข้าถึงได้จาก ABB: <https://new.abb.com/news/detail/58004/deep-learning>

Khuyen Le. (27 สิงหาคม 2022). *An overview of VGG16 and NiN models*. เข้าถึงได้จาก Medium: <https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484>

Amazon. (27 สิงหาคม 2022). *API คืออะไร*. เข้าถึงได้จาก Amazon: <https://aws.amazon.com/th/what-is/api/>

Xilinx. (27 สิงหาคม 2022). *DPU for Convolutional Neural Network*. เข้าถึงได้จาก Xilinx: <https://www.xilinx.com/products/intellectual-property/dpu.html>

Xilinx. (27 สิงหาคม 2022). *Alveo U50 Data Center Accelerator Card Data Sheet*. เข้าถึงได้จาก Xilinx: [https://www.xilinx.com/content/dam/xilinx/support/documents/data\\_sheets/ds965-u50.pdf](https://www.xilinx.com/content/dam/xilinx/support/documents/data_sheets/ds965-u50.pdf)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Xilinx. (27 สิงหาคม 2022). *Supported Operators and DPU Limitations*. เข้าถึงได้จาก Xilinx:

<https://docs.xilinx.com/r/1.4.1-English/ug1414-vitis-ai/Supported-Operators-and-DPU-Limitations>

Patchara Boonmathanaruk. (27 สิงหาคม 2022). *TECHNOLOGY Docker คืออะไร? เครื่องมือยอดนิยมที่จะช่วย Dev ทำงานเร็วขึ้น 2 เท่า!* เข้าถึงได้จาก Skooldio: <https://blog.skooldio.com/what-is-docker/>

Xilinx. (27 สิงหาคม 2022). *Xilinx/Vitis-AI*. เข้าถึงได้จาก Github: <https://github.com/Xilinx/Vitis-AI>

linux-console.net. (16 มกราคม 2023). *PostgreSQL คืออะไร? PostgreSQL ทำงานอย่างไร*. เข้าถึงได้จาก linux-console.net: <https://th.linux-console.net/?p=1801>

Arnon Puitrakul. (16 มกราคม 2023). *SQLite vs MySQL vs PostgreSQL เลือก RDMS ตัวไหนมาลง Project หน้าที่*. เข้าถึงได้จาก Arnodora: <https://arnodora.in.th/sqlite-vs-mysql-vs-postgresql>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำรับรองเล่มสหกิจศึกษาโดยสถานประกอบการ

วันที่ ..31.. เดือน .....มีนาคม..... พ.ศ. ....2566.....

ข้าพเจ้า .....ดร.สมรัักษ์ เพชรชาติรี.....ตำแหน่ง .....ผู้จัดการส่วนปฏิบัติการวิจัย ฝ่ายวิจัยและพัฒนา.....

เป็นตัวแทนของสถานประกอบการ .....บริษัท โทรคมานาคมแห่งชาติ จำกัด.....

ขอรับรองว่า ทางสถานประกอบการได้ตรวจสอบเล่มสหกิจศึกษา

เรื่อง .....ระบบต้นแบบค้นหาสินค้าและระบุเส้นทางด้วยภาพ (Visual Search).....

ของนักศึกษาชื่อ .....นายนริชฌา เสถียรปภิรณกร รหัสนักศึกษา 62050181 และ.....

.....นายนิรติศัย สารุ รหัสนักศึกษา 62050187.....ซึ่งเป็นนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เรียบร้อยแล้ว และไม่มีส่วนหนึ่งส่วน

ใดในเล่มสหกิจศึกษานี้ที่มีข้อมูลอ่อนไหว และ/หรือ ข้อมูลอันเป็นความลับอันจะก่อให้เกิดความเสียหายต่อสถาน

ประกอบการ รวมทั้งอนุญาตให้สามารถเผยแพร่ต่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ จึง

ลงชื่อไว้เป็นหลักฐาน

ลงชื่อ .....  
(.....ดร.สมรัักษ์ เพชรชาติรี.....)

ตัวแทนสถานประกอบการ

ข้าพเจ้า .....ดร.ประพจน์ ศรีนวัตติวงศ์.....อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบเล่มสหกิจศึกษาแล้วและรับทราบว่ายสถานประกอบการดำเนินการตรวจสอบเล่มสหกิจศึกษาแล้ว จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ .....  
(.....ดร.ประพจน์ ศรีนวัตติวงศ์.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้