

โครงการเลขที่ วศ.คพ. P008-2/2565

เรื่อง

แคปสแนป : ระบบจัดการการซื้อ-ขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้เทคนิค
ความฉลาดเชิงคำนวณ

โดย

นายพงศกร รัตนพันธ์ รหัส 630610749
นางสาวศุภาริญญา ศิลปะลิทธิ์ รหัส 630610765

โครงการนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2565

PROJECT No. CPE P008-2/2565

**CapSnap : Retail self-checkout system using Computational Intelligence
Technique**

**Pongsakorn Rattanapan 630610749
Suparida Silpasith 630610765**

**A Project Submitted in Partial Fulfillment of Requirements
for the Degree of Bachelor of Engineering
Department of Computer Engineering
Faculty of Engineering
Chiang Mai University
2022**

หัวข้อโครงการ : แคปสแนป : ระบบจัดการการซื้อ-ขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้เทคนิคความฉลาดเชิงคำนวณ
: CapSnap : Retail self-checkout system using Computational Intelligence Technique

โดย : นายพงศกร รัตนพันธ์ รหัส 630610749
นางสาวศุภาริยา ศิลปสิทธิ์ รหัส 630610765

ภาควิชา : วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษา : รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล

ปริญญา : วิศวกรรมศาสตรบัณฑิต

สาขา : วิศวกรรมคอมพิวเตอร์

ปีการศึกษา : 2565

ภาควิชาฯวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ได้อนุมัติให้โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต (สาขาวิศวกรรมคอมพิวเตอร์)

คณะกรรมการสอบโครงการ

..... หัวหน้าภาควิชาฯวิศวกรรมคอมพิวเตอร์
(รศ.ดร. สันติ พิทักษ์กิจนุกร)

..... ประธานกรรมการ
(รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล)

..... กรรมการ
(ผศ.ดร. เกษมสิทธิ์ ตียพันธ์)

..... กรรมการ
(รศ.ดร. นิพนธ์ ธีร正宗)

หัวข้อโครงการ : แคปسنป : ระบบจัดการการซื้อ-ขายในร้านค้าปลีกอัตโนมัติด้วยตนเองโดยใช้เทคนิคความฉลาดเชิงคำนวณ
: CapSnap : Retail self-checkout system using Computational Intelligence Technique

โดย : นายพงศกร รัตนพันธ์ รหัส 630610749
นางสาวศุภริภา ศิลปสิทธิ์ รหัส 630610765

ภาควิชา : วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา : รศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล

ปริญญา : วิศวกรรมศาสตรบัณฑิต

สาขา : วิศวกรรมคอมพิวเตอร์

ปีการศึกษา : 2565

บทคัดย่อ

ในปัจจุบันการซื้อสินค้าปลีก ณ จุดจำหน่ายสินค้าทั่วไปในประเทศไทย เช่น ชูปเปอร์มาเก็ต ห้างสรรพสินค้า หรือ ร้านค้าปลีกต่าง ๆ นั้น มีการใช้ระบบการจ่ายเงินที่จุดชำระสินค้า หรือแคชเชียร์กับพนักงาน ซึ่งระบบดังกล่าวนั้นไม่ตอบโจทย์กับความต้องการ และความสะดวกสบายของผู้ซื้อสินค้า เนื่องจากมีข้อเสีย คือการรอและต่อ隊列เพื่อชำระสินค้า นอกจากนี้จะเห็นว่าปัจจุบันบริษัทต่าง ๆ ได้เปลี่ยนรูปแบบการบริการให้ผู้บริโภคสามารถทำธุรกรรมได้ ผ่านโทรศัพท์มือถือ จึงมีความสมควรที่จะพัฒนาระบบซอฟแวร์ที่เป็นตัวช่วยให้ลูกค้าสามารถซื้อสินค้าได้ด้วยตนเอง โดยไม่ต้องผ่านจุดบริการของร้านค้า

แคปسنป(CapSnap) เป็นระบบจัดการการซื้อ-ขายสินค้าในร้านค้าปลีก เพื่ออำนวยความสะดวก และลดความยุ่งยากในการ ซื้อ-ขายสินค้าทั้งทางฝั่งลูกค้า และฝั่งร้านค้า โดยลูกค้าสามารถจัดการการซื้อสินค้าได้ด้วยตนเองผ่านทางแอปพลิเคชัน ในโทรศัพท์มือถือ โดยแอปพลิเคชันดังกล่าวจะทำการเชื่อมต่อกับร้านค้าเมื่อลูกค้าแสดงเข้าร้านค้า จากนั้นลูกค้าสามารถสแกนสินค้าที่ต้องการแบบเรียลไทม์ ซึ่งระบบจะใช้หลักการทำงาน Computational Intelligence เพื่อบอกรายละเอียดและราคาของสินค้าแต่ละชนิดที่ถูกสแกน ซึ่งจะเป็นประโยชน์ต่อคนที่มีปัญหาในการมองเห็นในการเลือกซื้อสินค้า ลูกค้าสามารถระบุรายการสินค้าได้ด้วยตนเอง ผ่านการเพิ่ม หรือลดสินค้าในตะกร้าของแอปพลิเคชัน ลูกค้าสามารถทำการชำระเงินด้วยตนเองผ่านแอปพลิเคชันหลังสแกนออกจากร้านค้า ซึ่งระบบรองรับการใช้งานแอปพลิเคชันกับร้านค้าปลีกหลาย ๆ ร้านที่มีสินค้าแตกต่างกัน นอกจากนี้ระบบยังมี Website Dashboard ให้บริการสำหรับผู้ร้านค้า เพื่อช่วยให้ร้านค้าปลีกสามารถจัดการร้านได้อย่างมีประสิทธิภาพมากขึ้น โดยสามารถจัดการ และดูข้อมูลคลังสินค้า รวมถึงข้อมูลการขายสินค้าได้ในเว็บไซต์เดียว

Project Title : CapSnap : Retail self-checkout system using Computational Intelligence Technique

Name : Pongsakorn Rattanapan 630610749
Suparida Silpasith 630610765

Department : Computer Engineering

Project Advisor : Assoc. Prof. Sansanee Auephanwiriyakul, Ph.D.

Degree : Bachelor of Engineering

Program : Computer Engineering

Academic Year : 2022

ABSTRACT

Nowadays, purchasing retail products at general retail stores in Thailand, such as supermarkets, department stores, or small retailers offer payment at the checkout point or cashier, which does not satisfy the requirements and convenience of customers because there are disadvantages, namely waiting and queuing up to pay. In addition, it can be seen at present that different companies have modified their services permitting consumers to conduct transactions using mobile phones. Thus, it is appropriate to develop a software system that enables clients to independently purchase things and checkout without having to go through the store's service counter. In order to simplify and lessen the complexity of purchasing and selling products on both the consumer side and the store side, we have designed the (CapSnap) system for handling purchasing in retail establishments via mobile application that can connect to the store after a customer scans into it and allows customers to make their own purchases. Moreover, customers can live-stream product images using the application in real-time. The details and costs of each scanned item are then retrieved by the system using a computational intelligence technique, which is advantageous for users with visual impairments. The shopping cart feature of the application allows users to add and remove products. Once scanning out of the store, customers can handle their own payments through the application. In addition, the application can be used with different retail stores. Eventually, the system offers a Website Dashboard service for the retailer side which enable them to monitor and view selling and stock information, which helps them run their stores more effectively.

สารบัญ

บทคัดย่อ	๑
Abstract	๑
สารบัญ	๒
สารบัญรูป	๒
สารบัญตาราง	๒
1 บทนำ	๑
1.1 ที่มาของโครงการ	๑
1.2 วัตถุประสงค์ของโครงการ	๑
1.3 ขอบเขตของโครงการ	๑
1.4 ประโยชน์ที่ได้รับ	๒
1.5 เทคโนโลยีและเครื่องมือที่ใช้	๒
1.6 แผนการดำเนินงาน	๒
1.7 บทบาทและความรับผิดชอบ	๓
1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม	๓
2 ทฤษฎีที่เกี่ยวข้อง	๔
2.1 WebSocket for Streaming image [7]	๔
2.2 Artificial neural networks [1]	๔
2.2.1 Multilayer perceptron [5]	๕
2.2.2 Multilayer perceptron - Feed forward [4]	๕
2.2.3 Classification	๖
2.2.4 Training	๖
2.3 Convolutional neural network	๗
2.4 Transfer learning	๘
3 โครงสร้างและขั้นตอนการทำงาน	๑๐
3.1 โครงสร้างของระบบ	๑๐
3.1.1 แผนภาพกระแสข้อมูล (Data Flow Diagram)	๑๑
3.2 เตรียมชุดข้อมูลฝึกสอน	๑๑
3.3 K-Fold Cross Validation	๑๒
3.4 Data augmentation การเพิ่ม traing data	๑๓
3.4.1 Model Architecture	๑๔
3.5 classification products	๑๕
3.6 การพัฒนา Mobile Application	๑๖
3.6.1 Requirement Specification	๑๖
3.7 การเตรียมฐานข้อมูล	๑๖
3.7.1 ฐานข้อมูลของระบบ	๑๖
3.7.2 ฐานข้อมูลในแต่ละร้านค้า	๑๖
3.8 การพัฒนา Website Dashboard	๑๖
3.9 การทดสอบการทำงานของซอฟแวร์	๑๗
3.9.1 Unit testing	๑๗
3.9.2 Integration testing	๑๘
3.9.3 System testing	๑๘
3.9.4 Acceptance testing	๑๘

4 การทดลองและผลลัพธ์	19
4.1 Classification model	19
4.2 Application UX/UI	20
4.3 Web Dashboard UX/UI	24
5 บทสรุปและข้อเสนอแนะ	26
5.1 สรุปผล	26
5.2 ปัญหาที่พบและแนวทางการแก้ไข	26
5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ	26
บรรณานุกรม	27
ก Data Flow	29

สารบัญ

2.1	WebSocket structure	4
2.2	Multilayer perceptron	5
2.3	Convolution	7
2.4	Convolutional Neural Network	8
2.5	The concept of transfer learning	8
3.1	Overall project structure	10
3.2	Data Flow Diagram	11
3.3	Google Drive	11
3.4	Dataset	12
3.5	Dataset generator	13
3.6	Mobile Application Database Schema	17
3.7	Store Website Dashboard Database Schema	18
4.1	Train fold	19
4.2	Confusion matrix	20
n.1	Work table	31

สารบัญตาราง

1.1 Planning	2
------------------------	---

บทที่ 1

บทนำ

1.1 ที่มาของโครงการ

จากระบบการขายสินค้าในร้านค้าปลีกส่วนใหญ่ของประเทศไทย ไม่ว่าจะเป็นห้างสรรพสินค้า ชุปเปอร์มาร์เก็ต หรือ ร้านค้าปลีกรายย่อยต่าง ๆ พบร่วมกันการที่ใช้ในการชำระเงิน คือ การชำระสินค้าที่ดำเนินการชำระเงินโดยมีพนักงานบริการ ซึ่งข้อเสียแรกของวิธีการชำระเงินดังกล่าว คือ การรอชำระเงินค้าที่แคชเชียร์นั้น เสียเวลา และไม่สะดวกรวดเร็ว ยิ่งหากต้องมีการต่อแวรอชำระเงิน ก็จะทำให้ผู้ใช้บริการเสียเวลามากขึ้น และเสียความพึงพอใจในการใช้บริการ รวมถึงต้องมีการจัดที่นั่งที่สำหรับการต่อแวรอีกด้วย ด้วยเหตุนี้ จึงควร พัฒนาเทคโนโลยีที่มาช่วยการแก้ปัญหาอย่างตรงจุด โดยให้ลูกค้าสามารถชำระเงินได้ด้วยตนเอง ผ่านการเลือกซื้อสินค้าได้อย่างสะดวกสบายผ่านอุปกรณ์โทรศัพท์มือถือ

นอกจากนี้สำหรับร้านค้าปลีกรายย่อยที่มีรูปแบบการขายเป็นแบบบริการตนเอง หรือแม้กระทั่งร้านค้าปลีกที่มีพนักงานชำระเงิน แต่ไม่มีระบบช่วยจัดการยอดขาย ก็สามารถพบปัญหาในการจัดการยอดขาย และคลังสินค้าได้ เนื่องจากพนักงาน หรือเจ้าของร้านต้องติดตามการขายสินค้าด้วยตนเองทั้งหมด โดยจะต้องคอยนับจำนวนสินค้าที่เหลืออยู่ภายในร้านเพื่อตรวจสอบว่าได้ขายอะไรไปแล้ว ทำให้เกิดความยากลำบาก และผิดพลาดได้ง่าย

ผู้จัดทำจึงพัฒนาระบบ CapSnap self-service เพื่อลดปัญหาที่เกิดจากการรอชำระเงินที่แคชเชียร์ โดยพัฒนาแอปพลิเคชันสำหรับผู้ใช้งานที่ว่าไปที่สามารถเชื่อมต่อกับร้านค้าที่เข้าร่วมเพื่อใช้บริการการซื้อสินค้าแบบ self-service ที่ร้านค้านั้น ๆ โดยลูกค้าสามารถเข้าใช้งานแอปพลิเคชันในโทรศัพท์มือถือแล้วทำการเลือกซื้อสินค้าผ่านฟังก์ชันการสตรีมมิ่งภาพสินค้าเพื่อ ทราบถึงรายละเอียดของสินค้า และเพิ่มสินค้าเข้าตะกร้า โดยระบบจะใช้หลักการทำงาน Computational Intelligence ในการแยกแยะสินค้า จากการสตรีมมิ่ง ลูกค้าสามารถชำระเงินได้ด้วยตนเองพร้อมนำสินค้าที่ซื้อออกจากร้านค้าได้เลย ซึ่งระบบจะเก็บบันทึกประวัติการขายสินค้าลง ในฐานข้อมูลเพื่อแสดงให้กับผู้ใช้งานสามารถจัดการคลังสินค้า และข้อมูลการขายได้ในที่เดียว เพิ่มประสบการณ์การใช้บริการที่ดีให้ลูกค้า และเพิ่มประสิทธิภาพในการบริหารจัดการสินค้าในร้านค้าสำหรับร้านค้าที่ต้องการให้บริการแบบ self-service ลดการว่าจ้างพนักงาน และต้องการเครื่องมือในการบริหารร้านค้า

1.2 วัตถุประสงค์ของโครงการ

- เพื่อพัฒนาระบบ แยกแยะชนิดสินค้าโดยใช้ Computational Intelligence ได้
- เพื่อสร้างเว็บไซต์และแอปโทรศัพท์เพื่อให้ผู้ใช้งานสามารถใช้งานระบบได้

1.3 ขอบเขตของโครงการ

ข้อมูลที่ใช้ในการฝึกอบรม Computational Intelligence ในการแยกแยะชนิดสินค้า เป็นชุดข้อมูลที่เก็บจากร้านค้าห้อง 422 ตึก 30 ปี คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่

1.4 ประโยชน์ที่ได้รับ

- แอปพลิเคชันในโทรศัพท์มือถือที่ลูกค้าสามารถแยกแยะชนิดสินค้าเพื่อให้ลูกค้าสามารถจ่ายเงินให้กับสินค้านั้นๆได้ด้วยตัวเอง โดยไม่ต้องคอยเลือกชนิดของสินค้านั้นๆ
- เว็บไซต์ที่แสดงจำนวนการขายของสินค้าแต่ละชนิดเพื่อให้ร้านค้าสามารถจัดการสินค้าได้

1.5 เทคโนโลยีและเครื่องมือที่ใช้

- Python และ Aiortc : สำหรับพัฒนาในส่วนของ Backend การรับข้อมูลstromมิ่งจากแอปพลิเคชัน การฝึกสอนโมเดล และการ classify Product โดยไม่ต้องคอยเลือกชนิดของสินค้านั้นๆ
- Flask และ Next.js: สำหรับการพัฒนา Frontend ในส่วนเว็บไซต์ของร้านค้า (Website Dashboard)
- Flutter และ WebSocket: สำหรับพัฒนา Frontend ในส่วน Application ในโทรศัพท์มือถือของลูกค้า และใช้ WebSocket ในการสตีมมิ่งภาพจากกล้องโทรศัพท์มือถือไปยังส่วน Backend เพื่อทำการประมวลผล
- Supabase: สำหรับเก็บฐานข้อมูลทั้งหมดที่แสดงผล

1.6 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ม.ค. 2566	ก.พ. 2566	มี.ค. 2566	เม.ย. 2566	พ.ค. 2566	มิ.ย. 2566	ก.ค. 2566	ส.ค. 2566	ก.ย. 2566	ต.ค. 2566	พ.ย. 2566	ธ.ค. 2566	ม.ค. 2567	ก.พ. 2567	มี.ค. 2567	Progress
Planning																70%
Document																70%
Collect data																70%
Back-end development																60%
App development																20%
Dashboard development																10%
Payment development																0%
Testing																0%

ตารางที่ 1.1: Planning

1.7 บทบาทและความรับผิดชอบ

นายพงศกร รัตนพันธ์ รหัส 630610749 รับผิดชอบในส่วนของ Backend ดังนี้

1. การรับข้อมูลสตรีมมิ่งจากระหว่าง Mobile Application และ Classifier ผ่าน WebSocket
 2. การฝึกสอนโมเดลเพื่อทำ Product classification
 3. การจัดเก็บฐานข้อมูลข้อมูลรูปภาพของสินค้าเพื่อทำการฝึกสอนโมเดล

นางสาวศุภาริญา ศิลป์สิทธิ์ รหัส 630610765 รับผิดชอบการพัฒนาส่วน Frontend และ ฐานข้อมูลบน Supabase ดังนี้

1. หน้าเว็บไซต์ของร้าน (website dashboard)
 2. Mobile Application
 3. จัดการการเก็บฐานข้อมูลทั้งหมดที่ใช้ในการแสดงผล

1.8 ผลกระทบด้านสังคม สุขภาพ ความปลอดภัย กฎหมาย และวัฒนธรรม

โครงการนี้ลดความซับซ้อนและเวลาที่ลูกค้าจะต้องรอต่อแถวเพื่อจ่ายเงินของสินค้า และ ช่วยให้สังคมสามารถเข้าถึงเทคโนโลยีที่ก้าวหน้าไปจากเดิมในการใช้กิจวัตรประจำวันอย่างการซื้อสินค้า โดยเปลี่ยนมาใช้การบริการตนเองผ่านแอปพลิเคชันที่อำนวยความสะดวกต่อผู้คน มือถือที่ใช้งานกันอย่างแพร่หลาย ทำให้สังคมคุ้มกันความทันสมัย และความสะดวกสบายมากขึ้น ตอบโจทย์ความต้องการ และรูปแบบการใช้ชีวิตของผู้คนในยุคสมัยใหม่ ช่วยให้เกิดการยืนรองรับสินค้า หรือการใช้สายตาในการหาข้อมูลสินค้า และพัฒนาสุขภาพจิตจากประสบการณ์การซื้อสินค้าที่ดีขึ้น

อีกทั้งยังเป็นอักขระเลือกหนึ่งในการที่ร้านค้าจะมาใช้ระบบ self-service ที่มีการจัดการที่ดี เจ้าของกิจ-การร้านค้าปลีกสามารถจัดการบริหารร้านค้าได้สะดวก และมีประสิทธิภาพมากขึ้น ทำให้เกิดความคุ้นเคยกับวัฒนธรรมการซื้อของแบบบริการตนเองในสังคมประเทศไทยมากขึ้น รวมถึงเป็นต้นแบบในการพัฒนาโครงสร้างในลักษณะเดียวกันเพื่อความก้าวหน้าทางเทคโนโลยีในประเทศต่อไป

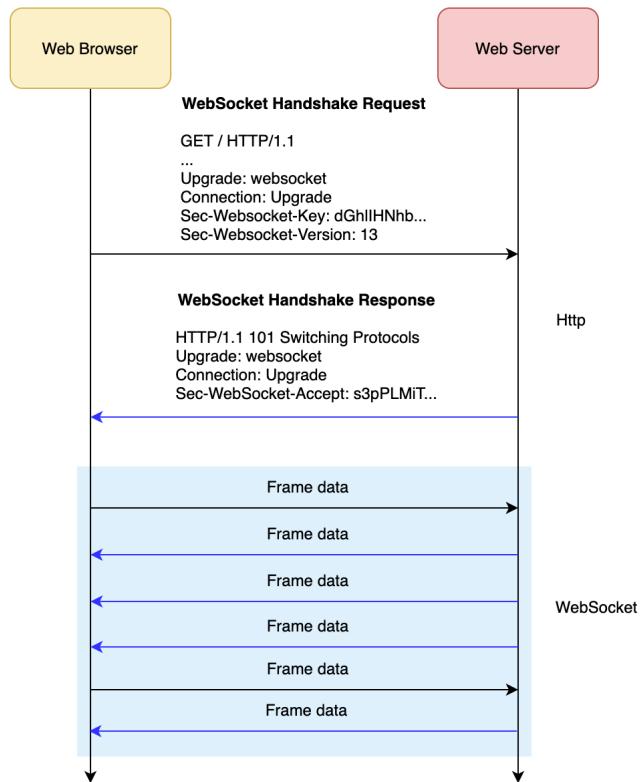
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

โครงงานนี้ได้นำองค์ความรู้ในด้านของ Computational Intelligence และ การ streaming รูปภาพจาก application ไปยัง backend ผ่าน WebSocket เพื่อให้ backend ที่เป็น Computational Intelligence ทำการ classification products

2.1 WebSocket for Streaming image [7]

WebSocket (computer communications protocol) 2.1



รูปที่ 2.1: WebSocket structure

2.2 Artificial neural networks [1]

เป็นแขนงหนึ่งของ Computational Intelligence ซึ่งได้รับแรงบรรดาลใจมาจากการทำงานของสมองของมนุษย์ ซึ่งประกอบด้วยหน่วยเล็กๆ เรียกว่า cell Neuron แต่ละ Neuron ก็จะเชื่อมต่อโยงใยกันด้วยเส้นประสาทเรียกว่า ไซแนปส์ (Synapse) เพื่อส่งสัญญาณไฟฟ้า ที่เกิดจากสิ่งเร้าต่างๆ ว่าจะตอบสนองต่อสิ่งเร้า

นั่นอย่างไร โดยแต่ละ Neuron จะได้รับ Input หลาย ๆ อัน จากกิ่งก้านสาขาของ Dendrite และนำมาระมวลผล ออกมานเป็น 1 Output ออกไปที่ Axon เพื่อส่งต่อไปให้ Dendrite ของ Neuron อื่น ๆ ใช้เป็น Input ต่อไป

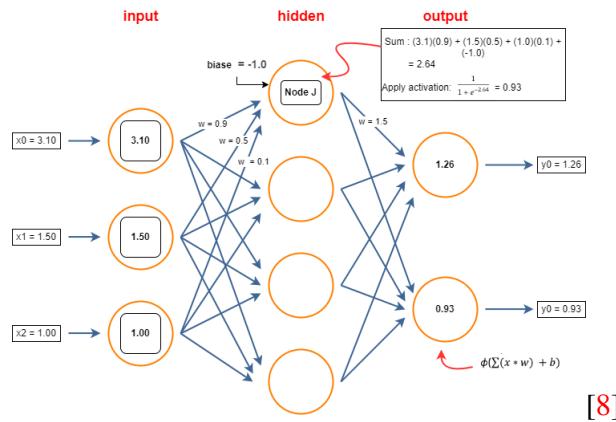
เมื่อมนุษย์เดินโดยขึ้น หาก Neuron ในนั้นตอบสนองต่อสิ่งเร้าประเภทไหนได้ดี ก็จะสามารถส่งสัญญาณไฟฟ้าได้แรงมากกว่า Neuron อื่นๆ เมื่อ Neuron หลายอันต่อกันหลาย ๆ layer ก็จะกลายเป็น Neuron network ของมนุษย์

โดยในทาง Computational Intelligence จะใช้ node เป็นตัวแทนของ Neuron โดยจะเรียกเป็นชั้นๆ (layer) โดยสัญญาณที่ส่งออกจากแต่ละ node จะมี weight ที่กำหนดความแรงของสัญญาณนั้นๆ เมื่อมีหลาย ๆ node และต่อกันหลาย ๆ layers ก็จะกลายเป็น neural network ขนาดใหญ่ เรียกว่า deep learning ซึ่งเลียนแบบการการทำงานของสมองมนุษย์ ทำให้ computers สามารถ process ข้อมูลในลักษณะเดียวกับที่ สมองของมนุษย์ทำการประมวลผลข้อมูล

2.2.1 Multilayer perceptron [5]

เป็นส่วนพื้นฐานของ neural network เป็นการที่ node ในแต่ละ layer เชื่อมต่อกับ ทุก node ของ layer+1 (fully connected) โดยทุกเส้นการเชื่อมต่อของ $node_i$ กับ $node_j$ จะมี weight w_{ji} ซึ่งเป็นความแรงของสัญญาณอยู่

โดยจะมี Input layer สำหรับข้อมูล (สิ่งเร้า) hidden layer ในการตัดสินใจ และ output layer ในการเลือกการกระทำกับสิ่งเร้านั้นๆ



รูปที่ 2.2: Multilayer perceptron

2.2.2 Multilayer perceptron - Feed forward [4]

โดยในแต่ละ node ที่ไม่ใช่ Input layer จะรับค่าผลรวมจาก node ก่อนหน้า เป็นผลรวมจากทุก Input (ทุก Dendrite) ของ node นั้นๆ

โดยสมการผลรวม Input ของ $node_j$

$$node_j = v_j = \sum (\forall w_{ji}) + biase \quad (2.1)$$

Activation function [2] คือ พังก์ชันที่รับ ผลรวมจากทุก Input (v_j) และคำนวณว่าจะส่งต่อเป็น Output เท่าไร ซึ่งมี function มากมายที่นิยมใช้ทำ Activation function ตัวอย่างเช่น

- Linear activation: $\phi(\mathbf{v}) = a + \mathbf{v}'\mathbf{b}$,
- ReLU activation: $\phi(\mathbf{v}) = \max(0, a + \mathbf{v}'\mathbf{b})$,
- Heaviside activation: $\phi(\mathbf{v}) = 1_{a+\mathbf{v}'\mathbf{b}>0}$,
- Logistic activation: $\phi(\mathbf{v}) = (1 + \exp(-a - \mathbf{v}'\mathbf{b}))^{-1}$.

โดยในทุกๆ $node_j$ จะมี Activation function สำหรับคำนวณค่า output ที่จะส่งต่อไปยัง layer ถัดไป

$$output_{node_j} = Y_j = y(v_i) \quad (2.2)$$

ตัวอย่าง ReLU Activation function ค่าที่ส่งออกไปยัง layer ถัดไปจะเป็น

$$y(v_i) = \max(0, v_i) \quad (2.3)$$

2.2.3 Classification

จาก Input layer ข้อมูลจะถูกส่งต่อไปยัง Layer ถัดๆไป เรียกว่า Feed forward และเมื่อถึง Layer สุดท้ายของ Neural Network เรียกว่า Output layer โดยส่วนใหญ่ใน output layer นี้จะมีจำนวนของ node เท่ากับจำนวนของ class ของข้อมูลที่ทำการ classification โดยแต่ละ node ใน Output layer จะเป็นตัวแทนของ class ซึ่ง หาก node ได้ให้ค่า Output node เยอะที่สุด Input data ก็ถูก classify เป็น class ของ node นั้นๆ

2.2.4 Training

คือการฝึกสอนโดยใช้ dataset เปลี่ยน weight ในแต่ละส่วนของ MLP เพื่อให้ตอบสนองต่อ Input ให้ใกล้เคียงกับสิ่งที่ควรจะเป็นมากขึ้น โดยหนึ่งในวิธีการเปลี่ยน weight คือ backpropagation

โดยค่า error ที่ได้มาจากการ output layer กับ ค่าที่ควรได้จาก Input ที่ใส่เข้าไป เพื่อเปลี่ยน weight ให้ค่า error มีค่าน้อยลงกว่าเดิม

$$output_{node_j} = Y_j = y(v_i) \quad (2.4)$$

สามารถคำนวณหาค่า error ในแต่ละ node ของ output layer ได้โดย

$$e_j(n) = d_j(n) - y_j(n) \quad (2.5)$$

where $d_j(n)$ is the desired target value , and $y_j(n)$ value produced by the perceptron at $node_j$

$$\mathcal{E}(n) = \frac{1}{2} \sum_{\text{output node } j} e_j^2(n). \quad (2.6)$$

และใช้ gradient descent ในการเปลี่ยน weight ในแต่ละเส้น

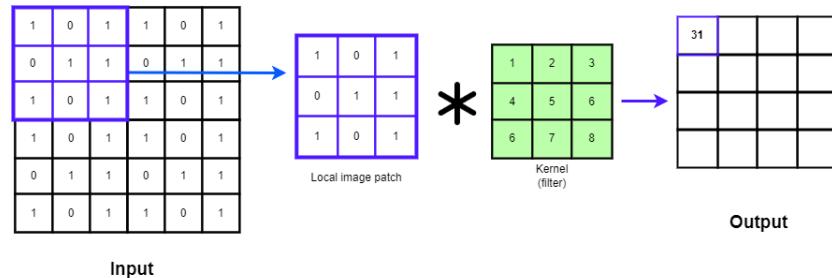
$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \quad (2.7)$$

2.3 Convolutional neural network

เป็น Computational intelligence ที่อุปแบบมาในงานด้านรูปภาพ ใช้การคำนวณการที่เรียกว่า ”convolution” (สัญลักษณ์ *) เป็นการดำเนินการทางคณิตศาสตร์ ระหว่างสองฟังก์ชัน $f * g$ เพื่อถูกการเปลี่ยนแปลงของฟังก์ชัน (f) เมื่อมีฟังก์ชัน (g) เข้ามา ใช้ในการเพื่อประมวลผลข้อมูลภาพ

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau. \quad (2.8)$$

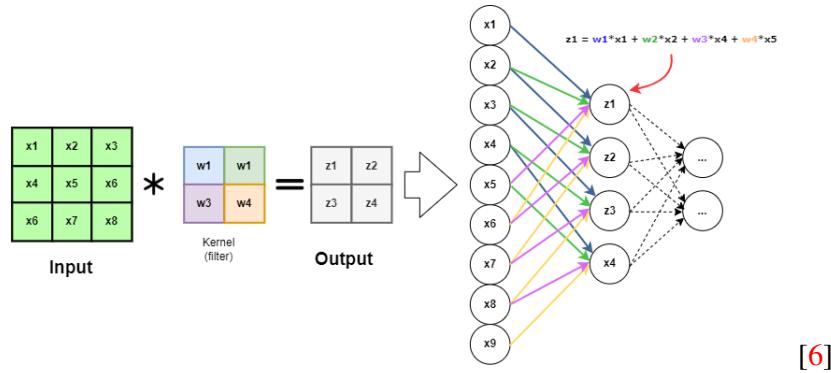
โดยจะแบ่งส่วนของภาพเป็นส่วนย่อยๆ ไปทั่วทั้งรูปภาพ และนำมายัง convolution กับ ”filters” หรือที่เรียกว่า ”kernels” เพื่อหา patterns ของรูปภาพ ที่เรียกว่า feature map



รูปที่ 2.3: Convolution

CNNs ต่างจาก Neural Networks อื่นๆ ตรงที่ shared-weight ร่วมกัน ซึ่งทำให้มีความสามารถในการแยกแยะ patterns ได้ดี

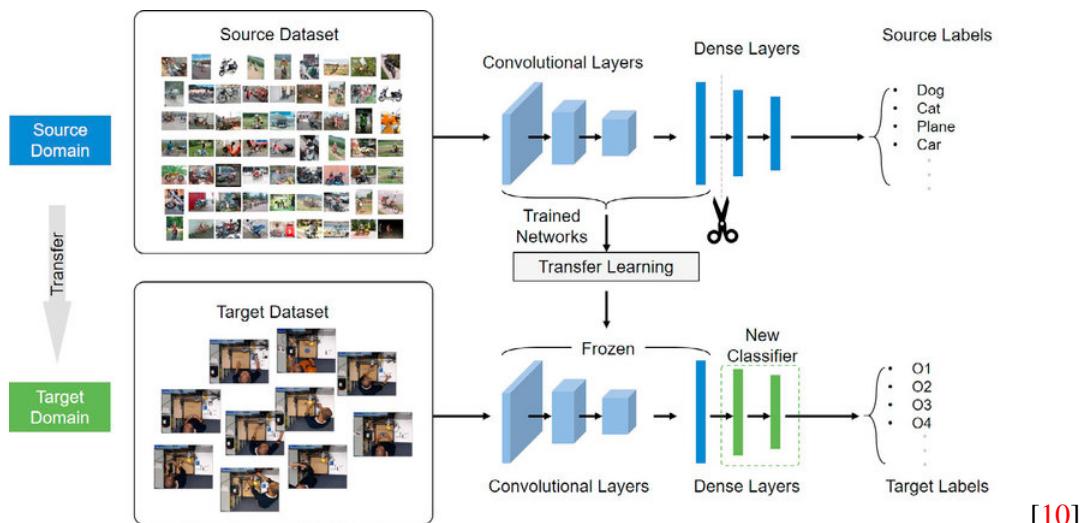
หากเขียนในโครงสร้าง Neural Network จะได้ดังรูป 2.4



รูปที่ 2.4: Convolutional Neural Network

2.4 Transfer learning

เป็นเทคนิคที่นำโมเดล ที่ผ่านการฝึกฝนจนแก้ปัญหา ในงานอื่นๆที่มีความคล้ายคลึงกัน นำมาเป็นโมเดลตั้งต้น สำหรับโมเดลในการแก้ปัญหาใหม่ๆ ตัวอย่างเช่น โมเดลที่ได้รับการฝึกฝนให้จดจำวัตถุในภาพสามารถใช้ เพื่อรับบุตถุที่คล้ายกันในภาพต่างๆ ได้ แม้ว่าภาพใหม่จะมีสภาพแสงหรือพื้นหลังต่างกันก็ตาม กุญแจสำคัญคือการระบุคุณสมบัติทั่วไปหรือการเป็นตัวแทนที่ใช่วร์มกันระหว่างโดเมนต้นทางและโดเมนเป้าหมาย วิธี Transfer Learning ที่ได้รับความนิยมมากที่สุดวิธีหนึ่งคือ fine-tuning คือการใช้โมเดลที่ผ่านการ pre-trained มาแล้ว นำมา train ต่อบน ชุดข้อมูลใหม่ และใช้ learning rate น้อยๆ เพื่อป้องกัน weight ที่เคยผ่านการฝึกสอนจนมีความแม่นยำเปลี่ยนแปลงไปมาก จนไม่มีความแม่นยำ อีกวิธีหนึ่งคือ feature extraction ซึ่งเกี่ยวข้องกับการใช้ โมเดลที่ผ่านการ pre-trained เป็นตัวแยกคุณลักษณะของ ข้อมูล และสร้าง โมเดลใหม่เพื่อ train จากคุณลักษณะเหล่านี้ที่โมเดลตั้งต้นแบ่งแยกออกมาได้



รูปที่ 2.5: The concept of transfer learning

Transfer learning สามารถอธิบายได้ในรูปของ domains และ tasks.

โดย domain \mathcal{D} ประกอบไปด้วย: feature space \mathcal{X} และ marginal probability distribution

$P(X)$, ซึ่ง $X = \{x_1, \dots, x_n\} \in \mathcal{X}$.

จะได้ว่าหากมี domain ใด, $\mathcal{D} = \{\mathcal{X}, P(X)\}$, หน้าที่ของ domain นั้นจะมี 2 ส่วนคือการ label space \mathcal{Y} และเป็น objective predictive function $f : \mathcal{X} \rightarrow \mathcal{Y}$.

ซึ่งฟังก์ชัน f จะถูกใช้ในการ ทำนาย ข้อมูลที่มีความสอดคล้องกันของ $f(x)$ ของข้อมูลใหม่ x . โดยหน้าที่ของ domain ,โดยเขียนแทนด้วยสัญลักษณ์ $\mathcal{T} = \{\mathcal{Y}, f(x)\}$, ทำการเรียนรู้จากชุดข้อมูลฝึกสอนที่เป็นคู่อันดับ $\{x_i, y_i\}$, โดยที่ $x_i \in X$ และ $y_i \in Y$.

หากมี domain ตั้งต้น \mathcal{D}_S และ learning task ตั้งต้น \mathcal{T}_S , target domain \mathcal{D}_T และ learning task เป้าหมาย \mathcal{T}_T โดยที่ $\mathcal{D}_S \neq \mathcal{D}_T$, หรือ $\mathcal{T}_S \neq \mathcal{T}_T$,

Transfer learning นั้นช่วยในการพัฒนาการเรียนรู้ของ target predictive function ที่ต้องการ $f_T(\cdot)$ ใน \mathcal{D}_T จากการใช้ความรู้ที่ได้รับจาก \mathcal{D}_S และ \mathcal{T}_S

บทที่ 3

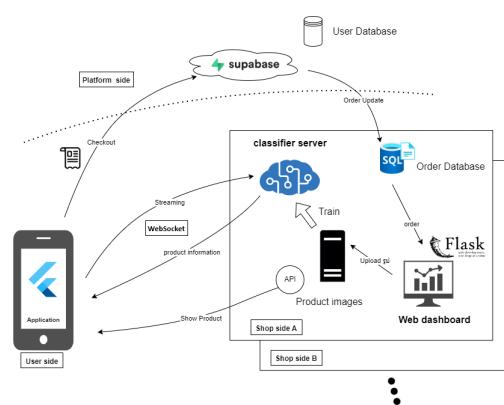
โครงสร้างและขั้นตอนการทำงาน

ในบทนี้ จะกล่าวถึงหลักการ, การนำทฤษฎีที่เกี่ยวข้องมาประยุกต์ใช้ และการออกแบบของระบบ

3.1 โครงสร้างของระบบ

Application โทรศัพท์มือถือ จะสามารถ classify สินค้าได้โดย application จะการถ่ายภาพของสินค้า และทำการ streaming ผ่าน WebSocket ไปยัง server ที่มี Computational Intelligence ทำหน้าที่ classification ข้อมูลที่ได้รับเข้ามา และตอบกับข้อมูลชนิดสินค้าและราคา กลับไปยัง application ผู้ใช้งาน application จะสามารถเพิ่มสินค้าที่ application ถ่าย ไปยังตะกร้าสินค้า และสามารถ กดจ่ายเงินซื้อสินค้า ได้

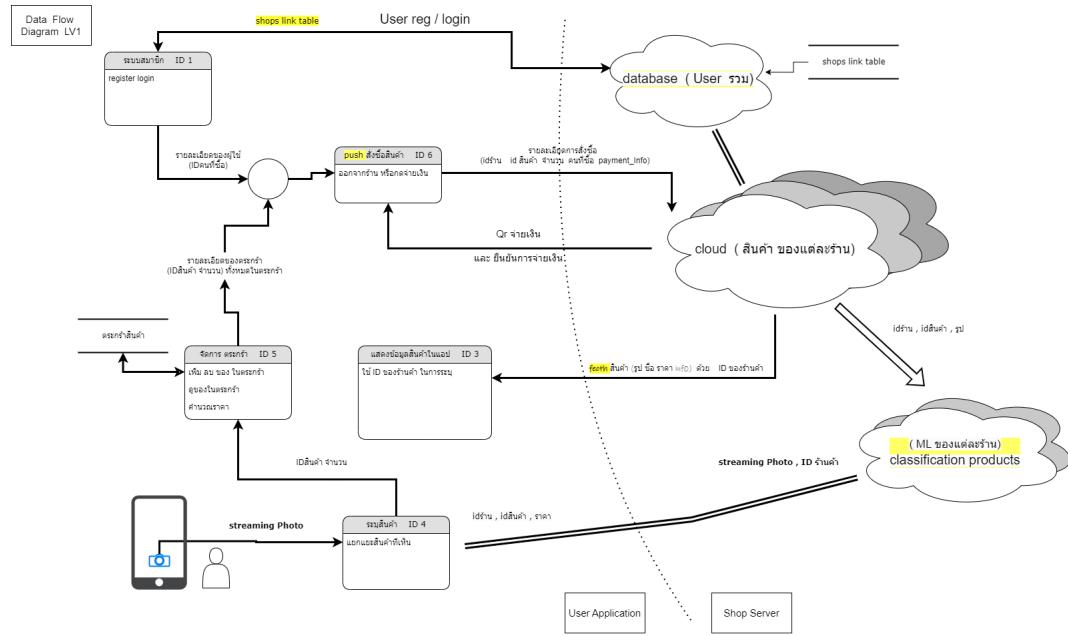
โดยร้านค้าจะการ บันทึกประวัติการซื้อขาย ของสินค้าภายในร้านทำให้สามารถจัดการ จำนวนของสินค้าภายในร้าน โครงสร้างของระบบ CapSnap จะเป็นดังรูป 3.1



รูปที่ 3.1: Overall project structure

3.1.1 แผนภาพกระแสข้อมูล (Data Flow Diagram)

จากขั้นตอนการทำงานของระบบ และการ Communication ของ application และ server ของร้านค้า จะมีโครงสร้างของ Data Flow Diagram ดังรูป 3.2



รูปที่ 3.2: Data Flow Diagram

3.2 เตรียมชุดข้อมูลฝึกสอน

ข้อมูลที่ใช้ในการ train model ใช้รูปภาพจากสินค้าสินค้าจากห้อง 422 ทำการจัดเก็บข้อมูลรูปภาพ โดยใช้กล้องมือถือ ในการถ่ายภาพโดยสินค้า 1 class (ชนิด) ทำการถ่ายภาพ 8 รูป ในมุมที่แตกต่างกัน โดยโมเดลในโครงงานนี้จะใช้สินค้าทั้งหมด 51 class และเพิ่มอีก 1 class สำหรับพื้นหลัง (background class) รวมเป็น 52 class จัดเก็บใน Google Drive ดังรูป 3.3 โดยทำการแยก รูปภาพตามหมวดหมู่ และทำการดึงข้อมูลมา train ผ่าน Google Colab โดยในแต่ละรูป จะเป็นข้อมูล Array ขนาด 224x224x3

```

directory/
  class_label_1/
    image1
    image2
  .
  .
  .
  class_label_2/
  .
  .
  .

```

รูปที่ 3.3: Google Drive

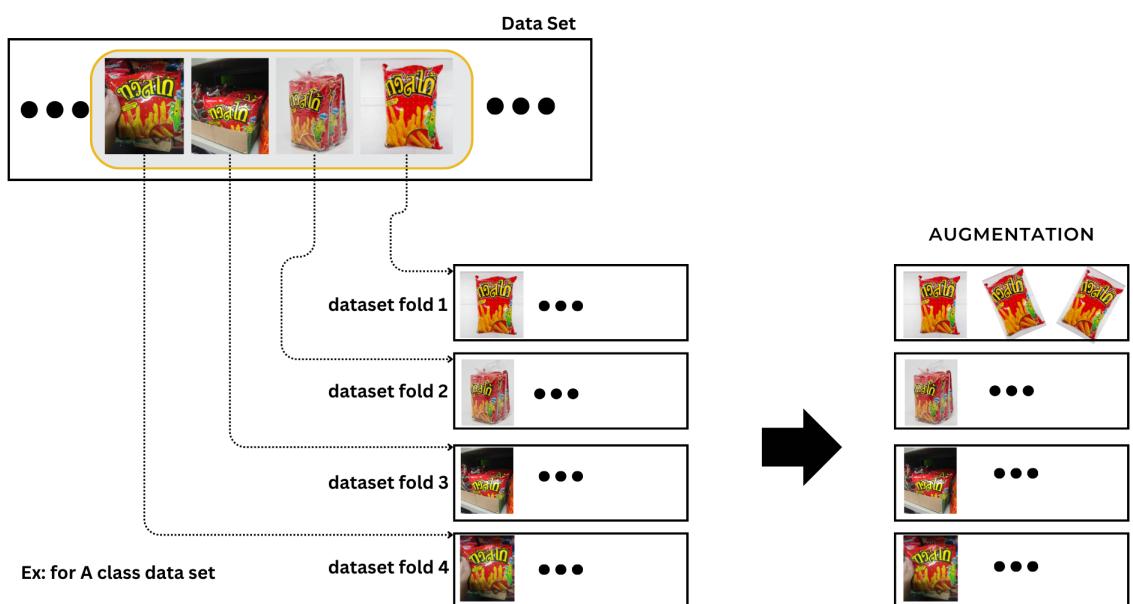
3.3 K-Fold Cross Validation

เพื่อตรวจสอบว่า model ที่เราทำการสร้าง ไม่ว่าจะเจอ dataset ที่ข้อมูลแตกต่างกันในแต่ละ fold ก็จะได้ accuracy ที่มีความใกล้เคียงในแต่ละ fold โดยที่ไม่แม่นยำแค่บาง dataset ซึ่งเป็นการ bias ต่อ dataset

4-fold validation (k=4)



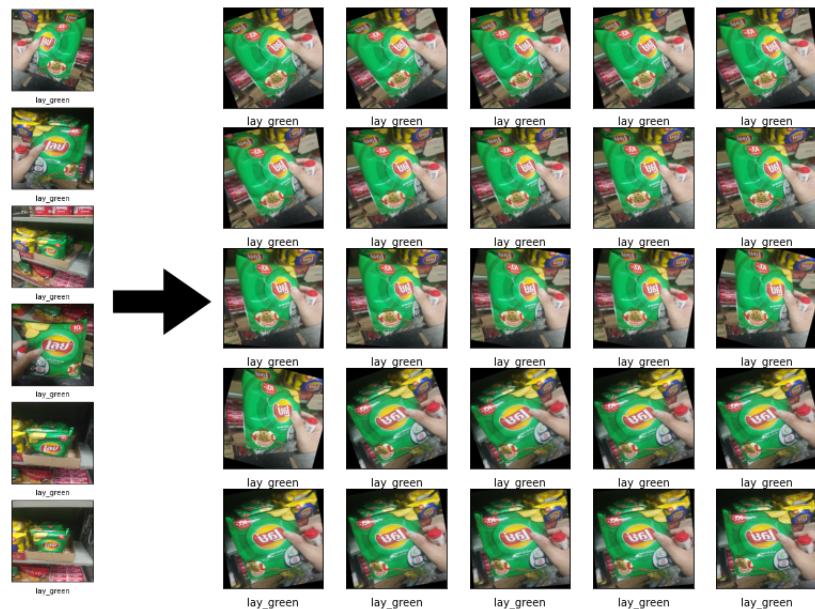
ซึ่งในโครงงานนี้ ได้ทำการแบ่ง dataset เป็น ทั้งหมด 4 fold ดังรูป 3.4 โดยใน dataset แต่ละ fold ก็ จะทำการ augmentation เพื่อเพิ่มจำนวนข้อมูลที่ใช้ในการ train model ของแต่ละ fold



รูปที่ 3.4: Dataset

3.4 Data augmentation การเพิ่ม traing data

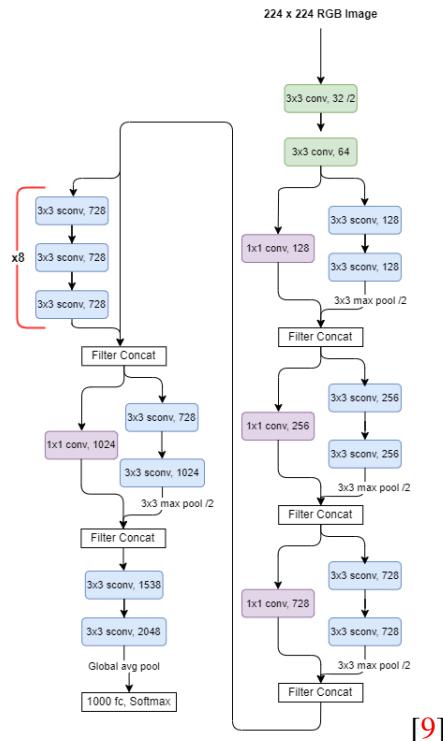
เพื่อให้มี train dataset ในจำนวนมาก โดยในแต่ละ 1 รูปภาพที่ถ่าย ซึ่งเป็นรูปภาพ RGB ขนาด 224x224 pixel นำไปสร้างเป็นรูปภาพใหม่ โดยใช้วิธีการ หมุน และ กลับด้าน ด้วยมุม -20,-15,-10,-5,0,5,10,15,20 องศา โดย 1 รูปภาพผ่านการ generate datasets จะกลายเป็น 18 รูปภาพซึ่งมีความแตกต่างกันเล็กน้อย ตามรูป 3.5



รูปที่ 3.5: Dataset generator

3.4.1 Model Architecture

โดยโมเดลในโครงงานนี้จะใช้ Xception pre-trained ซึ่งเป็น Convolutional Neural Network ซึ่งมีความลึก 71 layers. เป็นโมเดลที่ผ่านการจากรูปภาพต่าง ๆ มา กว่าล้านรูปภาพจาก ImageNet database . ซึ่งจะมีโครงสร้างดังรูป



โดยการนำ Xception model มาใช้ในการแยกคุณลักษณะเด่นของรูปภาพ และ สร้างโมเดลมาต่อท้ายเพื่อ เรียนรู้ลักษณะเด่นจากที่ Xception ทำการแยกออกมามาได้

Model: "model"		
Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
xception (Functional)	(None, 7, 7, 2048)	20861480
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 32)	3211296
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 16)	272
dense_3 (Dense)	(None, 36)	612

Total params: 24074188 (91.84 MB)
Trainable params: 3212788 (12.26 MB)
Non-trainable params: 20861480 (79.58 MB)

ซึ่ง model ที่จะใช้ก็จะเป็น Xception model ที่รับ Input ขนาด 224 x 224 x 3 ตามขนาดรูปภาพ 224x224 pixel และตาม layer RGB

โดย Xception เป็น model ตั้งต้นที่ผ่าน pre train model จะทำหน้าที่เป็น feature exercitation ลักษณะเด่นของรูปภาพ โดย Xception จะถูก lock ค่าของ weight เอาไว้เพื่อไม่ให้ค่าของ weight ที่ได้เรียนรู้ข้อมูลมาจาก ImageNet หายไป ซึ่ง exercitation มี weight ทั้งหมด 20 ล้าน parameter

และนำ layer มาต่อท้าย model ด้วย fully connected node ที่เรียกว่า Dense layer จำนวน 32 node และ 16 node และ layer สุดท้าย ซึ่งมีขนาดเท่ากับจำนวน class ของสินค้า ซึ่งทำหน้าที่เป็น classifier โดย Dense layer ซึ่งเชื่อมกันแบบ fully connected ที่สามารถเรียนรู้และปรับค่าของ weight ได้ในโครงสร้างทดสอบนี้จะมี weight ที่สามารถเรียนรู้ หรือ เปลี่ยนแปลงได้ 12 ล้าน paramiter

3.5 classification products

จากรูปภาพใน 52 class ผ่านการ generate datasets จะมี dataset ทั้งหมด 4432 sample ทำการแบ่งเป็น train 3546 sample และ 886 sample สำหรับการ evaluate โดยจาก train 3546 แบ่ง 50% สำหรับการ validation ในระหว่างการ train model

สำหรับเป็น service ในการ classify products ของ application ผ่าน aiortc และเว็บ WebSocket

3.6 การพัฒนา Mobile Application

ใช้ Flutter ในการสร้าง Mobile Application ในส่วนของผู้ใช้โดยใช้ service ของ WebSocket ติดต่อ กับ Classification server ที่ได้ทำการฝึกสอนไว้แล้ว โดยจะแอปพลิเคชันจะมีฟังก์ชันที่ทำการสตريمมิ่งภาพ สินค้าที่ลูกค้าถ่ายผ่าน WebSocket ไปยัง server ของทางร้านซึ่งมี Model classifier อยู่ จากนั้น Server ของทางร้านจะ Classify ว่าเป็นสินค้านิดใด และตอบกลับมาด้วยแอปพลิเคชันเพื่อแสดงผลให้กับผู้ใช้งาน

3.6.1 Requirement Specification

- ผู้คนทั่วไปสามารถลงทะเบียนเข้าใช้งานได้ผ่าน SSO อีเมล และหมายเลขโทรศัพท์มือถือ
- สามารถเชื่อมต่อกับข้อมูลร้านค้าได้ผ่านการสแกนคิวอาร์โค้ดเข้าใช้งานร้านค้า
- สามารถใช้กล้องโทรศัพท์มือถือสแกนสินค้าเพื่อสตريمภาพ Classification server เพื่อทำการดึงข้อมูลสินค้าจาก Server ของร้านค้ามาแสดงผลบนแอปพลิเคชัน
- สามารถเพิ่มหรือลดสินค้าในตะกร้าได้
- สามารถทำการชำระเงินได้ผ่าน Payment gateway ในตัวแอปพลิเคชัน
- สามารถ Checkout จากร้านค้าผ่านการสแกนคิวอาร์โค้ดออกจากร้านค้า

3.7 การเตรียมฐานข้อมูล

ในการเก็บฐานข้อมูลจะแบ่งเป็นฐานข้อมูลสองตัว โดยใช้ Supabase ในการสร้างโครงสร้าง และจัดฐานข้อมูล ทั้งหมดแบบ SQL ได้แก่

3.7.1 ฐานข้อมูลของระบบ

สำหรับเก็บข้อมูลร้านค้าที่เข้าร่วม และประวัติการซื้อสินค้าของผู้ใช้งานแอปพลิเคชันมือถือ ซึ่งมี Database schema ดังนี้

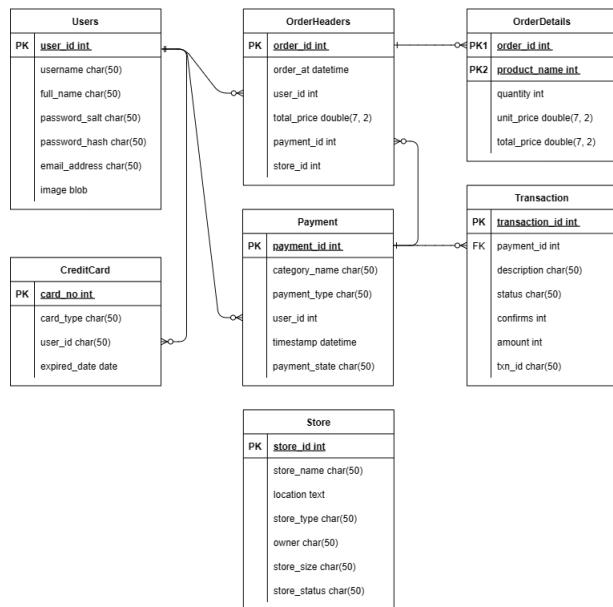
3.7.2 ฐานข้อมูลในแต่ละร้านค้า

สำหรับเก็บข้อมูลสินค้า และประวัติยอดขายของร้านค้า โดยร้านค้าแต่ละร้านจะมีฐานข้อมูลเป็นของตนเอง เพื่อใช้งานกับ Server ของร้านนั้น ๆ โดยตรง ซึ่งจะมี Database schema ดังนี้

3.8 การพัฒนา Website Dashboard

ออกแบบ UI/UX ของ Wedsite Dashboard และ Mobile Application ด้วย Figma หลังจากการออกแบบ และจัดการตั้งค่าฐานข้อมูลเสร็จแล้ว ก็จะพัฒนาในส่วนของเว็บไซต์ทางผู้ร้านค้าโดยใช้ Supabase ในการโฮสต์ เว็บไซต์ และเชื่อมต่อกับฐานข้อมูลที่ได้ตั้งค่าไว้ โดยใช้ Flask framework ในการจัดการ API และ Python ในการจัดการระบบ Backend ของเว็บไซต์โดยใช้ SQLAlchemy ในการดึงข้อมูลจากฐานข้อมูล Requirement Specification ดังนี้

Mobile Application Database Schema



รูปที่ 3.6: Mobile Application Database Schema

- สามารถเข้าใช้งานได้ผ่านการยืนยันตัวตนเป็น Administers เท่านั้น โดยสามารถมีได้ 1-5 คน
- สามารถดูคลังสินค้า และแก้ไขข้อมูลสินค้าในแต่ละชนิดได้แบบเรียลไทม์
- สามารถดูสถิติยอดขายสินค้าได้ทั้งแบบรายวัน รายเดือน และรายปี โดยแบ่งได้ 2 แบบ คือตามชนิด สินค้า และประเภทสินค้า
- สามารถดูประวัติการขายตามอเดอร์ของลูกค้าแต่ละคนได้ และดูข้อมูลของแต่ละออร์เดอร์ได้

3.9 การทดสอบการทำงานของซอฟแวร์

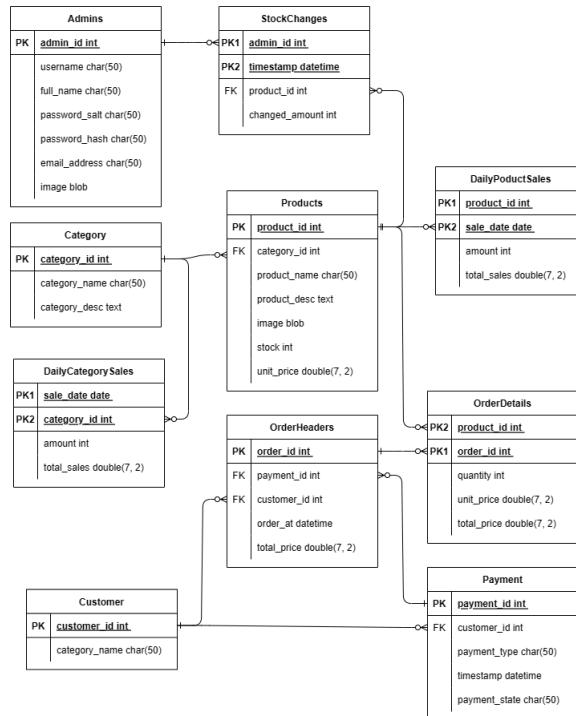
การทดสอบการทำงานของระบบ สามารถแบ่งเป็นขั้นตอนดังนี้

3.9.1 Unit testing

การทดสอบความถูกต้องของการทำงานในแต่ละฟังก์ชันหลักของระบบแยกกัน โดยยังไม่รวมแต่ละ Component เข้าด้วยกัน ซึ่งได้แก่

1. Classification system
2. Website dashboard
3. Mobile Application

Store Website Dashboard Database Schema



รูปที่ 3.7: Store Website Dashboard Database Schema

3.9.2 Integration testing

การทดสอบการทำงานเมื่อรวมระบบย่อยทั้งหมดเข้าด้วยกัน โดยหลัก ๆ จะทดสอบในเรื่อง API ว่ามีการรับส่งข้อมูลถูกต้องหรือไม่ และทำงานโดยรวมได้ถูกต้องทั้งหมดหรือไม่

3.9.3 System testing

การทดสอบระบบซึ่งแต่ละโมดูลข้างต้นจะถูกรวบ และทดสอบเป็นกลุ่ม เพื่อประเมินความสอดคล้องของระบบว่าทำงานได้ตามที่กำหนดไว้หรือไม่

3.9.4 Acceptance testing

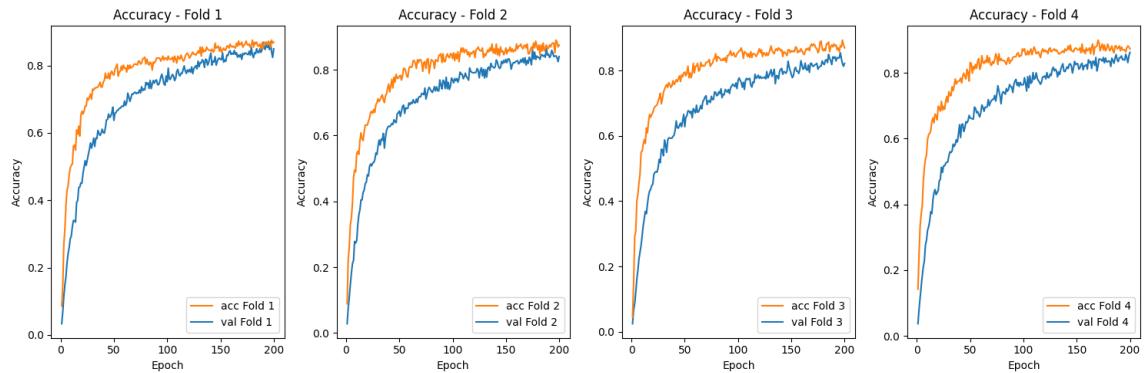
การทดสอบระบบโดยดูภาพรวมของการทำงาน ว่ามีการตอบสนองความต้องการของผู้ใช้งานในส่วนของฟังก์ชันการทำงาน และประสิทธิภาพการทำงาน ว่าสอดคล้องกับลักษณะของความต้องการของซอฟต์แวร์หรือไม่ โดยใช้การทดสอบแบบ Functional testing (Black box testing)

บทที่ 4

การทดลองและผลลัพธ์

4.1 Classification model

ผลลัพธ์ จากการ train & validation ด้วย 3546 sample เป็นจำนวน 200 Epoch
ตัวอย่าง ผลการทดลองของการ validation ด้วย 3546 sample



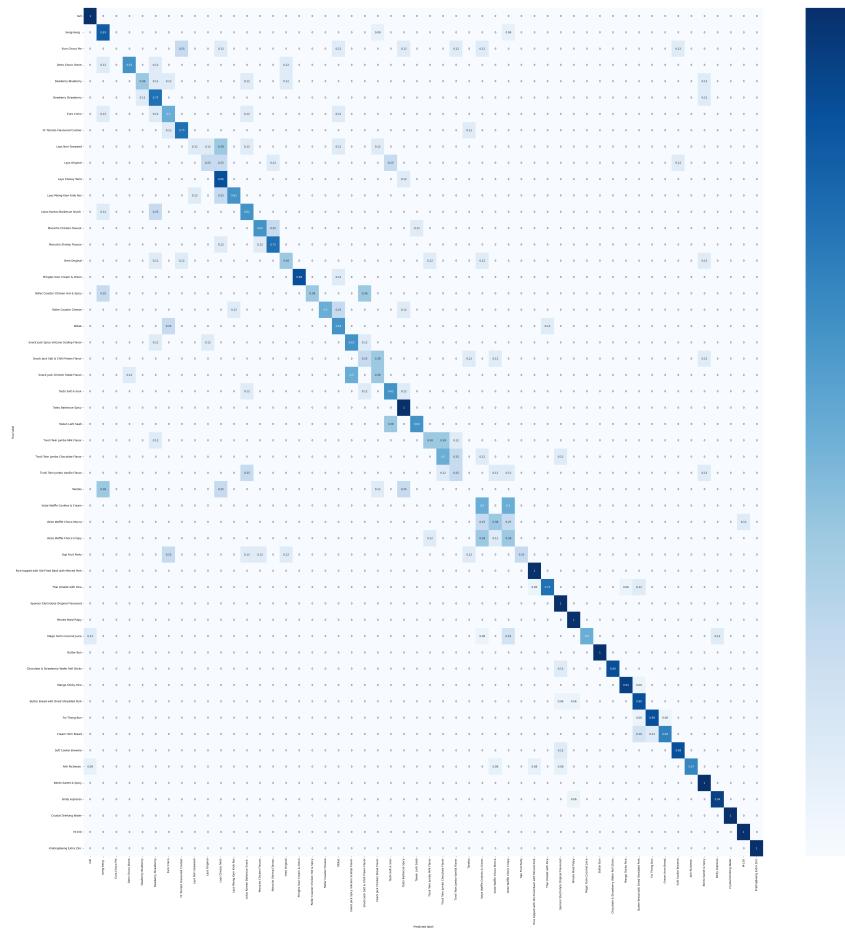
รูปที่ 4.1: Train fold

และทำการเลือก model ที่มีค่า Validation Score สูงสุดจากใน 4 model เพื่อนำไปทำการ Blind Test โดยได้นำ model ใน Fold 4 มาทำการ Evaluate (Blind Test)

$$\text{cross_entropy} = 1.175 \quad (4.1)$$

$$\text{accuracy} = 0.72 \quad (4.2)$$

Blind Test



รูปที่ 4.2: Confusion matrix

4.2 Application UX/UI

ในส่วนของการออกแบบส่วนสื่อประสานกับผู้ใช้ (GUI) ของแอปพลิเคชันมือถือนั้น ได้กำหนดให้มีหน้าการใช้งานหลักทั้งหมด 7 หน้า ได้แก่

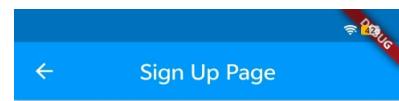
1. Login page: หน้าการเข้าสู่ระบบ
2. Register page: หน้าลงทะเบียนเข้าใช้งาน
3. Home page: หน้าแสดงชื่อผู้ใช้ และเมนูของฟังก์ชันต่าง ๆ
4. Streaming page: หน้าฟังก์ชันการสตรีมมิ่งรูปภาพสินค้าผ่านกล้องมือถือแบบเรียลไทม์ ซึ่งจะแสดงชื่อ และราคาของสินค้า ซึ่งมีปุ่มการทำดังนี้ ปุ่มกดดูรายละเอียดสินค้า , ปุ่มเพิ่มสินค้าลงตะกร้า , ปุ่มกดดูสินค้าในตะกร้า
5. Product Stock page: หน้าแสดงรายละเอียดของสินค้า
6. My cart page: หน้าแสดง และจัดการเพิ่ม-ลบสินค้าในตะกร้า
7. History page: หน้าแสดงประวัติการซื้อสินค้า



Email

Password

Login



Email

Password

Sign Up

New User? Create Account

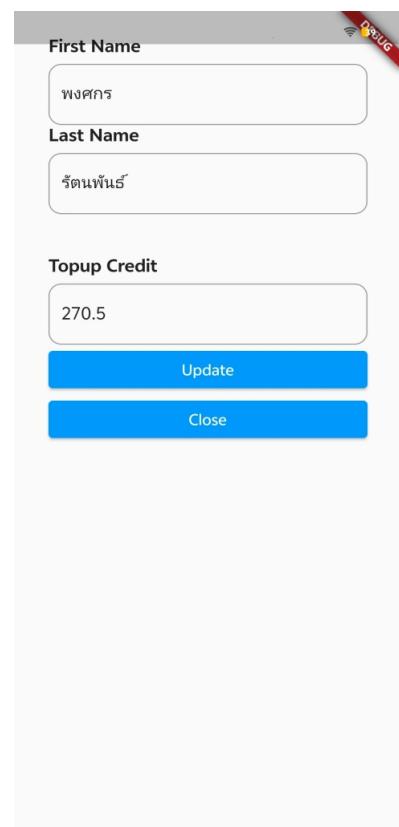
Already a User? Login

Login

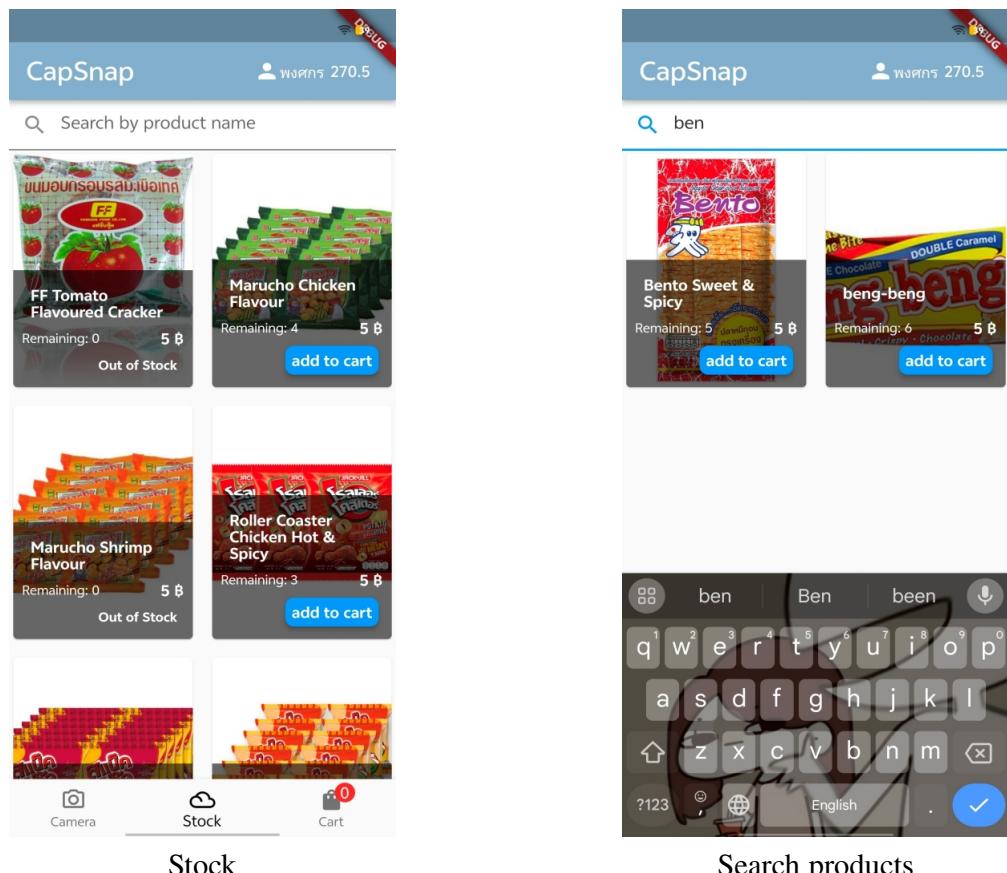


Profile

Signup



Edit profile

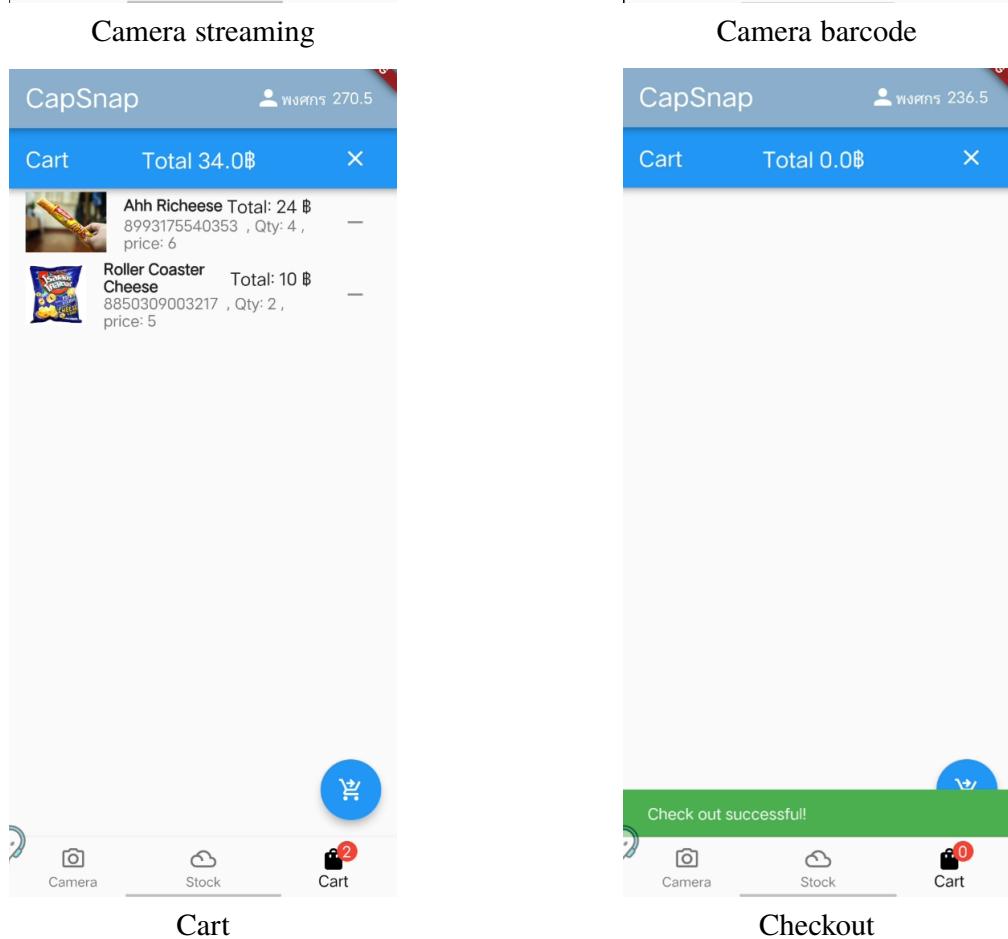
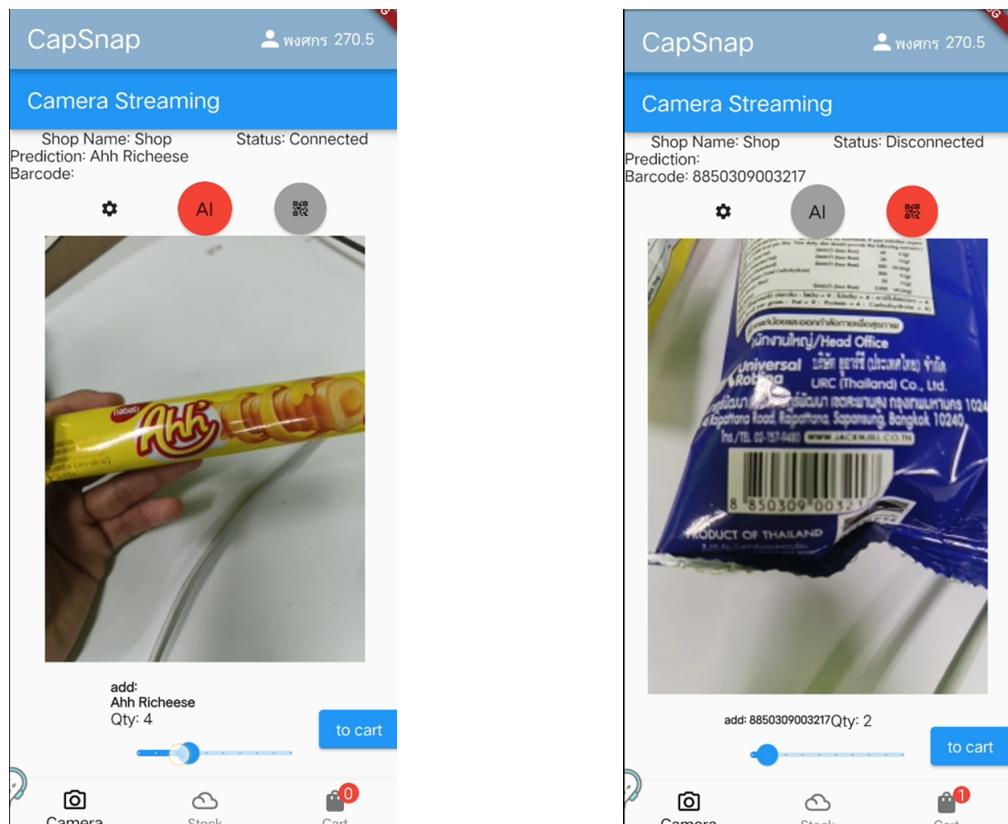


Stock

Search products

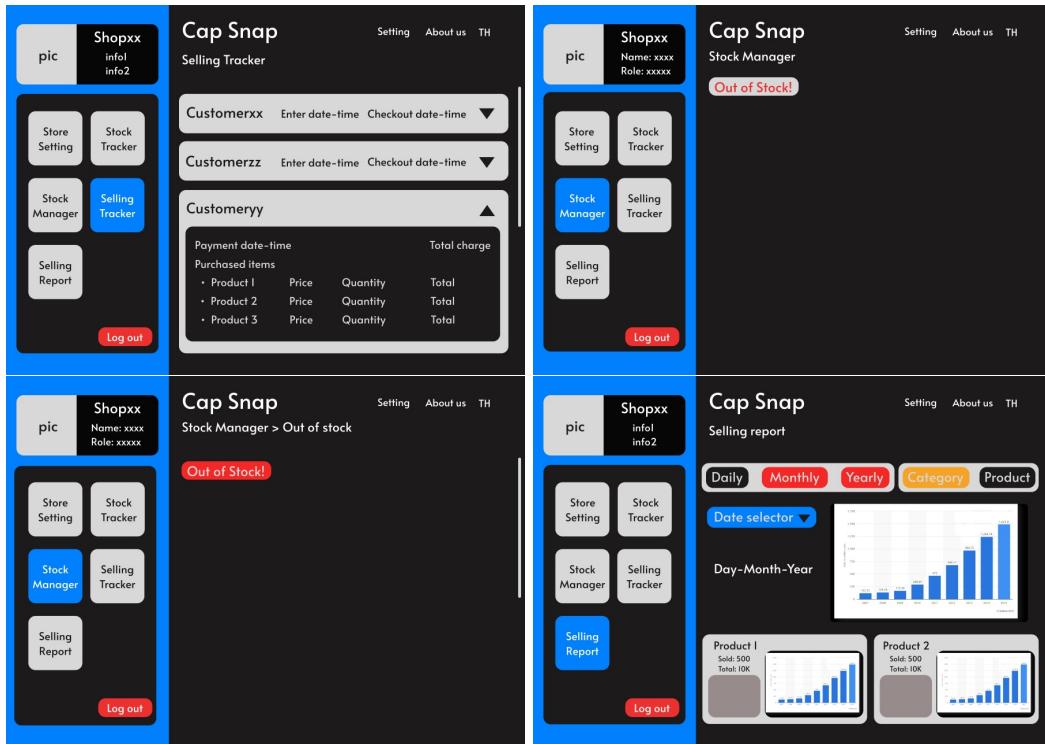
Product detail

Purchase history



4.3 Web Dashboard UX/UI





ในส่วนของการออกแบบส่วนสื่อประสารกับผู้ใช้ (GUI) ของ Website Dashboard นั้น ได้กำหนดให้มีหน้าการใช้งานหลักทั้งหมด 11 หน้า ได้แก่

1. Get started page: หน้าการเข้าสู่ระบบ หรือลงทะเบียนเข้าใช้งาน
2. Login page: หน้าการเข้าสู่ระบบ
3. Register page: หน้าการลงทะเบียนเข้าใช้งาน
4. Home page: หน้าแสดงเมนูของฟังก์ชันต่าง ๆ
5. Store setting page: หน้าการตั้งค่าข้อมูลร้านค้า
6. Stock tracker page: หน้าการติดตามคลังสินค้า
7. Stock manager page: หน้าการจัดการคลังสินค้า
8. Stock manager page > Out of stock page: หน้าการจัดการสินค้าที่ไม่เหลือในคลัง
9. Selling tracker page: หน้าการติดตามยอดขายตามลำดับคำสั่งซื้อ
10. Selling report page: หน้าการแสดงผลข้อมูลยอดขายตามรายวัน รายเดือน และรายปี โดยสามารถคุ้มครองข้อมูลของสินค้า หรือแยกตามสินค้าหนึ่ง ๆ ได้
11. About us page: หน้าแสดงรายละเอียดของผลิตภัณฑ์ของเรา

บทที่ 5

บทสรุปและข้อเสนอแนะ

- 5.1 สรุปผล**
- 5.2 ปัญหาที่พบและแนวทางการแก้ไข**
- 5.3 ข้อเสนอแนะและแนวทางการพัฒนาต่อ**

บรรณานุกรม

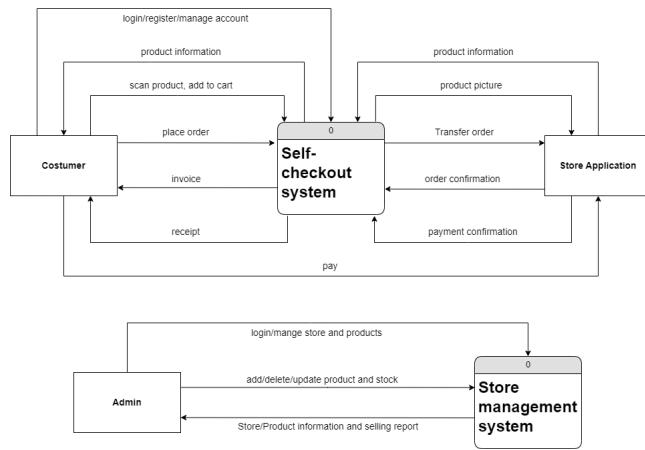
- [1] Artificial neural network. 2013.
- [2] Activation function. 2022.
- [3] Cross-validation. 2023.
- [4] Feedforward neural network. 2023.
- [5] Multilayer perceptron. 2023.
- [6] HD COE. Convolutional neural network คืออะไร. 2020.
- [7] jittagornp. Websocket คืออะไร ทำงานยังไง. 2021.
- [8] Dr. James McCaffrey. The neural network input-process-output mechanism. 2013.
- [9] Kathiravan Srinivasan, Lalit Garg, Debajit Datta, and Abdulellah Abdullah Alaboudi. Performance comparison of deep cnn models for detecting driver's distraction. 2021.
- [10] Wenjin Tao, Md. Al-Amin, Haodong Chen, and Ming C. Leu. Real-time assembly operation recognition with fog computing and transfer learning for human-centered intelligent manufacturing. 2020.

ภาคผนวก

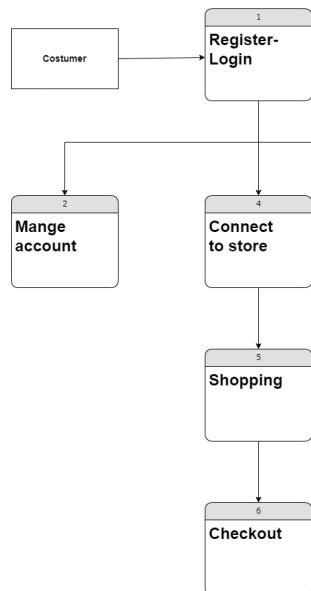
ການພັນວັນ ກ

Data Flow

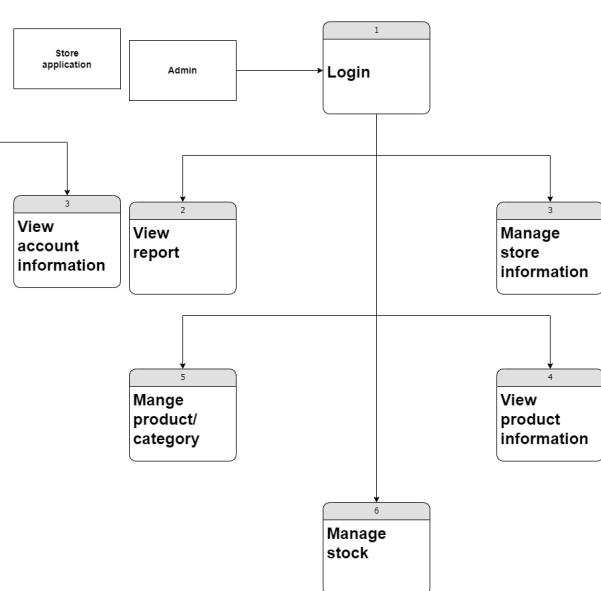
DFD level 0: Self-checkout system & Store management system



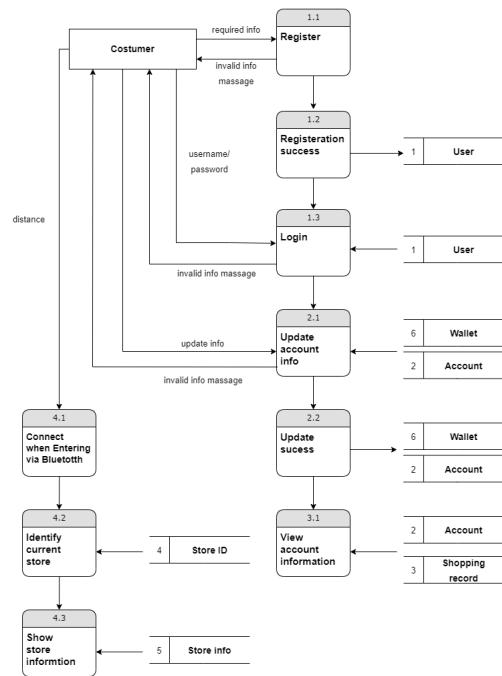
DFD level 1: Self-checkout system



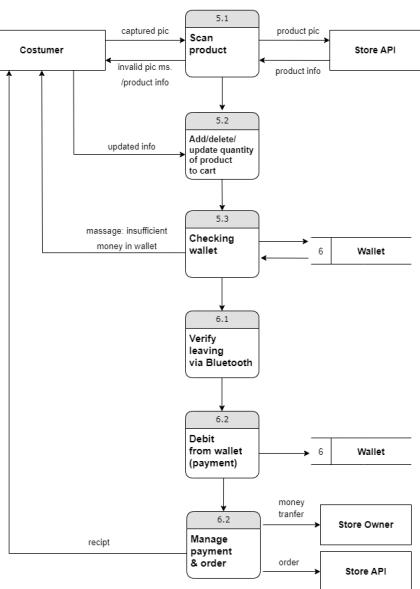
DFD level 1: Store management system



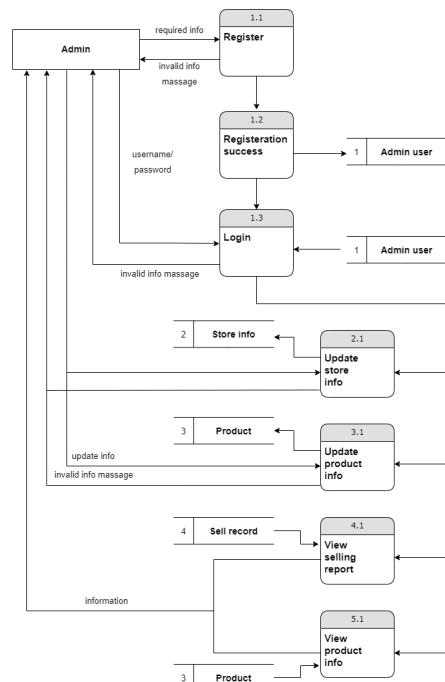
DFD level 2: Self-checkout system (Process 1-4)



DFD level 2: Self-checkout system (Process 5-6)



DFD level 2: Store management system



week	work
4 Dec - 10 Dec 2022	Init Github repositories for Back-end , Website and Application
11 Dec - 17 Mar 2022	Collect products data from 422 shop
18 Dec - 24 Dec 2022	Implementing Google colab for pre processing product data
25 Dec - 21 Dec 2022	Coding mechanism for Training model
1 Jan - 7 Jan 2023	Training model and model validation
8 Jan - 14 Jan 2023	Designing Mobile Application's UI/UX
15 Jan - 21 Jan 2023	Designing Website dashboard's UI/UX
22 Jan - 28 Jan 2023	Prepare for midterm presentation.
29 Jan - 4 Feb 2023	Designing architecture neural network model for classification
5 Feb - 11 Feb 2023	Feasibility study Website dashboard's architecture
12 Feb - 18 Feb 2023	Feasibility study Application's architecture
19 Feb - 25 Feb 2023	
26 Feb - 4 Mar 2023	Implementing supabase database for store data and products data
5 Mar - 11 Mar 2023	Refactor most of traing code
12 Mar - 17 Mar 2023	Write and fix the report

ຮູບທີ່ ໜ.1: Work table