

รายงานฉบับสมบูรณ์

ระบบคัดแยกขยะโดยใช้เทคนิคการประมวลผลด้วยภาพและซีอิจายแลกเปลี่ยนขยะ เพื่อแก้ไขปัญหาสิ่ง

แวดล้อมที่เกิดจากมลพิษขยะ

(Waste Separation Using Image Processing and Trading System to Solve Environmental
Problems Occurring from Waste Pollution)

เสนอต่อ

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

กระทรวงวิทยาศาสตร์และเทคโนโลยี

ได้รับทุนอุดหนุนโครงการนวัตกรรม

โครงการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทยครั้ง ๒๒

ประจำปีงบประมาณ ๒๕๖๓

โดย

นาย ธนาธิร์ เอี่ยมยิ่งสกุล

นาย นราภิญ ทับทิมโต

นาย วชิรินทร์ ศิริเนาวกุล

ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

กิตติกรรมประกาศ

ขอขอบคุณ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์ และเทคโนโลยีแห่งชาติ ที่ให้ทุนสนับสนุนในการพัฒนาโครงการ ระบบคัดแยกขยะโดยใช้เทคนิคการประมวลผลด้วยภาพและซื้อขายแลกเปลี่ยนขยะ เพื่อแก้ไขปัญหาสิ่งแวดล้อมที่เกิดจากมลพิษขยะ ภายใต้การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทยครั้งที่ 22

ขอบคุณ ดร. อัญชาติสา แต้ตระกูล อาจารย์ที่ปรึกษาโครงการ ผู้ที่สละเวลาในการให้ความรู้และคำแนะนำตลอดการพัฒนาโครงการ

ขอบคุณ CAST LAB, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ที่ให้การส่งเสริมและสนับสนุนผู้พัฒนา ในการเข้าร่วมการแข่งขันอันเป็นประโยชน์นี้ ตลอดจนเอื้อเพื่อสถานที่สำหรับการพัฒนาโครงการ

นอกจากนี้ ขอขอบคุณ ผู้ปกครองของผู้พัฒนา เพื่อน และอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ที่ให้การสนับสนุนทางด้านข้อมูลเกี่ยวกับขยะต่างๆ และคอยเป็นกำลังใจเสมอมา

ผู้พัฒนาโครงการ

CopyCatch

รายงานผลการตรวจสอบเอกสาร

(กรุณาแนบไฟล์รายงานผลฉบับนี้ในหน้าที่ 2 ของข้อเสนอโครงการ)

ชื่อเอกสาร : ระบบคัดแยกขยะโดยใช้เทคนิคการประมวลผลด้วยภาพและซื้อขายแลกเปลี่ยนขยะเพื่อแก้ไขปัญหาลักลอบที่เกิดจากมลพิษขยะ (22p14c0511)

ชื่อ-นามสกุล : ธนาสนธิ เอี่ยมยิ่งสกุล

เปอร์เซ็นต์ความคล้ายทั้งหมด : 1.76 % (ตรวจ ณ วันที่ 31 มกราคม 2563)

เปอร์เซ็นต์ความคล้ายทั้งหมด คือ เปอร์เซ็นต์ความคล้ายทั้งหมดที่เอกสารของเราเหมือนกับแหล่งอื่น

เปอร์เซ็นต์ความคล้ายตามแหล่งที่มา คือ เอกสารของเรามีความคล้ายเป็นกี่เปอร์เซ็นของแหล่งแหล่ง

* หมายเหตุ หากเปอร์เซ็นความคล้ายทั้งหมดเกิน 60% หรือมีรายการแหล่งที่มาใดที่มีค่าความคล้ายมากกว่า 20% ควรมีการอ้างอิงแหล่งที่มาในส่วนที่มีความคล้าย

รายการแหล่งที่มาที่ควรอ้างอิง

1	20p21c0103_fullreport	4.47%	
2	13P21E015	2.26%	
3	14p22i007: Invertebrate land แคนเนอร์ศัตรู	2.11%	
4	18p14s0010: โปรแกรมจำลองการผันน้ำลงทะเลด้วยเครื่องผลักดันน้ำ (มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต)	2.11%	
5	14p21i002: ผู้พิทักษ์ความสะอาด	2.09%	
6	14p41i015: โครงการ บทเรียนออนไลน์ เรื่องหลักปรัชญาเศรษฐกิจพอเพียง (โรงเรียนโකสพ วิทยาสรรพ)	1.99%	
7	18p34s0071: ระบบคาดการณ์ฝนตกและการประมวลผลภาพเพื่อวัดระดับน้ำท่วม (มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต)	1.76%	
8	20p21e0090_fullreport	1.72%	
9	20p15e0058_fullreport	1.62%	
10	20p23w0005_fullreport	1.48%	

บทคัดย่อ

ปัจจุบันปัญหาที่เกิดขึ้นจากการลักพาทายาดมีมากมายไม่ว่าจะเป็นปัญหามลพิษทางอากาศที่เกิดจากการเผาไหม้ในชุมชนและภาคหรือปัญหามลพิษทางน้ำจากการทิ้งขยะลงแม่น้ำและทะเล ซึ่งส่งผลเสียต่อทั้งสุขภาพร่างกายของผู้คนและสภาพแวดล้อมโดยรอบ ทั้งนี้ประเทศไทยเป็นประเทศที่มีจำนวนขยะจำนวนมหาศาล แต่ในขณะเดียวกัน ความสามารถในการจัดการขยะให้ถูกต้องในประเทศไทยเป็นในพิศทางตรงข้ามกัน^[1] หนึ่งในวิธีการลดปัญหาที่ ก่อมาได้ก็คือ การจัดการขยะที่มีประสิทธิภาพ เช่น การแยกขยะ การรีไซเคิลขยะ และ การอำนวยความสะดวก หลากหลายในการจัดการขยะ เป็นต้น โครงการนี้มีจุดประสงค์เพื่อที่จะสร้างระบบและแอปพลิเคชันที่ทำให้เกิดการ จัดการขยะที่มีประสิทธิภาพโดยมีความสามารถได้แก่ ให้ความรู้เกี่ยวกับขยะและการจัดการด้วยเทคนิคการ ประมวลผลภาพ เพิ่มแรงจูงใจในการจัดการขยะและขายขยะโดยการแสดงราคาขยะที่สามารถขายได้ เพิ่ม ความสะดวกสบายแก่การซื้อ-ขายขยะให้ทั้งผู้ที่ต้องการขายและซื้อขายด้วยระบบการติดต่อซื้อขายบนแอป พลิเคชัน นอกจากนั้นยังมีการทำระบบการนำเสนอข้อมูลเป็นรูปแบบของภาพและกราฟเพื่อให้ผู้ที่เกี่ยวข้องกับ การจัดการขยะสามารถนำข้อมูลเหล่านี้ไปเคราะห์ต่อยอดได้

คำสำคัญ Cross Platform; Mobile Application; Deep Learning; Image Processing; Object Detection; Waste Management; Marketplace; Computer Vision; Data Visualization

Currently, there are many problems with waste pollution, for example, air pollution problems caused by landfill or water pollution problems from dumping waste into rivers and seas. These problems affect both the physical health of people and the surrounding environment. Thailand is also one country that has been dealing with a growing amount of waste. However, the ability to properly manage waste in that country is in the opposite direction. One of the ways to reduce the mentioned problems is to have efficient waste management, such as waste separation, waste recycling, and convenience management, etc. This project aims to create a system that enables efficient waste management. The application provides knowledge about waste using image processing techniques. The prices of waste that can be sold will be shown to increase incentives for waste management. Moreover, users can buy or sell junk easily using the trading system in the application for their convenience. Also, we provide some inside-information through data visualization so that those involved in waste management can take these data for further analysis.

บทนำ

คงจะปฏิเสธไม่ได้ว่าปัจจุบันมีการผลิตสินค้าอุปโภคบริโภคออกสู่ตลาดมากมายจากการเติบโตอย่างต่อเนื่องของเศรษฐกิจส่งผลให้เกิดขยะเพิ่มขึ้นอย่างมหาศาล ถ้านับแค่ในปีพ.ศ. 2561 ประเทศไทยมีขยะมูลฝอยเกิดขึ้นถึง 27.8 ล้านตัน แต่ความสามารถในการจัดเก็บขยะนั้นกลับมีไม่ถึง 70% ของขยะทั้งหมด โดยมีเพียงแค่ 34% ที่ถูกคัดแยกและนำไปทำใหม่ ถูกต้องและยังมีของเสียที่เป็นอันตรายต่อชุมชนที่ถูกกำจัดอย่างถูกต้องเพียง 13% เท่านั้น^[1,2] จึงก่อให้เกิดขยะมูลฝอยตกค้างและปัญหาสิ่งแวดล้อมมากมายตามมา อาทิ ปัญหามลพิษทางอากาศจากการเผาขยะมูลฝอยกลางแจ้งที่ทำให้เกิดควันและมลพิษ ปัญหากลิ่นเหม็นที่ส่งผลต่อชุมชนโดยรอบที่เกิดจากการเก็บขยะมูลฝอยไม่หมดหรือเกิดจากสถานที่พักเก็บขยะในบริเวณนั้น ปัญหาแหล่งพากห่านนำโรค เช่น หนูและแมลง รวมถึงปัญหามลพิษทางน้ำที่เกิดจากขยะมูลฝอย ซึ่งประเทศไทยเป็นประเทศที่ขยะทึ่งลงสู่ทะเลมากที่สุดเป็นอันดับที่ 6 ของโลก^[3] โดยปัญหาทั้งหมดนั้นส่งผลเสียต่อสุขภาพของคนและสัตว์ทั้งทางตรงและทางอ้อม เช่น โรคปอดจากการสูดมลภาวะที่เป็นพิษ การได้รับสาร_PROTO_มากเกินไปจากการทานอาหารจำพวกปลาที่มีสารprotothekotoksing พลาสติกที่พับในกระเพาะของสัตว์และยังส่งผลเสียอีกด้วย

วิธีการที่จะช่วยลดปริมาณขยะที่ตกค้างสะสมในบ่อขยะฝั่งกลบจนก่อมลพิษทางสิ่งแวดล้อมก็คือการรีไซเคิลขยะผ่านการขายขยะรีไซเคิลให้กับผู้รับซื้อขยะ แล้วผู้รับซื้อขยะจะนำขยะที่ได้ไปเข้าสู่ระบบรีไซเคิลเพื่อดึงทรัพยากรกลับเข้าสู่ระบบ แต่อย่างไรก็ตามการแยกขยะก่อนทิ้งหรือการส่งขยะรีไซเคิลกลับเข้าสู่ระบบผ่านการขายให้ผู้รับซื้อขยะนั้นยังไม่นิยมและยังขาดความคล่องตัวอยู่ อีกทั้งยังจำเป็นที่จะต้องอาศัยความรู้ความเข้าใจและแรงจูงใจของผู้ทึ่งขยะด้วย โดยที่ข้อจำกัดในเรื่องของความคล่องตัวนั้นรวมถึงความลำบากในการติดต่อเรียกผู้รับซื้อขยะมารับซื้อขยะ ที่ผู้ขายจำเป็นต้องมีข้อมูลของผู้ซื้อขยะเพื่อใช้ในการติดต่อ และยังต้องบอกที่อยู่ทุกครั้งเมื่อต้องการขายขยะ อีกทั้งการที่จะรับทราบราคารับซื้อขยะจำเป็นต้องทำผ่านการติดต่อหาผู้ซื้อเป็นรายคนเนื่องมาจากผู้รับซื้อแต่ละรายจะหาราคารับซื้อที่ไม่เหมือนกัน ทำให้เป็นเรื่องยากในการตัดสินใจในการเลือกผู้รับซื้อขยะของผู้ขายหากมีความประสงค์ที่จะเบริรยบเทียบราคานอกจากนี้ยังมีความไม่คล่องตัวในผู้รับซื้อขยะ ที่การไปรับขยะอาจจะได้ปริมาณขยะที่ไม่มากพอตามเป้า เนื่องจากผู้ขายขยะไม่ได้คัดแยกขยะอย่างถูกต้อง ทำให้ขยะที่สามารถรับซื้อได้มีจำนวนลดน้อยลง หรืออาจจะต้องผ่านกระบวนการทำความสะอาดอีกครั้งก่อนเข้าสู่กระบวนการถัดไป ทำให้เพิ่มต้นทุนของผู้รับซื้ออีกทั้งรายได้จากการขายขยะของผู้ขายขยะนั้นลดลงจากการตีราคาของผู้ซื้อ และการติดต่อกับผู้ขายขยะเป็นจำนวนมาก ก็นำไปสู่การวางแผนการเดินทางที่ยากมากขึ้นด้วย

ดังที่กล่าวไปข้างต้น ปัญหาสำคัญที่ยังคงอาศัยอยู่ในตัวผู้ทึ่งขยะหลายคนก็คือ ความไม่เข้าใจในวิธีการจัดการขยะที่ถูกต้อง จากการค้นคว้างานวิจัยที่เกี่ยวข้อง ได้ค้นพบว่า ความไม่รู้และความไม่เข้าใจนี้ ส่งผลต่อการลงทะเบียนในการทึ่งขยะ^[4] ซึ่งจะส่งผลเสียร้ายแรงต่อกระบวนการกำจัดขยะที่ต้นทุนที่ใช้ในกระบวนการจะเพิ่มขึ้น และขยายเวลาในการกำจัดไม่ทัน ก็จะกลายเป็นขยะเน่าเสียสะสมในระบบต่อไป แต่ว่าในอีกมุมมองหนึ่ง การที่ผู้คนจะไม่ลงทะเบียนการคัดแยกขยะและให้ความสำคัญกับการจัดการอย่างถูกต้องมากขึ้นก็ต่อเมื่อพวก

เขามีความสอดคล้องสบายนในการกระทำ^[5-8] เนื่องจากขยะบางชนิด เช่นขวดน้ำพลาสติก กระบวนการคัดแยกต้องทำถึงขั้นที่แกะฉลาก แกะฝาพลาสติกออก เพราะเป็นพลาสติกคนละประเภท จึงปฏิเสธไม่ได้ว่ากระบวนการแยกขยะนั้นไม่ได้สอดคล้องสบายนี้และค่อนข้างใช้เวลาพอสมควร แต่อย่างไรก็ตาม นอกเหนือจากความสอดคล้องสบายนี้แล้ว ยังมีเรื่องของผลประโยชน์ ที่จะมาเป็นอีกส่วนสำคัญ ที่จะช่วยส่งเสริมให้ผู้คนมีความอยากรีดการขยายอย่างถูกต้องมากขึ้น เช่น การบีบขาดและแกะฉลากพลาสติกก่อนขาย ช่วยลดปริมาณถุงที่ผู้ทิ้งขยะจะต้องใช้และเพิ่มราคารับซื้อในบางพื้นที่ เพราะต้นทุนของผู้รับซื้อลดลงจากการที่วิธีนี้ช่วยลดงานตัดฉลากของคนงานในโรงงานเบเกอรี่ อีกทั้งขยายส่วนนี้ก็จะผ่านไปยังกระบวนการต่อไปได้ทันทีทำให้ในหนึ่งวันผู้รับซื้อขยะสามารถรับซื้อขยะได้มากขึ้น เพราะเวลาที่ใช้น้อยลงจะนั่นแล้วการแยกขยะอย่างมีประสิทธิภาพจะเกิดขึ้นได้จากการให้ความรู้เกี่ยวกับการแยกขยะและสร้างแรงจูงใจให้กับคนทั่วไปควบคู่ไปกับการเสริมสร้างความสอดคล้องสบายนี้ของทั้งผู้คนทั่วไปที่ทิ้งขยะเป็นประจำ และผู้ดูแลผู้รับซื้อขยะ

ด้วยเหตุนี้เราจึงตั้งเป้าหมายที่พัฒนาระบบและแอปพลิเคชันบนสมาร์ทโฟน เพื่อเพิ่มความสอดคล้องสบายนี้ให้ผู้ใช้งานจัดการกับขยะได้อย่างถูกต้องโดยไม่ต้องอาศัยความรู้ความเข้าใจ เราจะใช้การประมวลผลภาพที่จะแสดงข้อมูลของขยะประเภทต่างๆ ทั้งขยะที่ขายได้และไม่สามารถขายได้ อีกทั้งยังแสดงให้เห็นถึงมูลค่าของขยะแบบละเอียดเพื่อสร้างแรงจูงใจในการจัดการขยะของผู้ใช้งาน โดยฟังก์ชันนี้จะมาพร้อมกับระบบ Marketplace ที่จะเข้ามาช่วยเพิ่มความสอดคล้องสบายนี้ในขั้นตอนของการติดต่อซื้อขายระหว่างคนทิ้งขยะกับผู้รับซื้อขยะ ลดกระบวนการตรวจสอบเวลานัดรับขยะ และช่วยนำเสนอเส้นทางที่เหมาะสมในการเดินทาง สำหรับผู้รับซื้อขยะ ทำให้การวางแผนสามารถทำได้สะดวก และยืดหยุ่นมากขึ้น การเปิดโอกาสให้ผู้ที่มีความสนใจในข้อมูลการจัดการขยะภายในแอปพลิเคชันก็สามารถเข้าถึงได้ ผ่านฟังก์ชัน Data Visualization เพื่อนำข้อมูลที่ได้ไปใช้ประโยชน์ต่อไปในอนาคต

สารบัญ

	หน้า
กิตติกรรมประกาศ	
1. ผลการตรวจสอบการคัดลอกเอกสาร (CopyCatch)	iii
2. บทคัดย่อ	iv
3. บทนำ	v
4. สารบัญ	vii
5. วัตถุประสงค์และเป้าหมาย	1
6. รายละเอียดของการพัฒนา	2
6.1 เนื้อเรื่องย่อ	2
6.2 ทฤษฎีหลักการ	5
6.2.1 Convolution Neural Network (CNN)	5
6.2.2 Region-based Convolutional Neural Network	7
6.2.3 Selective Search	7
6.2.4 VGG Neural Network	7
6.2.5 Residual Neural Network (ResNet)	8
6.2.6 Fast Region Convolutional Neural Network (Fast RCNN)	8
6.2.7 Region Proposal Network (RPN)	9
6.2.8 Faster Region Convolutional Neural Network (Faster RCNN)	12
6.2.9 You Only Look Once version 3 (YOLOv3)	13
6.2.10 Cross Platform	15
6.2.11 Dijkstra's Algorithm	16
6.3 เครื่องมือที่ใช้	17
6.4 รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค	18
6.4.1 Input/Output Specification	18
6.4.2 Functional Specification	18
6.4.3 โครงสร้างซอฟต์แวร์	19
6.4.4 การพัฒนา Waste Detection	20
- ข้อมูลและการเตรียมข้อมูล	20
- การเลือกโมเดล	22
- วิธีการดำเนินการ	22

-	<i>Faster-RCNN</i>	22
-	<i>YOLOv3</i>	24
-	ทดสอบและประเมินผล	24
6.5	ขอบเขตและข้อจำกัด	26
7.	กลุ่มผู้ใช้โปรแกรม	28
8.	ผลของการทดสอบโปรแกรม	28
8.1.	การทดสอบโมเดล	28
8.2.	การทดสอบแอปพลิเคชัน	31
8.3.	การทดสอบเว็บไซต์	36
9.	ปัญหาและอุปสรรค	38
9.1.	โมเดล	38
9.2.	แอปพลิเคชัน	38
9.3.	เว็บไซต์	39
10.	แนวทางในการพัฒนาและประยุกต์ใช้ร่วมกับงานอื่นๆ ในขั้นตอนไป	40
11.	ข้อสรุปและข้อเสนอแนะ	40
12.	เอกสารอ้างอิง	42
13.	สถานที่ติดต่อของผู้พัฒนาและอาจารย์ที่ปรึกษา โทรคัพพ์ มือถือ โทรศัพท์ E-mail	45
14.	ภาคผนวก	a
-	คู่มือการติดตั้ง	a
-	คู่มือการใช้งาน	a
-	แอปพลิเคชัน	a
-	การสมัครสมาชิก	a
-	การลงชื่อเข้าใช้	b
-	การแก้ไขข้อมูล หรือ ลงทะเบียนออกจากระบบ	c
-	การปรับจำนวนขายที่ไม่อยู่ในคลังของผู้ขาย	c
-	การตรวจสอบรายละเอียดวิธีการจัดส่งและจัดเก็บเข้าคลังของผู้ขาย	d
-	การกดคำสั่งขายของผู้ขาย	e
-	การตรวจสอบคำร้องขอขายของผู้ขาย	f
-	การปรับเปลี่ยนราคารับซื้อของผู้รับซื้อ	f
-	การกดรับคำร้องขายของผู้ขายแบบด่วน ในระยะที่ล่นใจ	g
-	การกดรับคำร้องขอขายของผู้ขายเลือกขายให้โดยตรง	h

- การค้นหาเล่นทางที่เหมาะสมที่สุดในการเดินทางไปรับขยะตามสถานที่ต่าง *h*
- เว็บไซต์ *i*
 - การเข้าใช้เว็บไซต์ *i*
 - การดูข้อมูลจำนวนขยะทั้งหมดภายในแอปพลิเคชัน *i*
 - การดูข้อมูลจำนวนขยะตามรหัสไปรษณีย์ภายในเว็บไซต์ *j*
 - การเรียกขอดูข้อมูลที่อยู่ผู้รับซื้อขยะภายในแอปพลิเคชัน *k*

5. วัตถุประสงค์และเป้าหมาย

เป้าหมายของโครงการคือ การประยุกต์และนำใช้เทคโนโลยีต่างๆ มาพัฒนาแอปพลิเคชันเพื่อบรรเทาสาเหตุของปัญหาการแยกขยะ จากการสร้างความสะท้วงสบายน ให้ความรู้ และสร้างแรงจูงใจให้กับผู้เชิงงาน โดยการสร้างระบบ Marketplace เพื่อสร้างความสะท้วงสบายนในการจัดการขยะ และการประเมินภาพที่จะเข้ามาช่วยในการแสดงให้เห็นมูลค่าของขยะด้วยการแสดงราคาสำหรับขยะที่สามารถขายได้ และให้ความรู้ในการจัดการขยะสำหรับขยะที่ไม่สามารถขายได้ นอกจากนี้ยังสามารถเข้าถึงข้อมูลของแอปพลิเคชันผ่านตัวของ Data Visualization เพื่อให้ผู้ที่สนใจนำข้อมูลไปใช้ประโยชน์ต่อ

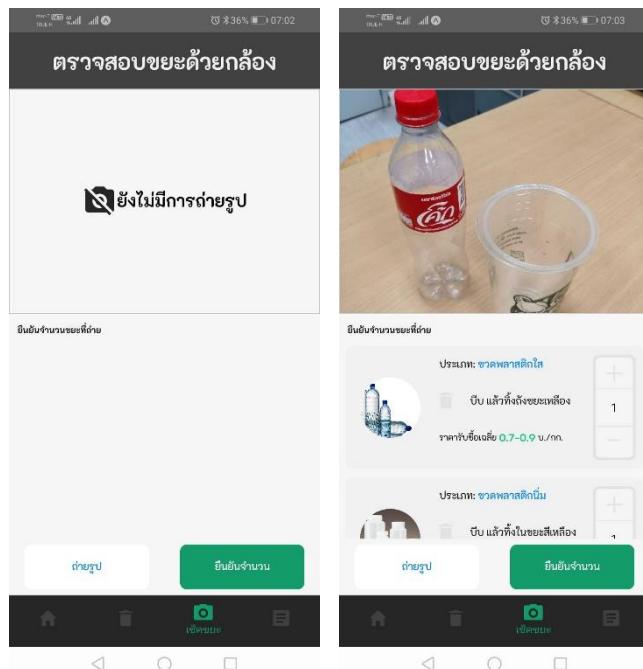
วัตถุประสงค์ของการจัดทำโครงการนี้

1. เพื่อให้ความรู้เกี่ยวกับการแยกขยะและสร้างแรงจูงใจในการแยกขยะให้กับผู้คนทั่วไป
2. เพื่อให้ผู้ขายขยะสามารถจัดการขยะได้อย่างมีประสิทธิภาพ
3. เพื่อสร้างความสะท้วงสบายน ให้กับทั้งผู้ขายและผู้รับซื้อขยะ
4. เพื่อสร้างช่องทางในการเก็บรวบรวมข้อมูลเกี่ยวกับขยะ ที่สามารถนำไปใช้ประโยชน์ต่อได้

6. รายละเอียดของการพัฒนา

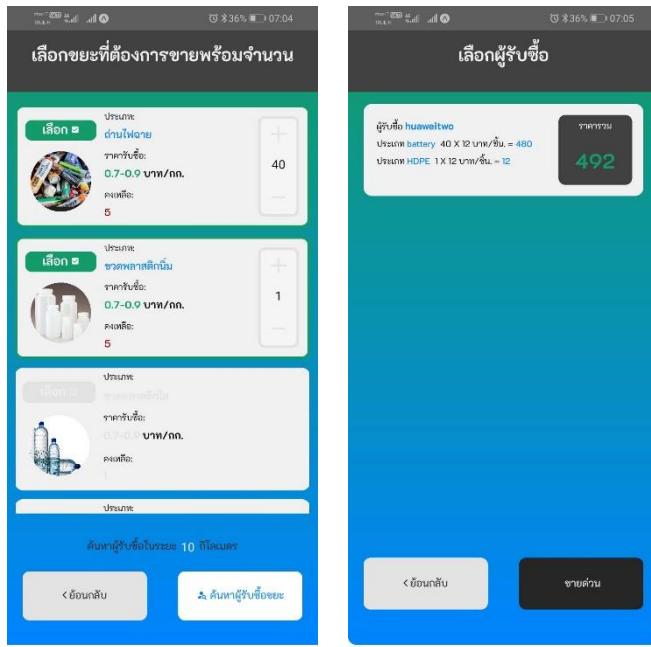
6.1. เนื้อเรื่องย่อ

เนื่องจากความสูงของสินค้าที่ต้องจัดการห้ามนำขยะเข้าสู่ระบบมากขึ้น โดยแอปพลิเคชันนี้จะทำให้การดำเนินการต่างๆ ตั้งแต่การขายจนถึงการซื้อขายนั้นมีความสะอาดมากยิ่งขึ้น ต้องห้ามนำขยะมาขายและผู้ซื้อขายจะต้องจัดการกับมันอย่างไร ปัญหานี้จะหมดไปเนื่องจาก Image Processing Function ในแอปพลิเคชันจะตรวจสอบประเภทของขยะและระบุราคากลางที่คนขายรับซื้อหรือวิธีการจัดการอย่างถูกต้องเมื่อขยะนั้นเป็นขยะที่ริใช้แล้วไม่ได้ให้โดยอัตโนมัติ อีกทั้งยังสามารถเช็คราคาขยะได้ดังรูปที่ 6.2 ทำให้ไม่จำเป็นต้องเข้า Website หลายขั้นตอนเพื่อค้นหาข้อมูลเหล่านี้อีก



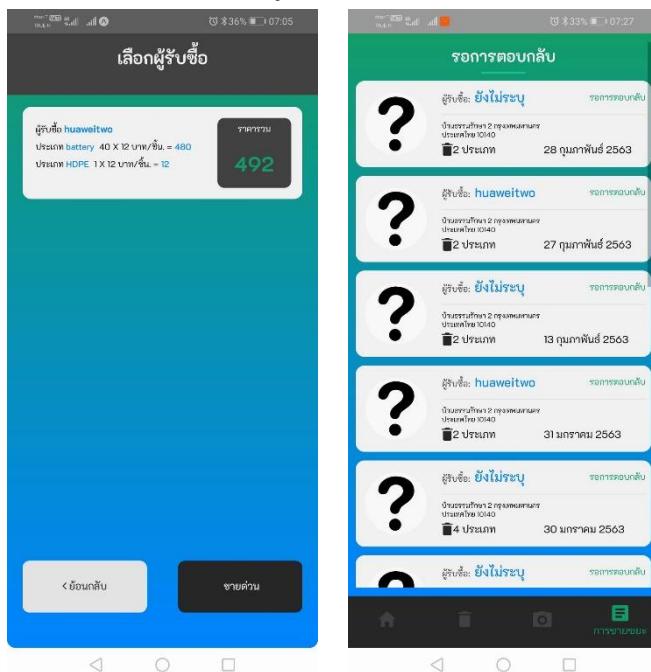
รูปที่ 6.1 การบอกระบบทะเบียนโดยใช้ Image Processing

เมื่อผู้ขายขยะได้รับข้อมูลทางเลือกต่างๆ ในการจัดการขยะแล้ว หากเขามีความประสงค์ที่จะขายขยะ ก็ไม่จำเป็นต้องกดโทรศัพท์เพื่อเรียกคนรับซื้อขยะ เพียงแค่เลือกผู้รับซื้อในระบบที่มีทั้งผู้รับซื้อรายใหญ่กับรายย่อยที่ผู้ขายเคยติดต่อไว้ หากผู้ขายต้องการความเร็วในการขายด้วยเพื่อรอให้ผู้รับซื้อคนใดก็ได้มาด้วยมารับคำร้องขอขายขยะ



รูปที่ 6.2 การเลือกตรวจสอบราคายาของขยะแต่ละประเภท

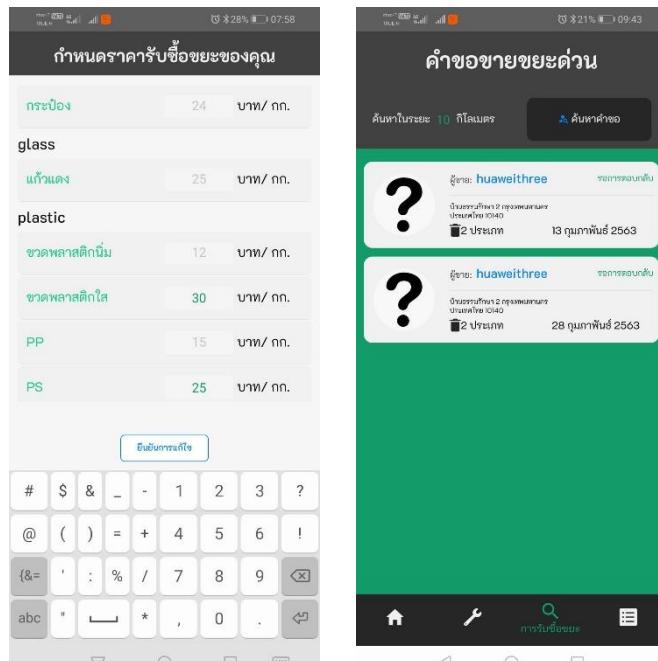
เมื่อเสร็จสิ้นการส่งคำร้องขอขายขยะแล้ว ผู้ขายจะสามารถติดตามสถานการณ์การตอบกลับของผู้รับซื้อขยะได้ในหน้า Transaction – การรับซื้อขยะ ดังรูปที่ 6.3 (ด้านขวา)



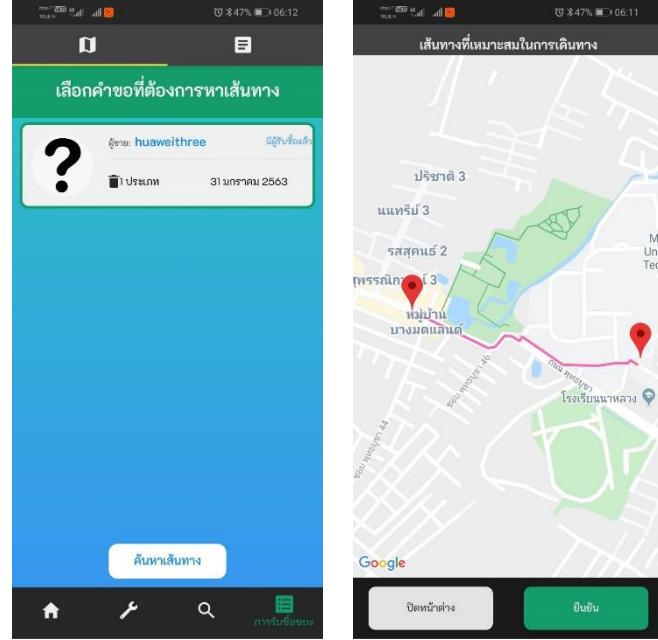
รูปที่ 6.3 การส่งคำร้องขอขายขยะ และการติดตามสถานะคำร้อง

ในมุมของผู้รับซื้อขยะ สามารถกำหนดประเภทขยะที่ตนต้องการรับซื้อได้ เมื่อมีผู้ขายขยะกดขายขยะด่วน ที่ตรงกันกับประเภทขยะที่ผู้รับซื้อรายนี้กำหนดเอาไว้ ก็จะมีการแจ้งเตือนที่แสดงขึ้นดังภาพที่ 6.4 (ด้านขวา)

จากนั้นการกำหนดแผนต่างๆ ของผู้ชี้อ้วว่าจะรับซื้อขยะอะไรจากสถานที่ใดก็จะสามารถทำได้่ายมากรีน โดยสามารถตรวจสอบตำแหน่งที่ต้องเดินทางไปเพื่อรับซื้อขยะนั้นได้ดังแสดงในรูป



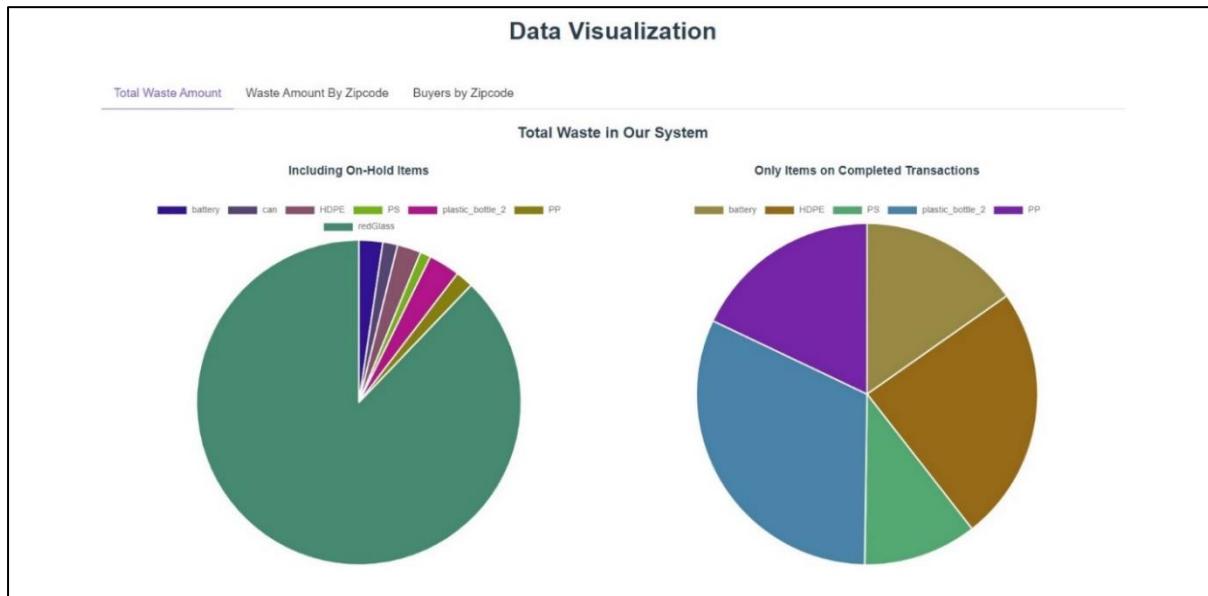
รูปที่ 6.4 การกำหนดประเภทขยะที่ต้องการรับซื้อ และการค้นหาคำขอในบริเวณใกล้เคียงที่ผู้ขายขยะกดขายด่วน



รูปที่ 6.5 การแสดงสถานที่ของผู้ขายขยะกับการค้นหาเส้นทางที่เหมาะสมที่สุด และการติดตามคำร้องขอขายขยะของผู้ขายขยะ



ท้ายที่สุด เมื่อมีข้อมูลการซื้อขายขยะแต่ละประเภท ตามท้องที่ต่างๆ บุคคลที่สนใจในการนำข้อมูลเหล่านี้ไปใช้ในการพัฒนาระบบการจัดการขยะต่อไปในอนาคตก็สามารถทำได้เช่นเดียวกัน โดยจะสามารถเข้าถึงได้ในรูปแบบเว็บไซต์ดังแสดงในรูปที่ 6.7



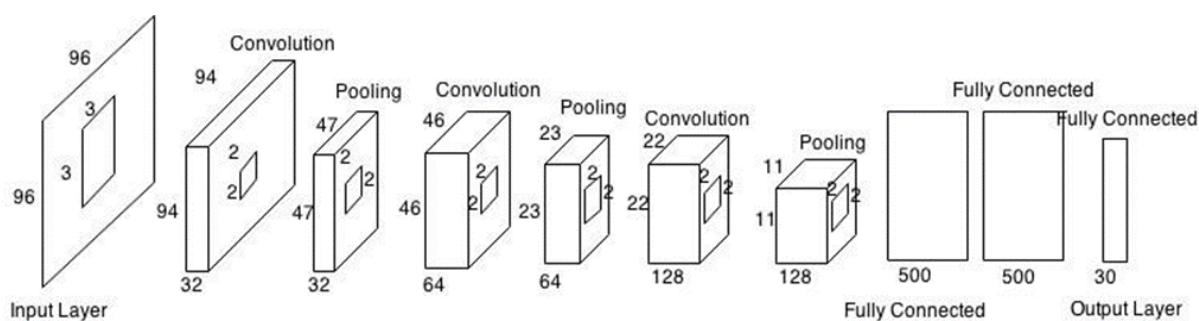
รูปที่ 6.7 สถิติการซื้อขายขยะและข้อมูลขยะในแต่ละพื้นที่
ที่มา [<https://www.columnfivemedia.com/how-to-data-visualization-report-design>]

6.2. ทฤษฎีหลักการ

6.2.1. Convolution Neural Network (CNN)

เนื่องจากแอปพลิเคชันจะมีโมดูลที่รับอินพุตเข้ามาเป็นภาพถ่ายและนำไปประมวลผลภาพเพื่อแสดงผลออกมากว่าในรูปภาพนั้นมีวัตถุหรือขยะอะไรซึ่งเป็นส่วนสำคัญในการที่จะนำไปสู่ขั้นตอนของการนำไปจัดการ

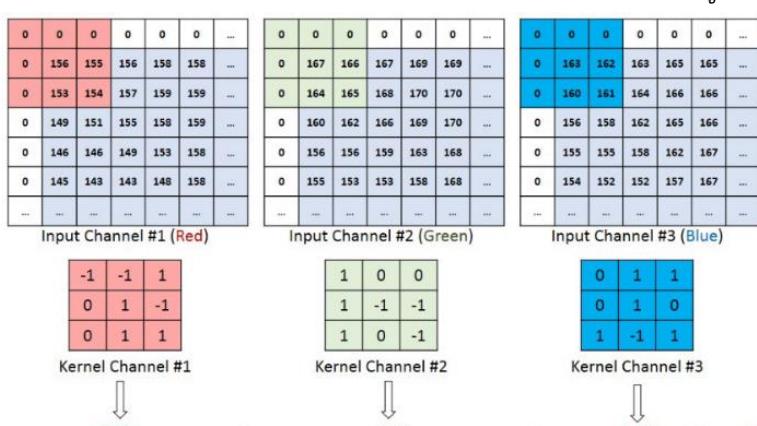
ขณะนี้ การนำไปใช้ ทำให้เราต้องศึกษาเทคนิคหรือเทคโนโลยีที่สามารถระบุวัตถุหรือขยะที่อยู่ในภาพได้อย่างแม่นยำ โดยเทคโนโลยีที่เราเลือกใช้ คือ สาขาวิชาการทางวิทยาศาสตร์ที่สามารถทำให้คอมพิวเตอร์มีความเข้าใจระดับสูงกับข้อมูลจำพวกรูปภาพและวิดีโอได้อย่าง Computer vision^[9-11] โดยเทคนิคที่เป็นที่นิยมสำหรับเทคโนโลยี Computer vision ในปัจจุบันคือ Convolutional Neural Network (CNN) ซึ่งเป็นหนึ่งในชนิดของการเรียนรู้เชิงลึกของคอมพิวเตอร์ (Deep Learning) ที่ได้รับแรงบันดาลใจจากเปลือกสมองส่วนการมองเห็นของสัตว์^[12,13] และถูกออกแบบให้เป็นระบบที่สามารถปรับเปลี่ยนเรียนรู้ลักษณะของข้อมูลที่เป็นลำดับและตำแหน่งโดยอัตโนมัติ ซึ่ง CNN เป็นโมเดลทางคณิตศาสตร์ที่โดยทั่วไปแล้วประกอบด้วย 3 ชนิดของเลเยอร์ ได้แก่ 1) Convolution Layer 2) Pooling Layer และ 3) Fully Connected Layers ดังรูปที่ 6.8



รูปที่ 6.8 – Convolution Neural Network Layer

ที่มา [<https://www.hackerearth.com/practice/notes/dependence-of-cnn-computation-time-on-the-filter-size-and-the-number-of-fully-connected-layer-units/>]

Convolution Layer เป็นส่วนที่สำคัญสำหรับ CNN ซึ่งเป็นส่วนที่ทำการสกัดคุณลักษณะ (Feature Extraction) โดยวิธี convolution ขึ้นจากภาพที่เป็นอินพุตเข้ามากับฟิลเตอร์หรือเครื่องเนลขนาดเล็กในทุกส่วนของภาพอินพุตซึ่งเป็นการทำให้ CNN มีประสิทธิภาพมากสำหรับการประมวลผลภาพ ดังรูปที่ 6.9

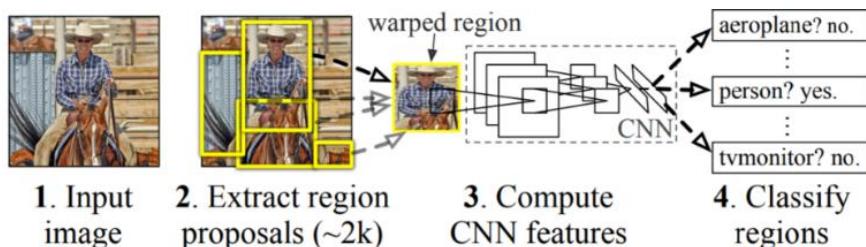


รูปที่ 6.9 – Convolution Layer

ที่มา [<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>]

6.2.2. Region-based Convolutional Neural Network

เป็น CNN ประเภทหนึ่ง ที่ถูกพัฒนาขึ้นมาเพื่อใช้สำหรับการทำ Object Detection เนื่องจากตัวของ CNN นั้นไม่สามารถทำ Object detection ได้ ทำได้เพียงแค่ Image Recognition เท่านั้น โดยวิธีในการทำ Object Detection นั้นจะทำการหา Bounding boxes หรือ Region proposals โดยใช้ Selective Search และวิจัยส่ง Bounding boxes ไปทำ Convolution ที่จะกล่องเพื่อหา Feature ในแต่ละภาพ และทำการส่งไปเพื่อคำนวณผลลัพธ์ในลำดับต่อไปดังรูปที่ 6.10



รูปที่ 6.10 – สถาปัตยกรรมของ RCNN

ที่มา

[http://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf]

6.2.3. Selective Search

เป็นหนึ่งในวิธีการหา ROI ของรูป โดยขึ้นแรกคือการสร้าง Sub-segmentation ขึ้นมาแบบสุ่มๆ แล้ววิจัยทำการรวมแต่ละ Subsegment ให้ได้ Segment ที่ใหญ่ขึ้นจนได้เป็นโครงของวัตถุที่จะเป็น ROI^[18] ดังรูปที่ 6.11



รูปที่ 6.11 – ขั้นตอนการทำ Selective Search

ที่มา [http://vision.stanford.edu/teaching/cs231b_spring1415/slides/ssearch_schuyler.pdf]

6.2.4. VGG Neural Network

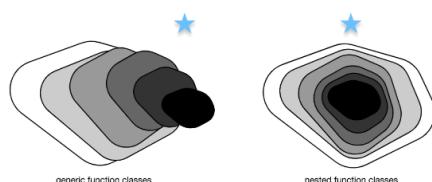
เป็นสถาปัตยกรรมประเภท CNN ที่ถูกสร้างขึ้นมาเพื่อการทำ Image Recognition โดยข้างในสถาปัตยกรรมเป็นดังรูปที่ 6.12 โดยที่มี Kernel ใน ConvNet เป็นขนาด 3x3 สำหรับการดึง Features ต่างๆ ออกมารูปภาพ และ Maxpooling เพื่อทำการ Downsampling ครั้งละครึ่งหนึ่ง^[16]



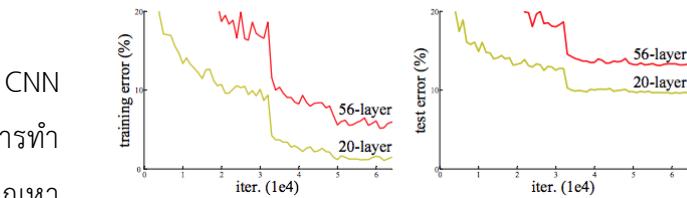
ที่มา [<https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>]

6.2.5. Residual Neural Network (ResNet)

เป็นสถาปัตยกรรมประเพณี CNN เช่นเดียวกับ VGG ถูกสร้างขึ้นมาเพื่อการทำ Image Recognition ResNet เข้ามาแก้ปัญหาในการที่ยังเพิ่มเลเยอร์เข้าไปจำนวนมาก จะยังทำให้ Error rate มากขึ้นตามไปด้วย^[17, 20] ดังรูปที่ 6.13 โดยแทนที่จะทำ ConvNet ติดกันทีละ 2-3 เลเยอร์ ResNet ไม่มี ConvNet ที่ติดกันเลย นอกจานี้ยังส่งต่อ Identity Block ซึ่งเป็นคุณลักษณะของเลเยอร์ก่อนหน้ามาคิดรวมในเลเยอร์ถัดไป ทำให้เลเยอร์ถัดๆ ไปยังคงคุณสมบัติของเลเยอร์ชั้นก่อนๆ อยู่ด้วย ส่งผลให้โอกาสที่ Function classes จะออกห่าง Ground truth น้อยลง^[2] ตามรูปที่ 6.14 โดยที่สถาปัตยกรรมของ ResNet ถูกออกแบบดังรูปที่ 6.15

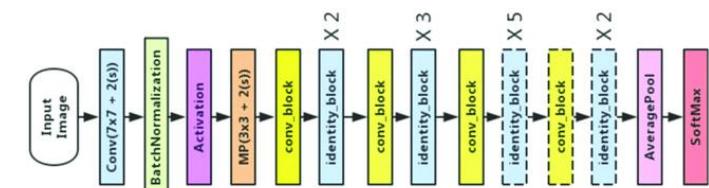


รูปที่ 6.14 – (ซ้าย) Function classes แบบปกติ (ขวา) Function classes แบบช้อน
ที่มา [<https://arxiv.org/pdf/1512.03385.pdf>]



รูปที่ 6.13 Error ที่เพิ่มจากการเพิ่มเลเยอร์ (ซ้าย)
training (ขวา) testing

ที่มา [<https://arxiv.org/pdf/1512.03385.pdf>]

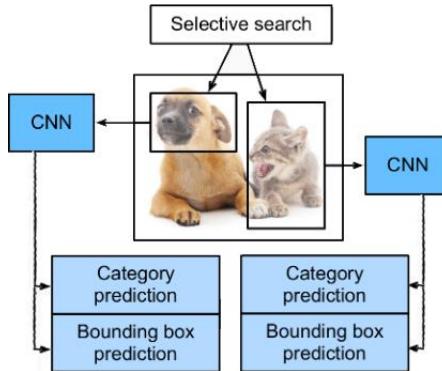


รูปที่ 6.15 – ส่วนประกอบต่างๆ ภายในสถาปัตยกรรม ResNet
ที่มา [<https://www.mdpi.com/1999-4893/12/3/51>]

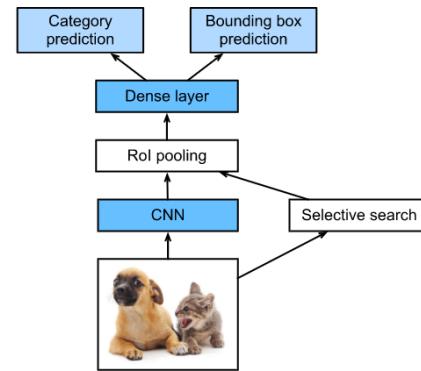
6.2.6. Fast Region Convolutional Neural Network (Fast RCNN)

เป็นสถาปัตยกรรมที่พัฒนาต่อยอดขึ้นมาจาก RCNN โดยทำการแก้ไขข้อเสียของ RCNN ในบางจุด จะเห็นได้ RCNN จะทำ Selective Search และไปหา Feature ในแต่ละ Region อีกที ดังรูปที่ 6.16 ซึ่งต่างจาก

Fast RCNN ที่จะทำการหา Feature ของทั้งภาพก่อน แล้วค่อยนำมาจับคู่กันใน RoI Pooling^[18-19, 21] ดังรูปที่ 6.17 ทำให้ไม่ต้องหา Feature ทุกๆ Region^[23]



รูปที่ 6.16 – สถาปัตยกรรมในการทำ Detection ของ RCNN
ที่มา [https://d2l.ai/chapter_computer-vision/rcnn.html]

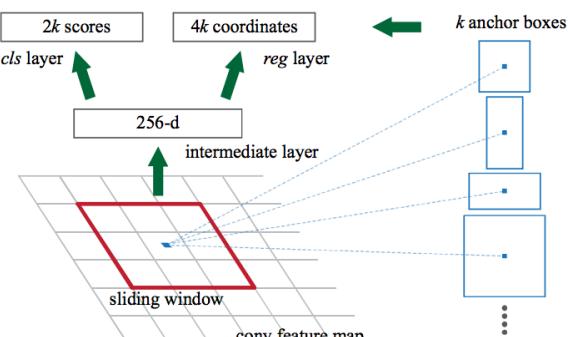


รูปที่ 6.17 – สถาปัตยกรรมในการทำ Detection ของ Fast RCNN
ที่มา [https://d2l.ai/chapter_computer-vision/rcnn.html]

6.2.7. Region Proposal Network (RPN)

เป็นอีกวิธีการหา RoI ที่มีประสิทธิภาพที่ดีกว่า Selective Search เนื่องจากว่า Selective Search เป็นอัลกอริทึม ทำให้ไม่สามารถใช้ GPU ในการประมวลผลได้ ขณะที่ RPN เป็น Network จึงสามารถใช้ GPU เพื่อเพิ่มความเร็วในการประมวลผลได้ โดยที่แต่ละขั้นตอนของ RPN^[17, 26] เป็นไปดังนี้

1. นำรูปที่มีความสูงเป็น 600 pixels เข้ามาดึง Feature ใน Pre-trained โมเดล จะได้ Feature Map ขนาด $60 \times 40 \times 512$
2. หลังจากที่คำนวนหา Feature Map แล้วก็ทำการสร้าง Anchor Boxes ขึ้นมา k จำนวน ดังรูปที่ 6.18 เพื่อนำมาใช้เป็น Ground truth ใน Loss function
 - a. จำนวนของ Anchor Boxes เป็น k ซึ่งถูกกำหนดขึ้นมาไว้ตั้งแต่ตอนแรก ซึ่งในที่นี่ $k = 9$



รูปที่ 6.18 – ขั้นตอนการสร้าง Anchor Boxes
ที่มา [<https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>]

- b. ถ้าหากค่า Intersection over Union (IoU) มากกว่า 0.7 จะให้ Anchor Boxes ที่ถูกสร้างขึ้นมาเป็น 1
- $IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$ เมื่อ A เป็น Anchor box และ B เป็น Ground truth bounding box
 - $|A \cap B|$ คือส่วนที่ทับกันระหว่าง A และ B
 - $|A \cup B|$ คือพื้นที่ทั้งหมดของ A และ B รวมกัน
3. เมื่อเสร็จจากการสร้าง Anchor Boxes แล้วจึงทำการดึง Feature อีกรอบ ด้วย Filter ขนาด $3 \times 3 \times 512$ จำนวน 512 ครั้ง แล้วเพิ่ม Padding อีก 1
- ผลลัพธ์ที่ได้ จึงเป็น Feature Map ขนาดเป็น $60 \times 40 \times 512$ เท่าเดิม ที่มีคุณลักษณะที่เจาะจงมากยิ่งขึ้น
4. หลังจากนั้นก็ทำการดึง Feature อีกครั้ง เป็นครั้งสุดท้าย ด้วย Filter ขนาด $1 \times 1 \times 512$ โดยจะแบ่งออกเป็น 2 ส่วนคือ
- Bounding boxes prediction หรือ Regression coefficients for boxes ซึ่งคือตำแหน่งของ Bounding boxes ในแกน x และ y ในระดับ pixels
 - โดยจะทำการดึง Feature ออกมาทั้ง 36 (4k) ครั้ง และได้ผลลัพธ์ออกมาเป็น Feature Map ขนาด $60 \times 40 \times 36(4k)$ เมื่อนำมาจัดเรียงใหม่ จึงได้ Feature Map ขนาด $40 \times 60 \times 9(k) \times 4$
 - ค่าใน 4 ช่องสุดท้ายคือตำแหน่งในแกน x และ y ตามด้วยความกว้างและยาวของ Bounding boxes ณ จุดเดียว
 - Binary category prediction หรือ Classification scores ซึ่งคือการทำนายว่าในตำแหน่งตามแกน x-y ใดๆ ที่สัมพันธ์กับ Feature Map ใน Bounding boxes prediction มีความน่าจะเป็นวัตถุมากน้อยแค่ไหน
 - โดยจะทำการดึง Feature ออกมาทั้งหมด 18 (2k) ครั้ง และได้ผลลัพธ์ออกมาเป็น Feature Map ขนาด $60 \times 40 \times 18(2k)$ เมื่อนำมาจัดเรียงใหม่ จึงได้ Feature Map ขนาด $40 \times 60 \times 9(k) \times 2$
 - ค่าใน 2 ช่องสุดท้ายคือ ความน่าจะเป็นที่เป็น Background และ ความน่าจะเป็นที่เป็น Object
 - หมายเหตุ ในการทดลอง โค้ดที่คณผู้จัดทำนำมาใช้ จะทำการดึง Feature Map ออกมาเป็นขนาด $60 \times 40 \times 9$ แล้วจัดเรียงใหม่เป็น $40 \times 60 \times 9 \times 1$ แทน เนื่องจากความน่าจะเป็นนี้เป็น Binary (0: Background, 1: Object)
5. ในท้ายที่สุด หลังจากที่รวม Bounding boxes กับ Classification scores เข้าด้วยกันแล้ว ก็จะทำการหา IoU ระหว่าง Bounding boxes ที่ทำนายออกมา ถ้าหากค่า IoU ระหว่าง 2 Bounding

boxes ใดๆ มีค่าเกินกว่า 0.7 จะทำการลบ Bounding boxes ที่มีค่า Classification scores น้อยกว่าออกไป เพราะถือว่าเป็น Bounding boxes เดียวกัน

Loss จากการทำเทรน์ RPN นั้นเป็น Multi-task Loss ซึ่งหมายถึงการมีค่า Loss หลายตัว สมการที่ใช้ในการลดค่า Loss จึงเป็นการรวมกันของสมการ Binary category loss กับ Bounding boxes loss เข้าด้วยกันเป็นสมการที่ (1) ตามตารางข้างล่าง โดยที่ค่า Bounding boxes loss ถูกแสดงในสมการที่ (2) และ Binary category ถูกแสดงในสมการที่ (3) และ (4)

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{classify}} \sum_i L_{classify}(p_i, p_i^*) + \lambda \frac{1}{N_{regression}} \sum_i p_i^* L_{regression}(t_i, t_i^*) \quad (1)$$

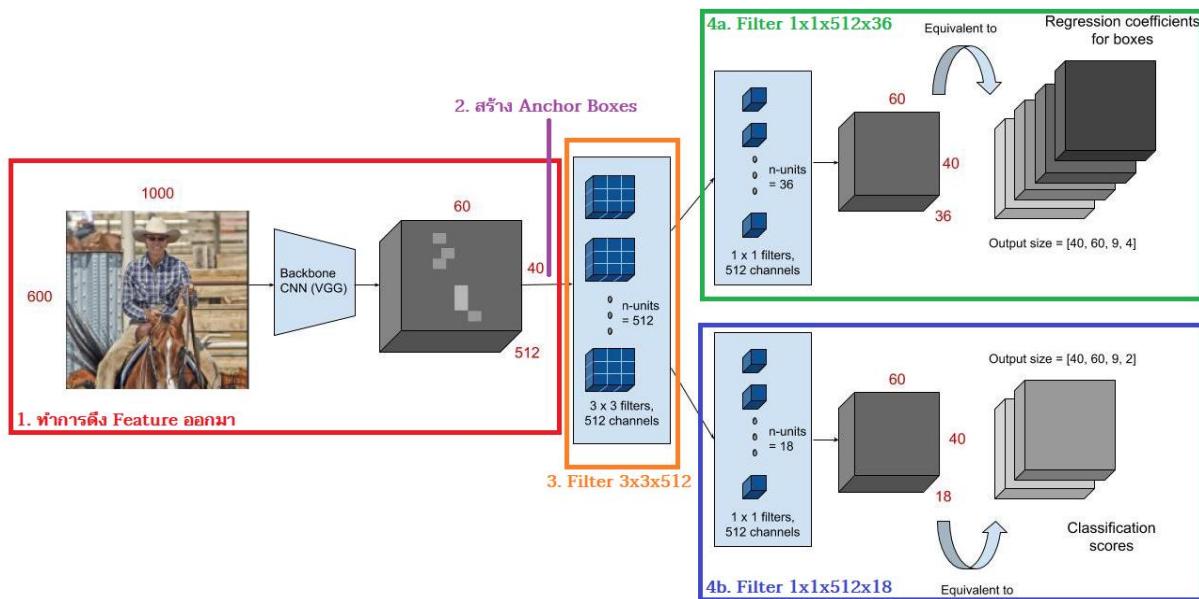
$$L_{classify}(p_i, p_i^*) = -\frac{1}{N_{classify}} \sum_i p_i^* \log(p_i) \quad (2)$$

$$L_{regression}(t_i, t_i^*) = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i^u - v_i) \quad (3)$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (4)$$

$$\begin{array}{ll} t_x = \frac{(x - x_{anchor})}{w_{anchor}} & t_y = \frac{(y - y_{anchor})}{h_{anchor}} \\ t_w = \log\left(\frac{w}{w_{anchor}}\right) & t_h = \log\left(\frac{h}{h_{anchor}}\right) \\ t_x^* = \frac{(x^* - x_{anchor})}{w_{anchor}} & t_y^* = \frac{(y^* - y_{anchor})}{h_{anchor}} \\ t_w^* = \log\left(\frac{w^*}{w_{anchor}}\right) & t_h^* = \log\left(\frac{h^*}{h_{anchor}}\right) \end{array} \quad (5)$$

p_i คือ ค่าความน่าจะเป็นของ Anchor box ที่ i ที่จะเป็นวัตถุ,	p_i^* เป็น 1 ถ้า Anchor box เป็นวัตถุ และ 0 ถ้าไม่ใช่,
t_i คือ เวคเตอร์ที่แทนค่า 4 ตัวแปร ของ Bounding box,	t_i^* เป็นเวคเตอร์ของ Ground truth bounding box,
? a คือค่าจาก Anchor box,	?* เป็นค่า Ground truth

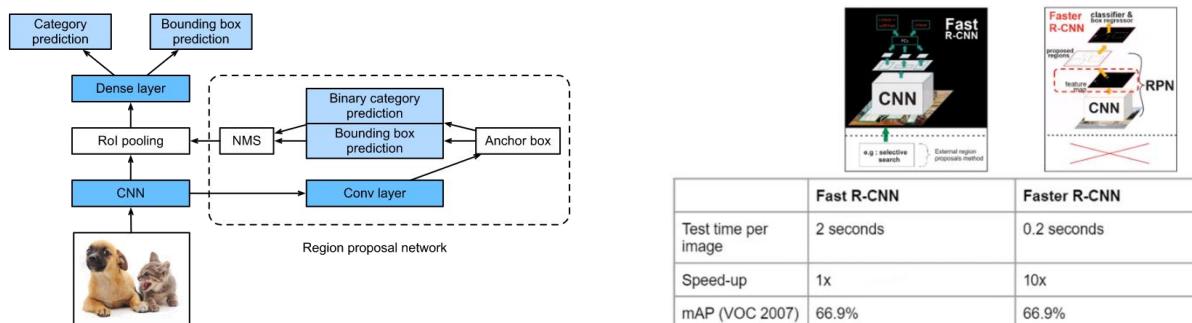


รูปที่ 6.19 – การทำ RPN ในแต่ละขั้นตอน แบ่งเป็นข้อๆ ตามการอธิบายข้างต้น

ที่มา [<https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46>]

6.2.8. Faster Region Convolutional Neural Network (Faster RCNN)

เป็นสถาปัตยกรรมที่ถูกพัฒนาต่อยอดขึ้นมาจาก Fast RCNN โดยความต่างของ Faster RCNN กับ Fast RCNN คือวิธีการหา ROI ที่เปลี่ยนจากการใช้ Selective Search มาเป็น Region Proposal Network (RPN) แทน^[17, 25-26] ดังรูปที่ 6.20 ทำให้เวลาในการประมวลผลดีขึ้นกว่าเดิมกว่า 10 เท่าจากเดิมดังรูปที่ 6.21



รูปที่ 6.21 – ผลการเปรียบเทียบระหว่าง Fast RCNN และ Faster RCNN

รูปที่ 6.20 – สถาปัตยกรรมของ Faster RCNN

ที่มา [https://d2l.ai/chapter_computer-vision/rcnn.html]

และ Faster RCNN

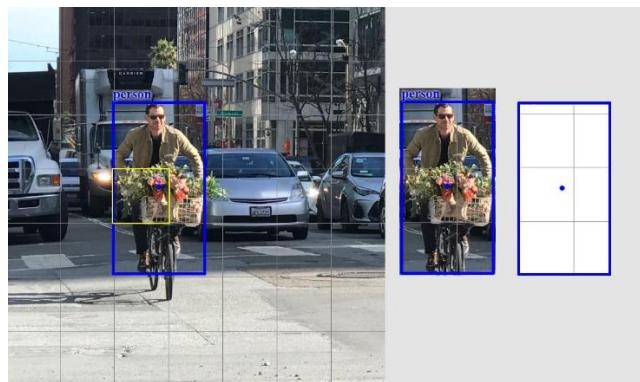
ที่มา

[<https://www.youtube.com/watch?v=v5bFVbQvFRk>]

6.2.9. You Only Look Once version 3 (YOLOv3)

เป็นอัลกอริทึมที่นิยมใช้ในการสร้างระบบตรวจจับวัตถุโดยตัว yolo นั้นไม่ใช้อัลกอริทึมที่เน้นไปทางให้ความแม่นยำในการตรวจจับวัตถุกับระบบ แต่สามารถให้ผลลัพธ์ที่ถูกใจได้ในระยะเวลาอันสั้นและไม่สูญเสียความแม่นยำไปมากนักเมื่อเทียบกับอัลกอริทึมในตระกูล RCNN โดยมีระบบการทำงานดังนี้^[25]

ระบบเมื่อทำการรับข้อมูลรูปภาพเข้ามาแล้วจะแบ่งรูปภาพออกเป็นหลายส่วนโดยแบ่งเป็น grid เป็นจำนวน $S \times S$ โดยในแต่ละ grid cell นั้นจะแทนข้อมูลของการทำนายว่าเป็นวัตถุชนิดใดได้เพียงชนิดเดียว



รูปที่ 6.22 – ตัวอย่างการแบ่ง grid cell

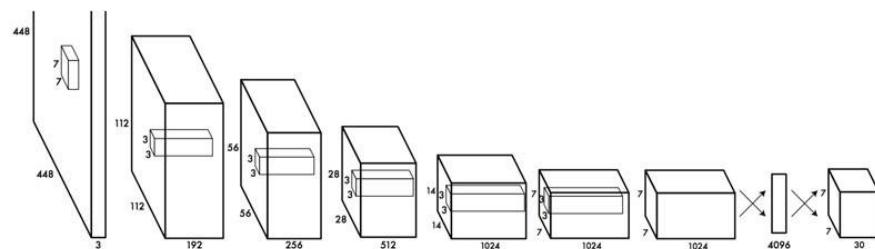
ที่มา [https://miro.medium.com/max/1424/1*6qZXYCDUkC5Bc8nRoIT0Mw.jpeg]

อย่างในรูปที่ 6.22 จะสามารถทำนายเป็นคนได้โดยมีตำแหน่งศูนย์กลาง(จุดสีน้ำเงิน) อยู่ใน grid cell นั้น ซึ่งในแต่ละ grid cell อัลกอริทึมจะทำนายขนาดของกรอบรูปภาพที่คลุมอยู่อย่างเช่นในรูป 6.22 กรอบนั้นก็จะเป็นกรอบที่คลุมตัวคน

โดยสรุปนั้นแต่ละ grid cell จะมีลักษณะดังนี้

- แต่ละ grid cell นั้นจะแทนด้วยวัตถุจำนวนชนิดเดียว
- แต่ละ grid cell นั้นจะทำนายขนาดของกรอบรูปภาพมาคือ จุดศูนย์กลาง(x,y) และความสูงยาว(w,h)
- แต่ละ grid cell จะมีค่าที่บ่งบอกถึงความน่าจะเป็นที่ grid cell จะมีรูปคือ box confidence score
- แต่ละ grid cell จะมีค่า condition class probabilities เพื่อบอกความน่าจะเป็นว่าวัตถุใน grid cell นั้นเป็นวัตถุชนิดใด

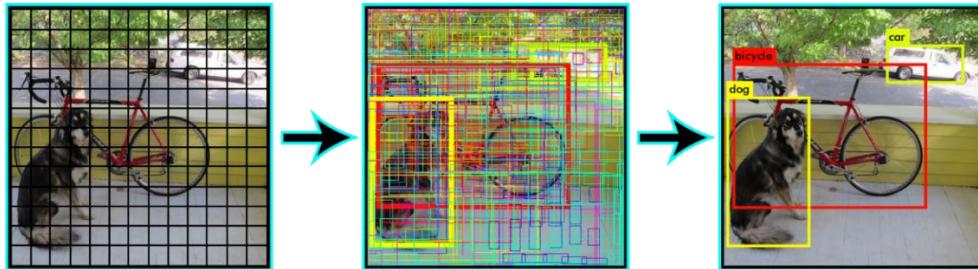
โดยตัวสถาปัตยกรรมของ YOLO นั้นจะเป็นดังรูปที่ 6.23



รูปที่ 6.23 – สถาปัตยกรรม YOLO

ที่มา [https://miro.medium.com/max/2400/1*9ER4GVUtQGVA2Y0skC9OOQ.png]

YOLO เป็นสถาปัตยกรรมที่มี 24 convolutional layers และตามด้วย 2 fully connected layers ซึ่งใน layer สุดท้ายของ convolutional layer จะได้ข้อมูลที่มีขนาด $(7, 7, 1024)$ เพื่อทำการ flatten และนำไปเข้า fully connected layers เพื่อให้ได้ output สุดท้ายของมีขนาด $(7, 7, 30)$ ซึ่งในการตัดขั้นตอนสุดท้ายก่อนที่โมเดลจะส่งข้อมูลออกไปโมเดลจะทำการเลือกเฉพาะ grid cell ที่ค่า confidence scores สูงเท่านั้น (มากกว่า 0.25) ดังรูปที่ 6.24



รูปที่ 6.24 – output ของ YOLO

สำหรับ loss function ของ YOLO นั้นจะแบ่งออกเป็น 3 พังก์ชันได้แก่

- Classification loss ในกรณีที่วัตถุถูก detect ได้ใน grid cell นั้น loss function ของมันจะเป็น squared error มีสมการดังนี้

$$\sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

$$\mathbb{I}_i^{obj} = 1 \text{ ถ้า grid cell } i \text{ พบวัตถุ}$$

$$\text{นอกนั้น } \mathbb{I}_i^{obj} = 0$$

$\hat{p}_i(c)$ แทนด้วยความน่าจะเป็นของ class c ใน cell i

- Localization loss จะเป็นการวัด error ในการคำนวณ bounding box หรือ กรอบรูป ซึ่งจะเป็นค่าของ ตำแหน่ง (x, y) และขนาด(w, h) ซึ่ง loss function นี้จะคิดก็ต่อเมื่อวัตถุถูก detect ใน grid cell นั้นซึ่ง มีสมการดังนี้

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_i^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_i^{obj} [(\sqrt{w}_i - \sqrt{\hat{w}_i})^2 + (\sqrt{h}_i - \sqrt{\hat{h}_i})^2]$$

$$\mathbb{I}_i^{obj} = 1 \text{ ถ้ากรอบรูป } j \text{ th ใน grid cell } i \text{ พบรูป } \text{นอกนั้น } \mathbb{I}_i^{obj} = 0$$

λ_{coord} เป็นค่าคงที่ ที่ช่วยเพิ่ม loss weight ของกรอบ bounding box

- Confidence loss

เป็นการวัดความน่าจะเป็นที่ใน grid cell นั้นๆ จะมีวัตถุอยู่ใน grid cell หรือเรียกว่าค่า objectness โดย เมื่อเจอวัตถุใน grid cell จะมีสมการดัง (1) และเมื่อไม่เจอวัตถุใน grid cell จะมีสมการดัง (2)

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} (C_i - \hat{C}_i)^2$$

\hat{C}_i เป็นค่า confidence score ของกรอบ j ใน grid cell i

$\mathbb{I}_{ij}^{obj} = 1$ ถ้ากรอบรูป $j th$ ใน grid cell i พบรัตตุ นอกนั้น $\mathbb{I}_{ij}^{obj} = 0$

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

\mathbb{I}_{ij}^{noobj} เป็นส่วนกลับ \mathbb{I}_{ij}^{obj}

\hat{C}_i เป็นค่า confidence score ของกรอบ j ใน grid cell i

λ_{coord} เป็นค่าคงที่ ที่ลดเพิ่ม loss weight เมื่อตรวจจับรูปได้เป็นพื้นหลัง

ซึ่งใน YOLOv3 จะมีอัลกอริทึมที่เปลี่ยนไปดังนี้

YOLOv3 ทำการแทนที่ softmax function ด้วย independent logistic เพื่อกำหนณความแม่นยำของรูปที่รับเข้ามากับวัตถุชนิดต่างๆแบบพิเศษ นอกจากนี้ยังใช้ binary cross-entropy ในการคำนวณ classification loss แทนด้วย mean square error เพื่อลดความซ้ำซ้อนของโมเดลให้น้อยลง

YOLOv3 เปลี่ยน feature extractor จาก Darknet-19 มาเป็น Darknet-53 โดย Darknet-53 ประกอบไปด้วย 3×3 และ 1×1 filter นอกจากนั้นยังมีการใช้ skip connections เชื่อมกับ residual network โดยมีสถาปัตยกรรมดังรูปที่ 6.25

Type	Filters	Size	Output
Convolutional	32	3×3	256×256
Convolutional	64	$3 \times 3 / 2$	128×128
1x	32	1×1	
	64	3×3	
	Residual		128×128
Convolutional	128	$3 \times 3 / 2$	64×64
2x	64	1×1	
	128	3×3	
	Residual		64×64
Convolutional	256	$3 \times 3 / 2$	32×32
8x	128	1×1	
	256	3×3	
	Residual		32×32
Convolutional	512	$3 \times 3 / 2$	16×16
8x	256	1×1	
	512	3×3	
	Residual		16×16
Convolutional	1024	$3 \times 3 / 2$	8×8
4x	512	1×1	
	1024	3×3	
	Residual		8×8
Avgpool		Global	
Connected		1000	
Softmax			

รูปที่ 6.25 – สถาปัตยกรรม YOLOv3

ที่มา [https://miro.medium.com/max/418/1*biRYJyCSv-UTbTQTa4Afqg.png]

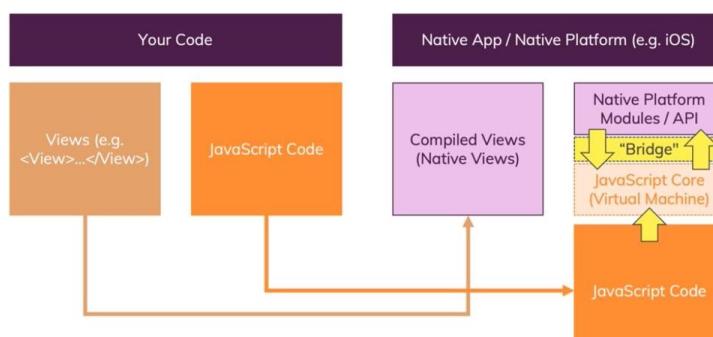
6.2.10. Cross Platform

กลุ่มของเราเลือกที่จะพัฒนาแอปพลิเคชันบนมือถือ เพราะส่วนแบ่งตลาดของระบบปฏิบัติการยอดนิยมในตอนนี้จะเป็นระบบปฏิบัติการ Android กับ iOS^[14] แต่เดิม การที่จะพัฒนาแอปพลิเคชันเพื่อใช้งานบนระบบปฏิบัติการทั้งสอง ต้องเขียนด้วยภาษาหลักหรือเนทฟอยย่าง Java หรือ Kotlin สำหรับ Android และ

Objective C สำหรับ iOS จึงจำเป็นต้องใช้เวลาจำนวนมากในการเรียนรู้ภาษาทั้งสอง ดังนั้นการใช้ Mobile Cross-Platform technology จึงเป็นตัวเลือกที่น่าสนใจ โดย Concept ของ Cross-Platform ก็คือเราสามารถพัฒนาแอปพลิเคชันบนแพลตฟอร์มใด ๆ ก็ได้ โดยใช้แค่เพียงภาษาเดียว ซึ่งเป็นภาษาที่จะถูกแปลงไปเป็นภาษาหลักหรือก็คือภาษาเนทีฟที่แต่ละแพลตฟอร์มเข้าใจ

สำหรับ Cross-platform Technology ที่ทางทีมพัฒนาจะเลือกใช้ก็จะเป็น React Native โดยเขียนภาษา JavaScript (ES6) ร่วมกันกับ XML markup อย่าง JSX (การเขียน HTML Tag ผสมกับ JavaScript) เพื่อบอกว่าโครงสร้างแอปพลิเคชันจะมีการจัดวางโครงสร้างเป็นอย่างไร จากนั้น React Native ก็จะไปเรียก Native API เพื่อ Render Component ของแต่ละแพลตฟอร์มตามโครงสร้างที่เขียนไว้ ทำให้องค์ประกอบหลักของแอปพลิเคชันนั้นมาจากการ Render Component จริง ๆ ของแต่ละแพลตฟอร์ม เพียงแต่เขียนด้วยภาษา React Native ภาษาเดียว

ข้อเสียของ Cross-platform อย่าง React-native คือมันทำได้แค่เพียงช่วยให้ Developer สามารถพัฒนาแอปพลิเคชันหลากหลายแพลตฟอร์มได้โดยใช้ภาษาเดียว ไม่ใช้การเขียนรอบเดียวแล้วนำไปรันได้หลายระบบปฏิบัติการ ดังนั้น developer ยังมีความจำเป็นที่จะต้องพัฒนาแอปพลิเคชันให้สอดคล้องกับความเฉพาะตัวของแพลตฟอร์มแต่ละชนิด ตัวอย่างเช่น ขนาดหน้าจอที่ไม่เท่ากัน user interface ของแต่ละระบบปฏิบัติการ สำหรับอีกข้อเสียหนึ่งของ Cross-platform ก็คือจะเร็วไม่เท่ากับการเขียนลงบน Native จริง ๆ เนื่องจากการ Optimize Resource บน Native นั้นสามารถทำได้ดีกว่า อ้างอิงจากผลลัพธ์การทำ Benchmark พบว่า React Native กินทรัพยากร CPU มากกว่า Native ประมาณ 15 % โดยเฉลี่ย^[15]



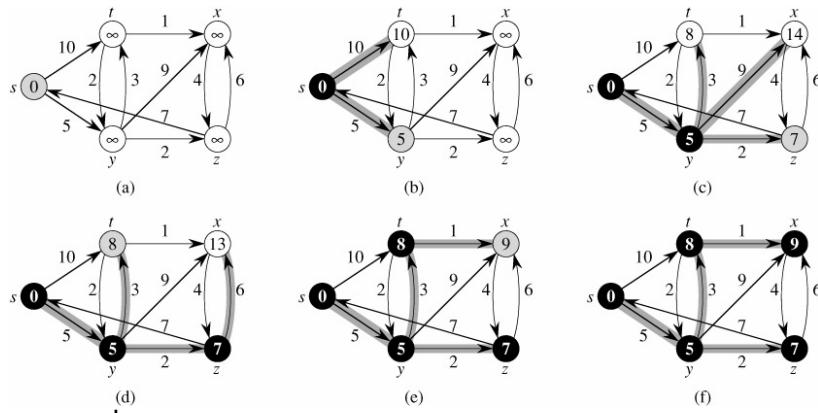
รูปที่ 6.26 – How React Native work behind the scene

ที่มา [Course: React Native - The Practical Guide by Maximilian Schwarzmüller]

6.2.11. Dijkstra's Algorithm

เป็น Algorithm ที่ใช้ในการหาเส้นทางที่สั้นที่สุดระหว่าง node ซึ่ง Algorithm นี้จะถูกนำมาประยุกต์ใช้กับส่วนของ Path Optimization โดยมีขั้นตอนดังต่อไปนี้^[26]

- ทำการสร้างตัวแปร array ไว้เก็บระยะห่างของแต่ละ node โดยทุก node ตอนแรกจะมีค่าเป็นอนันต์
- ทำการเลือก node ที่สามารถไปต่อได้จากจุดที่อยู่และสั้นที่สุด และทำการเปรียบเทียบกับค่าในตัวแปร array หากมีค่าที่น้อยกว่าค่าข้างใน array ให้ทำการทับค่าใหม่เข้าไปแทน
- ทำซ้ำไปเรื่อยๆ จนกว่าจะได้ครบทุก node หรือถึงจุดหมาย



รูปที่ 2.27 – ภาพการดำเนินการของ Dijkstra's Algorithm

ที่มา [https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSDkt4XHALypl1ywXYrX_J3_reRFAuYJdPaSGExpL8Y5BLFxDo]

6.3. เครื่องมือที่ใช้ในการพัฒนา

- **TensorFlow** เครื่องมือที่ใช้ในการสร้างระบบตรวจสอบวัตถุ (Object Detection) ที่สามารถแยกแยะ และตรวจสอบวัตถุได้ว่าเป็นวัตถุชนิดใดโดยใช้หลักการของการเรียนรู้เชิงลึกของคอมพิวเตอร์ (Deep Learning) การสร้างระบบตรวจสอบวัตถุดังกล่าวจะสามารถทำให้สามารถแยกแยะและระบุชนิดของภาพถ่ายขยะได้
- **React Native** เครื่องมือที่ใช้ในการสร้าง Cross platform Mobile Application ที่จะช่วยในการสร้าง ทั้งแอปพลิเคชันบนระบบปฏิบัติการ iOS และ Android ซึ่งการสร้างแอปพลิเคชันที่เป็น Cross platform Mobile Application นั้นสามารถทำให้ผู้พัฒนาพัฒนาโปรแกรมเพียงครั้งเดียว ก็สามารถใช้งานได้ทุก ระบบ โดยภาษาที่ใช้นั้นคือ Javascript
- **VueJS** เครื่องมือที่ใช้ในการสร้างเว็บไซต์ โดยใช้ภาษา Javascript ซึ่งจะเข้ามาช่วยในการจัดการ โครงสร้างของเว็บไซต์
- **ChartJS** เครื่องมือ Open source ในการสร้างกราฟในรูปแบบต่างๆ ใช้ในการทำส่วนของ Data Visualization เพื่อแสดงข้อมูลต่างๆ ภายใต้แอปพลิเคชัน
- **Firebase** เครื่องมือที่ใช้ในการจัดการข้อมูล Backend ของ Google ที่จะทำให้การจัดการ Backend ของแอปพลิเคชันได้ง่ายขึ้น โดยที่ Firebase จะมี Firestore สำหรับการจัดการฐานข้อมูล Storage สำหรับการเก็บไฟล์ต่างๆ Authentication สำหรับการทำระบบล็อกอินต่างๆ และ Cloud Functions สำหรับการอัปโหลดไฟล์ต่างๆ ขึ้นไป
- **Google Maps Service** เครื่องมือที่ใช้ในการหยิบแผนที่จาก Google มาแสดงสถานที่อยู่ของผู้ขายขยะ ให้ผู้รับซื้อขยะเดินทางไปได้อย่างสะดวกมากขึ้น และยังสามารถทำ Path Optimization ได้เพียงแค่ส่ง สถานที่ต่างๆ เข้าไปให้ Google วิเคราะห์หาเส้นทางที่ดีที่สุดออกมาให้เท่านั้น

- **Google Colab** – เป็น Cloud service ฟรีของ Google ที่สนับสนุนการใช้งาน GPU ถูกออกแบบมาสำหรับการทำงานเกี่ยวกับ Data Science และการเทรนโมเดลโดยเฉพาะ โดยภายในจะมี Library สำหรับการทำ Data Science และการเทรนโมเดล พร้อมใช้งาน โดย Cloud service นี้มีข้อจำกัดที่สามารถใช้งานต่อเนื่องได้เพียงแค่ 12 ชั่วโมงเท่านั้น หากใช้มากกว่านั้น จะเป็นต้องรอเป็นระยะเวลาหนึ่งถึงจะกลับมาใช้งานได้

6.4. รายละเอียดโปรแกรมที่ได้พัฒนาในเชิงเทคนิค

6.4.1 Input/Output Specification

ในฝั่งของคนขายขยะ แอปพลิเคชันจะสามารถรับข้อมูลรูปภาพจากกล้องหรือคลิปรูปของสมาร์ทโฟน แล้วนำไปประมวลผลเพื่อหาผลลัพธ์ออกมาเป็นประเภทขยะ จำนวนขึ้นของขยะ วิธีการจัดการ และตัวเลือกในการขายหรือบริจาค โดยการบอกประเภทและจำนวนขึ้นของขยะจะถูกประมวลผลผ่าน Cloud และส่งคำตอบกลับมา และการบอกวิธีการจัดการขยะขึ้นน้ำหากไม่สามารถขายได้ โดยจะบอกเป็นประเภทถังขยะที่ควรทิ้งลงเป็นจังจิกฐานข้อมูล

ในฝั่งของคนรับซื้อขยะ แอปพลิเคชันจะแสดงคนที่ต้องการจะขายขยะ โดยจะมีบอร์ดรายละเอียดทั้งประเภทและจำนวนของขยะที่ขาย เมื่อกดรับซื้อขยะแล้วก็จะแสดงเส้นทางในการเดินทางจากที่อยู่ปัจจุบันไปยังสถานที่รับซื้อขยะ โดยถ้าหากรับซื้อขยะไว้หลายที่ ก็จะแสดงเส้นทางที่ดีที่สุดในการเดินทางไปยังสถานที่ต่างๆ โดยใช้ Google Maps Service

6.4.2. Functional Specification

Waste Detection Model เป็นพักร์ชันที่สามารถระบุประเภทและจำนวนของขยะได้โดยการใช้ Deep Learning

- สามารถใช้รูปภาพที่เป็นรูปถ่ายจากกล้องสมาร์ทโฟน
- สามารถให้คำตอบเป็นประเภทขยะ และแสดงตัวเลือกในการจัดการต่อไป
- สามารถบอกจำนวนสิ่งของได้
- สามารถส่งผลลัพธ์จากการทำนายได้ภายใน 2 วินาที

Marketplace System เป็นระบบการซื้อขายขยะที่รองรับการใช้งานทั้งผู้ซื้อและขายขยะ

- สามารถสมจำนวนขยะไว้ แล้วเลือกที่จะขายที่เดียวได้
- สามารถรองรับระบบขายขยะตามคำร้องขอ กับผู้ขายขยะ ผู้ขายขยะจะสามารถตัดสินใจเลือกผู้รับซื้อขยะ เป็นคราวๆได้ โดยกดส่ง request ไปให้กับคนรับซื้อขยะเอง
- สามารถรองรับระบบขายขยะแบบเร่งด่วน โดยจะให้ผู้ขายขยะตั้งไว้ แล้วจะให้ผู้รับซื้อขยะในบริเวณนั้นมาซื้อขยะไป โดยแข่งกันที่ผู้รับขยะคนใดกดซื้อก่อนก็จะได้ไป
- สามารถรองรับระบบในการบริจาคขยะตามแคมเปญที่ผู้ใช้งานสนใจ
- สามารถแสดงความคิดเห็นหรือคำวิจารณ์ต่อผู้รับซื้อขยะแต่ละเจ้าได้

- สามารถย้อนกลับไปดูประวัติการขายของตัวเองได้ โดยจะมีข้อมูลทั้งราคา ผู้รับซื้อ และสิ่งที่ขายไป
- Path Optimization** เป็นการเพิ่มประสิทธิภาพในการเดินทางของผู้รับซื้อขาย ในกรณีที่กรับซื้อจากผู้ขายหลายๆ แห่ง ทำให้ผู้รับซื้อขายไม่จำเป็นต้องเดินทางอ้อมไปอ้อมมาและสะดวกมากยิ่งขึ้น
- สามารถแสดงเส้นทางที่สั้นหรือดีที่สุดตามเงื่อนไขของผู้รับซื้อขาย
- สามารถบอกเส้นทางได้ในระยะเวลาที่ไม่จำกัดโดยวิถี

Visualization Function เป็นฟังก์ชันที่ไว้แสดงขยะประเภทต่างๆ จากฐานข้อมูลภายในแอปพลิเคชัน เพื่อให้นักวิจัยหรือผู้มีความสนใจนำข้อมูลเหล่านี้ไปใช้ประโยชน์ในลำดับถัดไป ซึ่งฟังก์ชันนี้จะถูกใช้งานนอกตัวแอปพลิเคชัน ทำให้สามารถเข้าถึงข้อมูลได้โดยไม่ต้องเข้าแอปพลิเคชัน

- สามารถแสดง Visual Graphic จากข้อมูลภายในแอปพลิเคชันได้

6.4.3. โครงสร้างซอฟต์แวร์



รูปที่ 6.28 – แผนผังโครงสร้างของโปรแกรม

Application – เป็นส่วนของแอปพลิเคชันที่ไว้สื่อสารกับผู้ใช้งาน และรองรับการใช้งานฟังก์ชันต่างๆ โดยที่ในส่วนนี้จะถูกพัฒนาขึ้นด้วยใช้ Framework ที่ชื่อว่า React Native ในการพัฒนาในส่วนของ Frontend และใช้ Cloud Function ใน Firebase เป็นสมือน Backend ของแอปพลิเคชัน โดยแบ่งเป็น 4 ส่วนดังนี้

- Waste Processing Signal เป็นส่วนที่ส่งคำร้องไปหา Cloud Computing เพื่อทำการทำความสะอาดจากรูปโดยใช้โมเดล Faster RCNN ภายใน Cloud
- Waste Storage Signal เป็นส่วนที่ส่งคำร้องไปหา Waste Storage ในส่วนของ Cloud Function เพื่อทำการเพิ่มขยะเข้าไปในคลังของผู้ขายขาย เพื่อนำไปขายในลำดับถัดไป
- Trading System Signal เป็นส่วนที่ส่งคำร้องไป Trading System ในส่วนของ Cloud Function เพื่อทำการซื้อขายขายและระบบภายในที่เกี่ยวข้องกับการซื้อขาย
- Path Optimization Signal เป็นส่วนที่ส่งคำร้องไป Path Optimization ในส่วนของ Cloud Function เพื่อเสนอเส้นทางในการเดินทางที่ดีที่สุด โดยสามารถจัดแต่งสถานที่ที่จะไปก่อนหลังได้ด้วย

Firebase – เป็นส่วน Backend และฐานข้อมูลของแอปพลิเคชันและเว็บไซต์ โดยจะแบ่งหน้าที่หลักๆ ออกเป็น 3 ส่วนดังนี้

- Cloud Functions จะเก็บฟังก์ชันต่างๆ ของแอปพลิเคชันเอาไว้ โดยจะมีหลักๆ ทั้งหมด 3 ฟังก์ชันคือ Waste Storage, Trading System และ Path Optimization

- Google Authentication จะทำหน้าที่ตรวจสอบผู้ใช้งาน และอนุมัติการเข้าใช้งานระบบ
- Firestore เป็นฐานข้อมูลของแอปพลิเคชัน โดยที่เป็นฐานข้อมูลประเภท NoSQL

Cloud Computing – เป็นการประมวลผลแบบ Serverless ที่เก็บโมเดลที่มีสถาปัตยกรรมแบบ Faster RCNN เอาไว้ โดยที่ตัวโมเดลถูก เตรนด์ขึ้นมาเพื่อทำการคำนวณและตรวจจับขยะภายในรูปภาพ

Google Maps Services – เป็นบริการของ Google ที่จะทำงานในส่วนของการแสดงแผนที่และ Route Optimization

Website – เป็นส่วนการแสดงผลของข้อมูลที่ได้ออกมาจาก Firestore โดยที่ทำออกแบบเป็น Data Visualization ให้สวยงาม

6.4.4. การพัฒนา Waste Detection

โครงการนี้จะมีส่วนของระบบการตรวจจับและจัดจำแนกเพื่อเพิ่มความสะดวกสบายและให้ความรู้แก่ผู้ใช้ ทางผู้จัดทำการได้ทำการศึกษาและค้นคว้าโดยมี 2 สถาปัตยกรรมที่คิดว่าจะเหมาะสมกับระบบนี้ได้แก่ Faster Region Convolutional Neural Network (Faster RCNN) และ You Only Look Once version 3 (YOLOv3) โดยจะนำมาทดสอบและเลือกสถาปัตยกรรมที่ให้ผลลัพธ์ที่ดีที่สุด

สำหรับ Faster RCNN ผู้จัดทำได้นำ source code มาจาก https://github.com/RockyXu66/Faster_RCNN_for_Open_Images_Dataset_Keras จากนั้นผู้จัดทำการเปลี่ยนแปลงบางส่วนรวมถึงส่วนของ Backbone หรือ Architecture ของ CNN layer ให้เป็น Resnet-50 (ในตัว source code ตอนแรกเป็น VGG-16) ในส่วนของ YOLOv3 ผู้จัดทำได้นำ source code มาจาก <https://pjreddie.com/darknet/yolo>

6.4.4.1. ข้อมูลและการเตรียมข้อมูล

ข้อมูลที่ใช้ในการเทรนด์โมเดลมีจำนวนรูปภาพทั้งหมด 6100 รูป และทำการสร้างกรอบรูปใบแต่ละรูปเพื่อบอกถึงวัตถุที่อยู่ในรูปภาพแต่ละรูป โดยมีรายละเอียดดังนี้

ตารางชุดข้อมูลขยะที่คณะผู้จัดทำสร้างขึ้นมา			
ประเภทขยะ	จำนวนกรอบรูป	ประเภท	จำนวนกรอบรูป
ถ่านไฟฉาย	841	แก้วพลาสติก	676
ขยะอิเล็กทรอนิก	455	ไฟม	421
หลอดไฟ	446	ห่อ PVC	237
ขวดแดง	452	กระดาษขาว	375
ขวดขาว	534	กล่องนม	1194
ขวดสเปรย์	556	หนังสือพิมพ์	692
ขวดพลาสติก	1405	กล่องลัง	539
ขวดน้ำ	299	กระป๋อง	896
		รวม	10018



รูปที่ 6.29 – ตัวอย่างรูปภาพพัฒนา

นอกจากนี้ผู้จัดทำได้ทำการแบ่งรูปภาพออกเป็น 2 ส่วน ส่วนแรกสำหรับการเทรนโมเดล จำนวน 5620 รูป และส่วนที่สองสำหรับการทดสอบโมเดล จำนวน 480 รูป โดยมีรายละเอียดดังนี้

ตารางชุดข้อมูลขยะที่จะถูกนำไปเทรนร์			
ประเภทขยะ	จำนวนกรอบรูป	ประเภทขยะ	จำนวนกรอบรูป
ถ่านไฟฉาย	751	แก้วพลาสติก	646
ขยะอิเล็กทรอนิก	395	โฟม	391
หลอดไฟ	393	ห่อ PVC	206
ขวดเดง	422	กระดาษขาว	344
ขวดขาว	504	กล่องนม	1096
ขวดสเปรย์	508	หนังสือพิมพ์	575
ขวดพลาสติก	1374	กล่องลัง	507
ขวดน้ำ	269	กระป๋อง	822
		รวม	9228

ตารางชุดข้อมูลขยะที่จะถูกนำไปทดสอบ			
ประเภทขยะ	จำนวนกรอบรูป	ประเภทขยะ	จำนวนกรอบรูป
ถ่านไฟฉาย	90	แก้วพลาสติก	30
ขยะอิเล็กทรอนิก	36	โฟม	30
หลอดไฟ	53	ห่อ PVC	31
ขวดเดง	30	กระดาษขาว	31
ขวดขาว	30	กล่องนม	98
ขวดสเปรย์	48	หนังสือพิมพ์	117
ขวดพลาสติก	31	กล่องลัง	32
ขวดน้ำ	30	กระป๋อง	74
		รวม	790



รูปที่ 6.30 – ตัวอย่างภาพที่นำไปทดสอบ

สำหรับการเตรียมข้อมูลสำหรับการเทรนโมเดลจะถูกทำการ Resize และ Augmentation โดยที่แต่ละรูปภาพให้มีด้านที่สั้น มีความยาวเป็น 600 และให้ด้านที่ยาวมีความยาวตามอัตราส่วนของรูปภาพนั้นยกตัวอย่างเช่น ภาพยาว 1200x1500 ก็จะถูก Resize เป็น 600x750 และในส่วนของ Augmentation จะทำการสุ่มตัดแบ่งรูป โดยจะมีการ Horizontal flip, Vertical flip และ Rotation 90, 180, 270 องศาเพื่อให้การดึง Feature ออกมาจากรูปภาพได้อย่างมีประสิทธิภาพมากขึ้น

6.4.4.2. การเลือกโมเดล

วิธีการดำเนินการ

ในส่วนของการพัฒนาโมเดลเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดผู้จัดทำจะทำการทดลองโดยแบ่งออกเป็น 2 สถาปัตยกรรมที่ได้ทำการศึกษามาได้แก่ Faster-RCNN และ YOLOv3 โดยจะทดสอบความความสามารถในการทำงานและเลือกสถาปัตยกรรมที่ให้ผลลัพธ์ที่ดีที่สุด

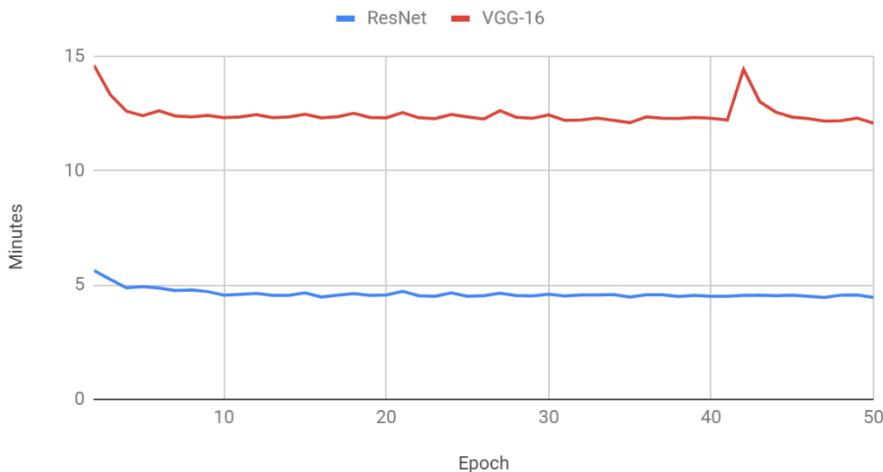
- **Faster-RCNN** จะประกอบไปด้วยส่วนของ CNN ซึ่งทำ Image Classification และ RPN ซึ่งทำ Object Detection สำหรับสถาปัตยกรรม CNN จะทดลองเลือกใช้ระหว่าง VGG-16 กับ ResNet การทดลองว่าจะเลือกใช้สถาปัตยกรรม CNN ตัวไหนก็จะพิจารณาจากความเร็วที่ใช้ และค่า Loss ที่ได้

จากการลองเทรนโดยใช้ Pre-trained โมเดล VGG-16 และ ResNet ด้วยรูปทดลอง ประเภท 100 รูปภาพจำนวน 50 epochs เพื่อทำการคุณลักษณะ พบร่วมเวลาที่ใช้ในการเทรนด์ต่างกันอย่างมีนัยสำคัญ โดยที่ VGG ใช้เวลาโดยเฉลี่ยประมาณ epoch ละ 12.45 นาที แต่ ResNet ใช้เวลาเพียงแค่ 4.62 นาทีเท่านั้น ดังที่แสดงในรูปที่ 6.31 แต่สำหรับค่า Loss รวมแล้ว VGG-16 และ ResNet มีความแตกต่างที่สั้นเกตเได้ แต่ยังไม่ชัดเจนเมื่อเทียบกับเวลาที่ใช้ในการเทรนด์ ดังรูปที่ 6.32 โดยที่ค่า Loss ของ VGG-16 ใน Epoch ที่ดีที่สุดเป็น 0.786 และสำหรับ ResNet ใน Epoch ที่ดีที่สุดเป็น 0.703 ซึ่งเมื่อคิดเป็นเปอร์เซ็นแล้ว ResNet ดีกว่า VGG-16 ถึง 10.56%

โดยที่สาเหตุที่ทำให้ ResNet เทรนด์ได้เร็วกว่า VGG-16 อย่างเห็นได้ชัดเจนนั้น เป็นเพราะจำนวน Floating Point Operations Per Second (FLOPS) ที่แตกต่างกันระหว่าง VGG-16 ที่มีทั้งหมด

15.3 พันล้าน FLOPS ขณะที่ ResNet มีเพียงแค่ 3.8 พันล้าน FLOPS เท่านั้น^[24] ทำให้เวลาที่ใช้ในการคำนวณน้อยกว่าอย่างเห็นได้ชัด

Time taken in each epoch



ค่าทางสถิติที่น่าสนใจ

Average time:

VGG-16: 12.45

ResNet: 4.62

$$\text{Difference: } \frac{12.45}{4.62} = 2.69$$

Loss Comparison



ค่าทางสถิติที่น่าสนใจ

Least loss:

VGG-16: 0.786

ResNet: 0.703

Difference:

$$1 - \frac{0.703}{0.786} = 10.56\%$$

Average

ResNet/VGG: 8.51%

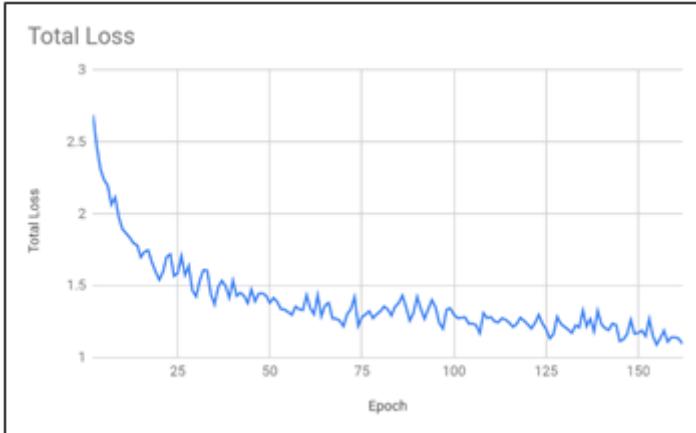
รูปที่ 6.32 – กราฟค่า Loss ในการเทรนด์แต่ละ epoch ของ Pre-trained โมเดล

เมื่อสามารถเลือก CNN ได้แล้วก็จะนำไปประกอบกับ RPN การเทรนด์โมเดล Faster-RCNN จะเป็นการเทรนด์ RPN ด้วย จากการเทรนด์ทั้งหมด 160 epochs ผลปรากฏว่าโมเดลที่มีค่า Total loss น้อยที่สุด (ผลรวมจาก loss ทั้ง 4 รูปแบบ) ได้แก่ โมเดลจาก epoch ที่ 155 โดยมีค่า loss ดังต่อไปนี้

1. RPN Classification Loss = 0.6
2. RPN Regression Loss = 0.139
3. Classification Loss (Fast R-CNN module) = 0.251
4. Regression Loss (Fast R-CNN module) = 0.1

5. Total Loss = 1.09

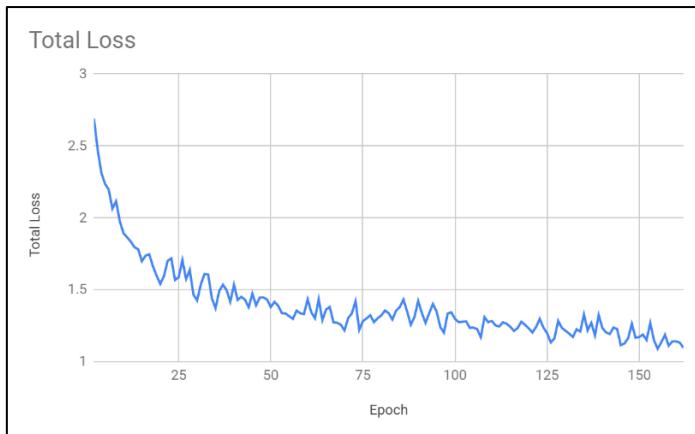
โดยกราฟผลที่ได้จากการเทรนด์ในแต่ละ epoch ดูได้จาก รูปภาพที่ 4.5



รูปที่ 6.33 – กราฟ Total loss (รวมทั้ง 4 loss) ในการเทรนด์แต่ละ epoch

ซึ่งทางผู้จัดทำได้ทำการเทรนด์ข้อมูลบน Google Colab ซึ่งใช้ GPU Tesla K80 ในการประมวลผลซึ่งใช้เวลาในการประมวลผลเฉลี่ย epoch ละประมาณ 17 นาที

- YOLOv3 จากการลองเทรนด์ พบร่วมกับการลดโมเดลเป็นจำนวน 160 epochs ได้ค่า loss ทั้งหมดที่ดีที่สุดอยู่ที่ epoch ที่ 155 คือ loss = 8.391512



รูปที่ 6.34 – กราฟ Total loss ในการเทรนด์แต่ละ epoch

ซึ่งทางผู้จัดทำได้ทำการเทรนด์ข้อมูลบน Google Colab ซึ่งใช้ GPU Tesla K80 ในการประมวลผลซึ่งใช้เวลาในการประมวลผลเฉลี่ย epoch ละประมาณ 15 นาที

การทดสอบและประเมินผล

โดยผลลัพธ์ที่ได้จากการทำข้อมูลมาทดสอบจะใช้ metric ในการวัดก็คือ Average Precision (AP) ซึ่งคำนวนได้จากสมการ $AveragePrecision = \frac{\sum rP@r}{R}$, $Precision = \frac{TP}{TP+FP}$ และ $Recall = \frac{TP}{TP+FN}$ ซึ่งผลการทดสอบออกมาเป็นดังนี้

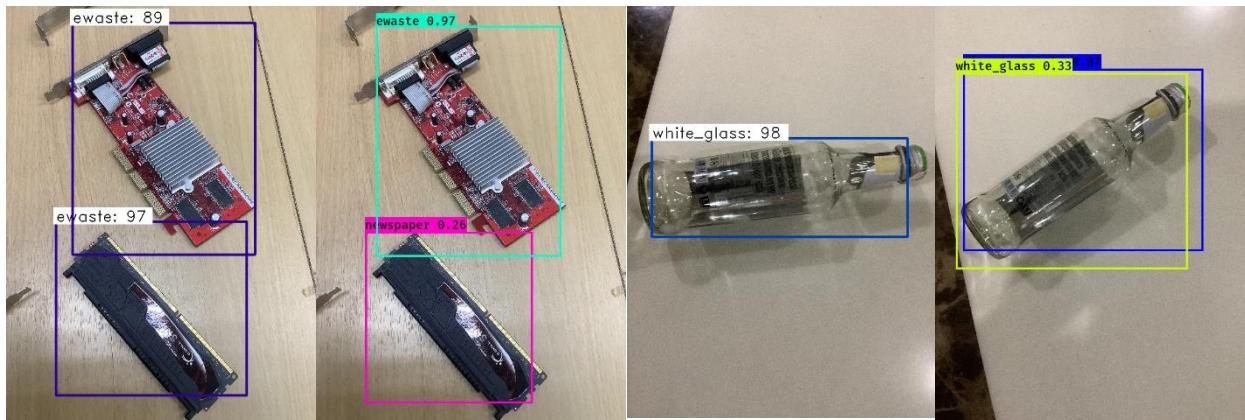
Faster-RCNN				YOLOv3			
ประเภทขยะ	AP	ประเภทขยะ	AP	ประเภทขยะ	AP	ประเภทขยะ	AP
ถ่านไฟฉาย	0.73	แก้วพลาสติก	0.97	ถ่านไฟฉาย	0.99	แก้วพลาสติก	1.0
e-waste	0.81	โฟม	0.98	e-waste	0.99	โฟม	1.0
หลอดไฟ	0.80	ท่อ PVC	0.80	หลอดไฟ	0.99	ท่อ PVC	1.0
ขวดแดง	0.90	กระดาษขาว	0.76	ขวดแดง	1.0	กระดาษขาว	0.99
ขวดขาว	0.85	กล่องนม	0.53	ขวดขาว	1.0	กล่องนม	0.98
ขวดสเปรย์	0.80	หนังสือพิมพ์	0.43	ขวดสเปรย์	0.99	หนังสือพิมพ์	0.90
ขวดพลาสติก	0.99	กล่องลัง	0.72	ขวดพลาสติก	1.0	กล่องลัง	0.99
ขวดนิ่ม	0.99	*mAP	0.799	ขวดนิ่ม	1.0	*mAP	0.991

จากการทั้งสองข้างบนจะเห็นได้ว่าตัวโมเดลที่สร้างมาจากสถาปัตยกรรม YOLOv3 นั้นให้ประสิทธิ์ด้านความแม่นยำได้ดีกว่าสถาปัตยกรรม Faster-RCNN อย่างมาก ซึ่งจากการลองนำตัวโมเดลไปทดสอบใช้ในสถานการณ์จริงปรากฏว่ามีตัวโมเดลของ YOLOv3 นั้นไม่ได้มีความสามารถที่ดีอย่างผลการทดสอบที่สอบที่ออกมา ทั้งนี้น่าจะเกิดจากปัญหา overfitting ของตัวโมเดลที่ทำให้รูปภาพที่มีลักษณะที่แตกต่างจากตัว dataset เมื่อนำมาเข้าสู่โมเดลแล้วได้ผลลัพธ์ที่ไม่ถูกต้อง

นอกจากนั้นทางผู้จัดทำได้ทำการนำโมเดลทั้งสองไปทดสอบเปรียบเทียบในสถานการณ์จริงได้ตัวอย่างผลลัพธ์ดังต่อไปนี้

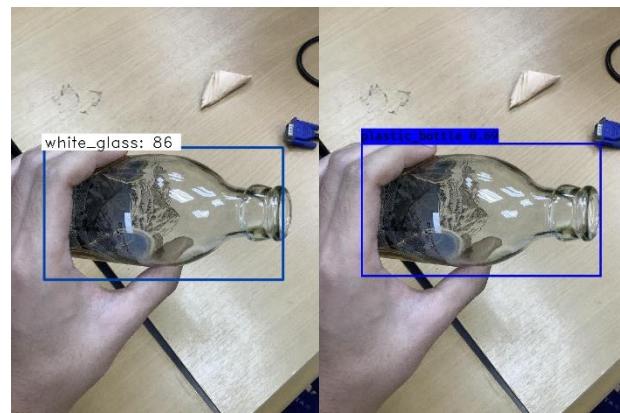


รูปที่ 6.35 – รูปเปรียบเทียบผลลัพธ์ระหว่าง Faster-RCNN(รูปซ้าย) กับ YOLOv3(รูปขวา) ที่ 1 รูปที่ 6.36 – รูปเปรียบเทียบผลลัพธ์ระหว่าง Faster-RCNN(รูปซ้าย) กับ YOLOv3(รูปขวา) ที่ 2



รูปที่ 6.37 – รูปเปรียบเทียบผลลัพธ์ระหว่าง Faster-RCNN(รูปซ้าย)
กับ YOLOv3(รูปขวา) ที่ 3

รูปที่ 6.38 – รูปเปรียบเทียบผลลัพธ์ระหว่าง Faster-RCNN(รูปซ้าย)
กับ YOLOv3(รูปขวา) ที่ 4



รูปที่ 6.39 – รูปเปรียบเทียบผลลัพธ์ระหว่าง Faster-RCNN(รูปซ้าย)
กับ YOLOv3(รูปขวา) ที่ 5

จากรูปภาพด้านบน รูปภาพทางซ้ายเป็นภาพที่ได้จากการนำนายของโมเดล Faster-RCNN และรูปภาพทางขวาเป็นภาพที่ได้จากการนำนายของโมเดล YOLOv3 ซึ่งจะเห็นได้ว่าตัวโมเดลทั้งสองมีความสามารถในการนำนายที่ใกล้เคียงกันแต่ในบางสถานการณ์นั้นตัวโมเดลของ Faster-RCNN จะนำนายได้แม่นยำกว่า YOLOv3

ซึ่งด้วยเหตุผลนี้ทางผู้จัดทำจึงเลือกใช้ตัวโมเดลของ Faster-RCNN เป็นโมเดลที่จะนำมาใช้ในระบบการคัดแยกขยะโดยการประมวลผลภาพของโครงการนี้

6.5. ขอบเขตและข้อจำกัด

- โทรศัพท์จะต้องใช้ระบบปฏิบัติการที่สามารถลงแอปพลิเคชัน Expo ได้ สำหรับ Android คือเวอร์ชัน 5.0 ขึ้นไป และสำหรับ iOS คือ iOS เวอร์ชัน 10 ขึ้นไป
- จะต้องสมัครบัญชีในแอปพลิเคชัน Expo เพื่อใช้งานการรับส่ง Notification จากแอปพลิเคชัน
- ต้องมีอินเตอร์เน็ตที่เสถียรในการใช้แอปพลิเคชัน

- การทำนายประเภทของโมเดลจะสามารถทำนายได้เพียงแค่ 16 ประเภท
- จำนวนประเภทภายในตอนนี้มีเพียงแค่ 16 ประเภทเท่านั้น
- เนื่องจาก Google API ที่ใช้ใน Path Optimization คิดเงินเป็นจำนวนครั้งที่เรียกใช้ ในตอนนี้จึงทำการจำกัดการใช้งานของฟังก์ชันนี้ไว้

7. กลุ่มผู้ใช้โปรแกรม

1. กลุ่มของผู้ขายขยะ ได้แก่ กลุ่มของผู้ใช้ทั่วไปที่มีพื้นที่ในการเก็บขยะ เนื่องจากทางคณะผู้จัดทำเห็นว่าใน การที่กลุ่มของผู้ใช้งานที่จะใช้งานแอปพลิเคชันนี้จะต้องมีพื้นที่ในการเก็บขยะไว้ก่อนที่จะเรียกใช้บริการให้ ผู้ซื้อขยะมารับซื้อไป
2. กลุ่มของผู้ซื้อขยะ ได้แก่ ร้านรับซื้อของเก่า หรือ ร้านรับซื้อขยะ ซึ่งเป็นกลุ่มที่ปกติแล้วเดินทางออกไปรับ ซื้อขยะเป็นปกติอยู่แล้ว

8. ผลการทดสอบโปรแกรม

การทดสอบแอปพลิเคชันต่อไปนี้เป็นการทดสอบแอปพลิเคชันโดยผู้จัดทำ โดยจะแบ่งออกเป็น 3 ส่วน การ ทดสอบโมเดล แอปพลิเคชัน และเว็บไซต์

8.1. การทดสอบโมเดล

- วิธีการดำเนินการ

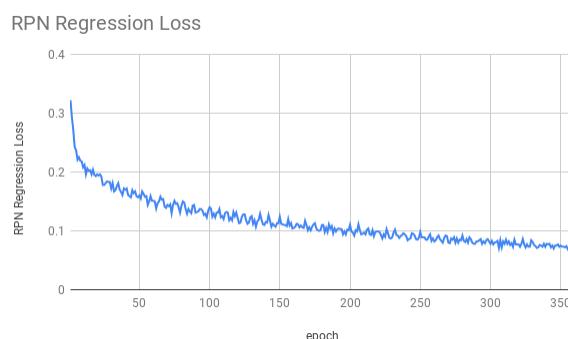
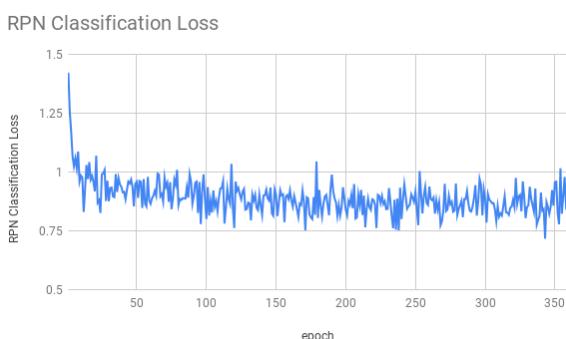
ผู้จัดทำได้เลือกใช้สถาปัตยกรรม Faster R-CNN(Resnet-50) ในการพัฒนาโมเดลโดยใช้ dataset ใน การดำเนินการทั้งหมดเป็นรูปขยะ 6100 รูป จำนวนกรอบรูปทั้งหมด 10018 ส่วน แบ่งชนิดของขยะได้ 16 ชนิด โดยแบ่งเป็น 2 ส่วนได้แก่

1. ส่วนของการเทรนด์ 5620 รูป เป็นกรอบรูปทั้งหมด 9228 กรอบรูป
2. ส่วนของการทดสอบ 480 รูป เป็นกรอบรูปทั้งหมด 790 กรอบรูป

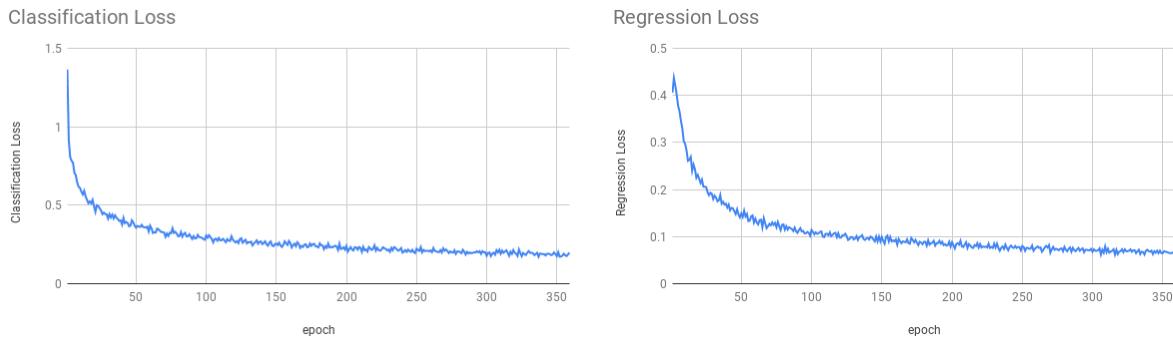
จากการเทรนด์ทั้งหมด 360 epochs ผลปรากฏว่าโมเดลที่มีค่า Total loss น้อยที่สุด (ผลรวมจาก loss ทั้ง 4 รูปแบบ) ได้แก่ โมเดลจาก epoch ที่ 353 โดยมีค่า loss ดังต่อไปนี้

1. RPN Classification Loss = 0.778
2. RPN Regression Loss = 0.072
3. Classification Loss (Fast R-CNN module) = 0.173
4. Regression Loss (Fast R-CNN module) = 0.067
5. Total Loss = 1.089

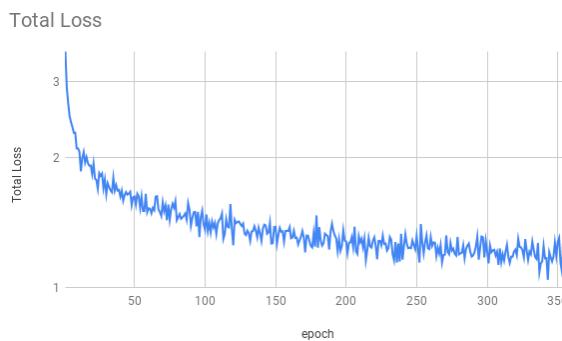
โดยกราฟผลที่ได้จากการเทรนด์ในแต่ละ epoch ดูได้จาก รูปภาพที่ 4.1 – 4.5



รูปที่ 8.1 – กราฟ Classification loss ของ RPN module ในการเทรนด์และ epoch รูปที่ 8.2 – กราฟ Regression loss ของ RPN module ในการトレนด์และ epoch



รูปที่ 8.3 – กราฟ Classification loss ของ Fast R-CNN module รูปที่ 8.4 กราฟ – Regression loss ของ Fast R-CNN module ในในการเทรนด์แต่ละ epoch
การเทรนด์แต่ละ epoch



รูปที่ 8.5 – กราฟ Total loss (รวมทั้ง 4 loss) ในการเทรนด์แต่ละ epoch

ซึ่งทางผู้จัดทำได้ทำการเทรนด์ข้อมูลบน Google Colab ซึ่งใช้ GPU Tesla K80 ในการประมวลผล ซึ่งใช้เวลาในการประมวลผลเฉลี่ย epoch ประมาณ 17 นาที

- การทดสอบและประเมินผล

โดยผลลัพธ์ที่ได้จากการทำข้อมูลมาทดสอบจะใช้ metric ในการวัดก็คือ Average Precision (AP) ซึ่งคำนวณได้จากการ $AveragePrecision = \frac{\sum r^P@r}{R}$, $Precision = \frac{TP}{TP+FP}$ และ $Recall = \frac{TP}{TP+FN}$ ซึ่งผลการทดสอบอยู่มาตั้งนี้

ประเภทขยะ	AP	ประเภทขยะ	AP
ถ่านไฟฉาย	0.79	แก้วพลาสติก	0.95
ขยะอิเล็กทรอนิก	0.89	โฟม	0.94
หลอดไฟ	0.89	ห่อ PVC	0.91
ขวดแดง	0.96	กระดาษขาว	0.90
ขวดขาว	0.86	กล่องนม	0.67
ขวดสเปรย์	0.90	หนังสือพิมพ์	0.54
ขวดพลาสติก	0.99	กล่องลัง	0.94
ขวดน้ำ	0.96	กระป๋อง	0.82
		*mAP	0.878

*mAP ย่อมาจาก Mean Average Precision

ซึ่งทางผู้จัดทำได้ทำการทดสอบข้อมูลบน Google Colab ซึ่งใช้ GPU Tesla K80 ในการประมวลผล ซึ่งใช้เวลาในการประมวลผลเพื่อให้ได้รูปภาพที่ถูกต้องมากับกรอบรูปภาพของขยะและคะแนนความน่าจะเป็นของขยะชิ้นนั้น ใช้เวลาประมาณ 2 วินาที โดยมีตัวอย่างผลลัพธ์ดังนี้



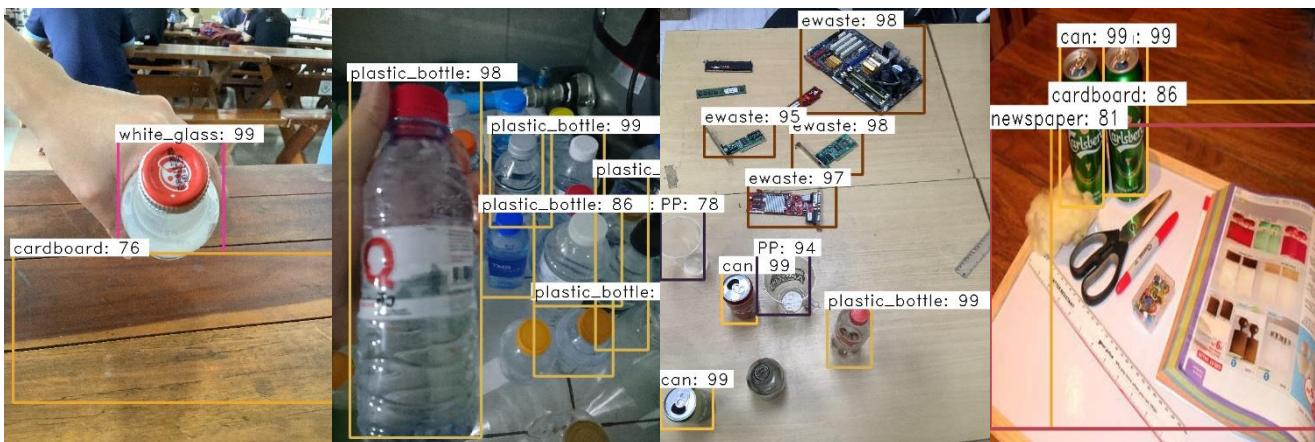
ไม่เดลสามารถทำงานได้ถูกต้องที่อุบัติเหตุที่ไม่คาดเดา ไม่สามารถทำงานได้ในกรณีที่วัตถุหลายชิ้นวางอยู่ในรูปเดียวทันที ไม่มีวัตถุชนิดอื่นวางอยู่ด้วยกัน ไม่สามารถทำงานได้ถูกต้อง ไม่สามารถทำงานได้ถูกต้อง ไม่สามารถทำงานได้ถูกต้อง ไม่สามารถทำงานได้ถูกต้อง



ในรูปภาพที่มีวัตถุจำนวนมากและชนิดที่แตกต่างกันวางอยู่ใกล้กันแต่สามารถมองเห็นตัวของวัตถุได้ชัดเจน ไม่เดลก็สามารถทำงานได้ถูกต้อง

ในกรณีที่วัตถุวางอยู่ใกล้กันมากและวางช้อนกันดังรูป ไม่เดลก็สามารถทราบได้แต่ไม่ช้อนกันได้อย่างถูกต้องโดยในรูปมีเพียงหมุดถูกทั้งหมด 9 ข้างบนที่จริงแล้วเป็นกล่อง 2 กล่องแต่ชิ้นแต่ละไม่เดลพยายามได้เท่ากับที่จริงมีกระป๋อง 8 ชิ้นแต่ทำงานได้ 7 ชิ้น

ในรูปมีกระป๋องน้ำอัดลมที่วางโดยมีตำแหน่งใกล้เคียงกันและช้อนกันไม่เดลก็สามารถทำงานได้เกือบถูกทั้งหมดโดยในรูปที่จริงมีกระป๋อง 8 ชิ้นแต่ทำงานได้ 7 ชิ้น

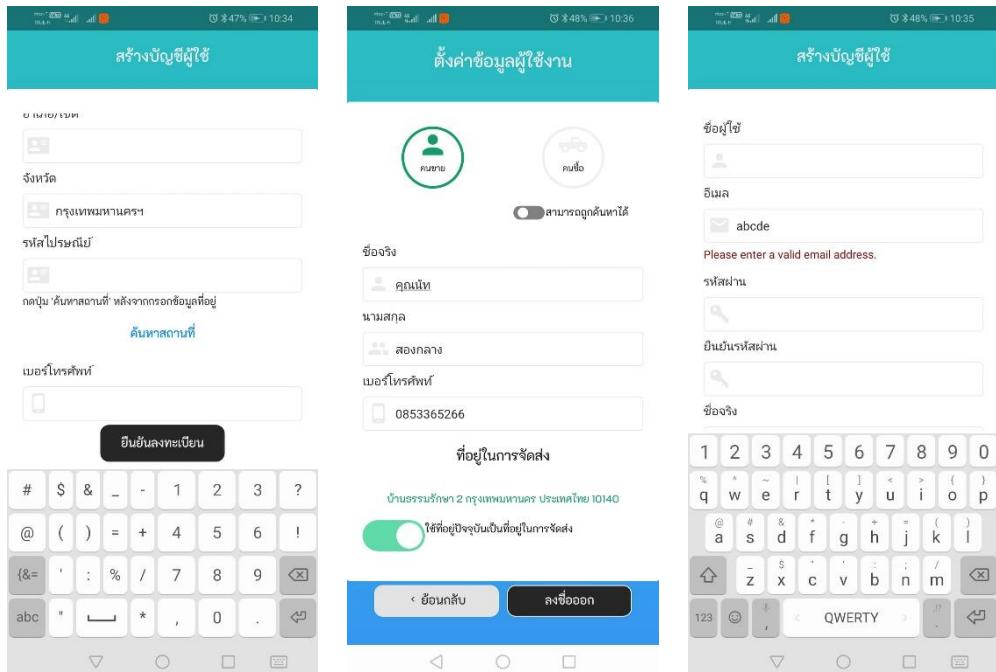


ไม่เดลจะมีปัญหาที่พื้นหลังของวัตถุที่มีรูปบางชนิดที่มีความซับซ้อนมากโดยมีวัตถุจากรูปจะเห็นได้ว่ามีวัตถุบางชนิดที่ไม่เดล เมื่อในรูปภาพมีวัตถุจำนวนมากซึ่งอาจจะลักษณะคล้ายคล้ายคลึงกับกล่องลังทำให้ ที่วางในตำแหน่งที่ข้อนกันจำนวนหลายชิ้น ไม่สามารถทำงานได้ดีกว่าเป็นวัตถุชนิดเดียว เป็นชนิดที่อยู่ใน dataset หรือไม่ออยู่อาจจะไม่เดลทำงานพื้นหลังนั่นว่าเป็นกล่องลัง จะทำให้ไม่เดลทำงานออกแบบได้ไม่ถูกต้อง เนื่องจากมุมของภาพและขนาดของวัตถุใน ทำให้ไม่เดลเกิดการทำงานหายได้ผิดพลาดดังทั้งหมดดังรูป

ภาพที่ให้รายละเอียดด้านอย่างรูป

8.2. การทดสอบแอปพลิเคชัน

■ การยืนยันตัวตนผู้ใช้และข้อมูลผู้ใช้งาน



รูปที่ 8.6 – ขั้นตอนพิมพ์
ตัวเลขเมื่อกรอกเบอร์

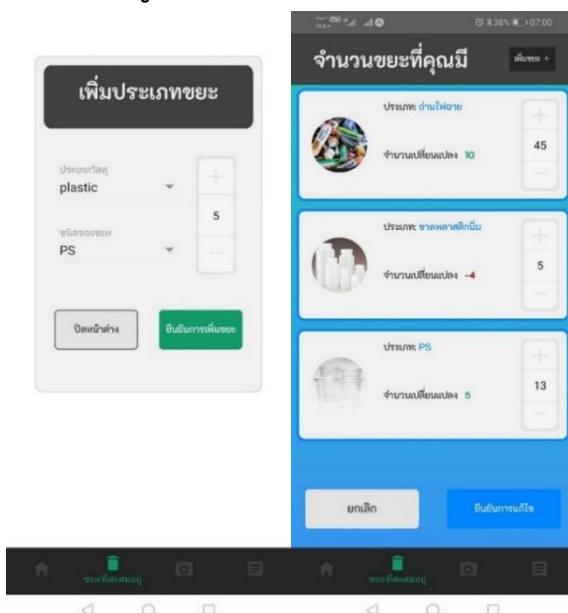
รูปที่ 8.7 – ค่า Default
ข้อมูลในมิติตอน Edit

รูปที่ 8.8 – มีการแจ้งเตือน
เมื่อกรอกผิดเงื่อนไข

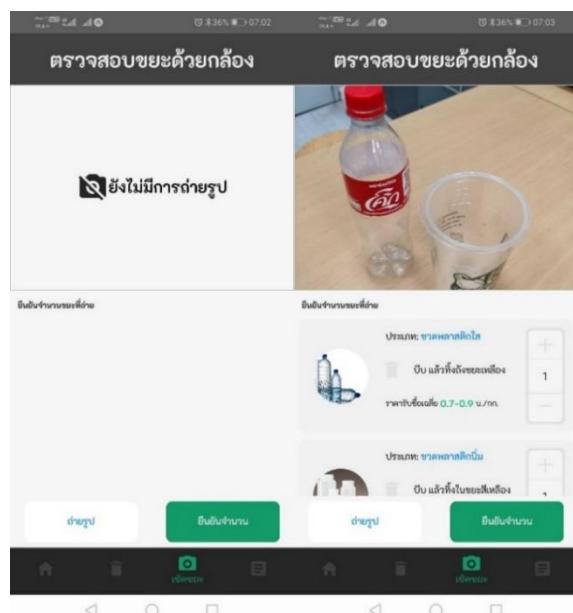
- การสมัครบัญชีผู้ใช้ – การกรอกข้อมูลประเภทตัวเลข แป้นพิมพ์ตัวเลขต้องปรากฏขึ้นแทนแป้นพิมพ์แบบปกติที่พิมพ์ตัวอักษร การค้นหาที่อยู่ของผู้ใช้งานผ่าน Map จะแสดงได้ก็ต่อเมื่อมีการกรอกที่อยู่ไว้แล้วกดปุ่ม ‘ค้นหาสถานที่’ ในส่วนของปุ่ม ‘ยืนยันการสมัครสมาชิก’ จะอนุญาตให้กดได้ เมื่อข้อมูลที่ผู้ใช้งานกรอกทั้งหมดถูกต้องตามเงื่อนไข
- การลงทะเบียนเข้าใช้ – เมื่อกรอกข้อมูลผิดจะต้องมีแจ้งเตือนเกิดขึ้น หากถูกต้องก็จะเข้าสู่หน้า homepage

- การลงชื่อออก – เมื่อกดออก ต้องเด้งกลับมาที่หน้า ‘ลงชื่อเข้าใช้งาน’ และข้อมูลเกี่ยวกับผู้ใช้งานก็จะต้องถูกรีเซ็ตเป็นไม่มีค่า เพื่อไม่ให้การลงชื่อเข้าใช้ในรอบถัดไปเกิดข้อมูลซ้ำ
- การแก้ไขprofile – เมื่อเข้ามาในหน้านี้ ค่าเริ่มต้นจะต้องถูกแสดงบนฟอร์ม และเป็นข้อมูลเดียวกันกับที่ผู้ใช้งานเคยกรอกมาตอนสมัครบัญชีผู้ใช้งาน
- สรุปการทดสอบการยืนยันตัวตน – ไม่มีปัญหารื่องการกรอกข้อมูลแบบปกติ เมื่อผู้ใช้งานกรอกข้อมูลนิดๆ จะมีการแจ้งเตือนเกิดขึ้น แต่ท่าว่าการกรอกข้อมูลประเภทที่ต้องเปิด Map เพื่อค้นหาที่อยู่ตามที่ผู้ใช้กรอก มีปัญหาตรงที่ความแม่นยำไม่มีแม่นยำมากเท่าที่ควร ผู้ใช้งานยังจำเป็นต้องเลื่อนไปยังตำแหน่งที่ต้องการด้วยตนเอง

■ ขยะของผู้ขาย



รูปที่ 8.9 – การทดสอบเพิ่มขยะด้วยหน้าต่างแยก
หน้า



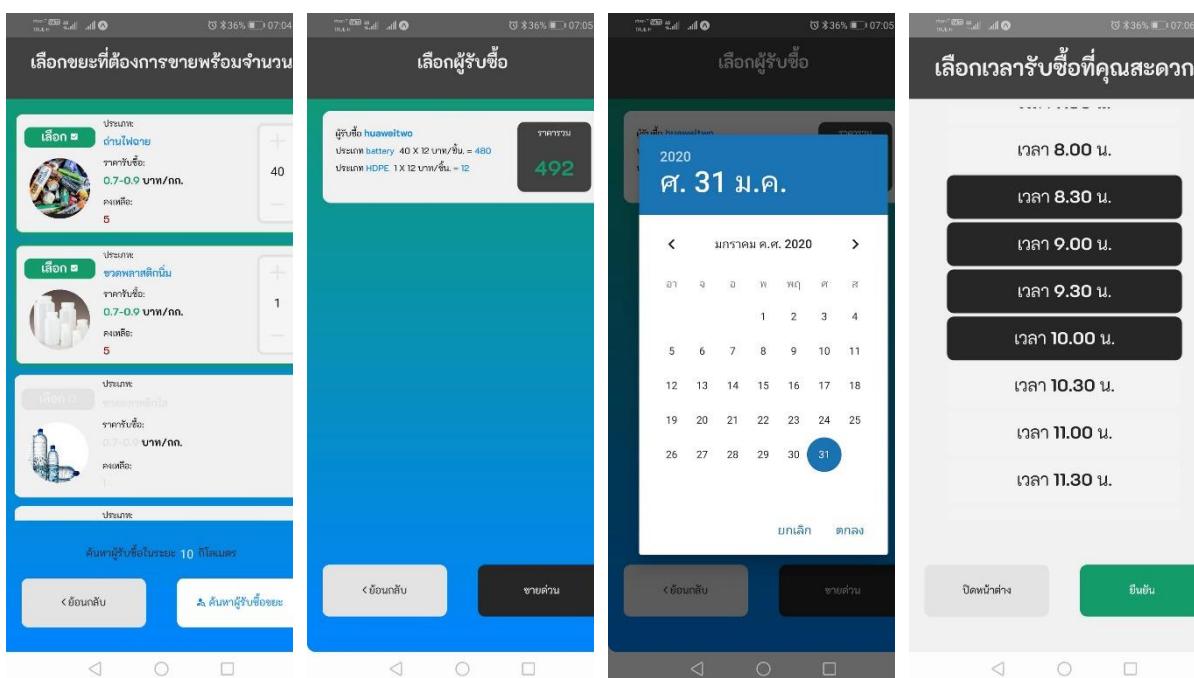
รูปที่ 8.10 – การทดสอบ ตรวจสอบขยะโดยใช้
กล้อง

- การปรับจำนวนขยะที่ผู้ขายขยะมีอยู่ในระบบคลังขยะ – จะสามารถแก้ไขขยะจากการกดเพิ่มและลดได้ก็ต่อเมื่อมีการกดปุ่ม ‘แก้ไขจำนวนขยะ’ เมื่อกดปุ่ม เพิ่ม (เครื่องหมายเพิ่ม) จำนวนขยะประเภทนั้นจะเพิ่มขึ้น 1 การกด ลด (เครื่องหมายลบ) จำนวนขยะประเภทนั้นจะลดลง 1
- การเพิ่มขยะชนิดใหม่พร้อมทั้งระบุจำนวนของผู้ขายขยะ - จะสามารถแก้ไขขยะได้ก็ต่อเมื่อมีการกดปุ่ม ‘แก้ไขจำนวนขยะ’ หลังจากนั้นปุ่ม ‘เพิ่มขยะ’ ก็จะปรากฏ เมื่อกดปุ่มนี้หน้าต่างประเภท Modal จะแสดงให้ผู้ใช้งานกรอกข้อมูลประเภทขยะหลัก ประเภทขยะย่อย และจำนวนเพื่อเพิ่มชนิดขยะเข้าไป ซึ่งค่า Default ในแต่ละฟอร์มประเภทขยะ จะเป็นค่าแรกของแต่ละประเภทหลักและประเภทย่อย ผู้ใช้ไม่สามารถกดเพิ่มขยะได้หากกรอกฟอร์มไม่สมบูรณ์ เมื่อมีการกดปิดหน้าต่าง Modal นี้จะต้องถูกปิดไป เมื่อ

มีการกด ‘ยืนยัน’ หน้าต่างนี้จะถูกปิดไปเช่นกันแต่จะต้องไปเพิ่มจำนวนขยะในหน้าแสดงข้อมูลขยะของผู้ขายตามจำนวนที่กรอกใน Modal ด้วย เพื่อให้ผู้ใช้งานกดยืนยันอีกที

- การพนวก Waste Detection Function ใน Python รวมกับแอปพลิเคชัน - ผู้ใช้งานกดปุ่ม ‘ถ่ายรูป’ ระบบต้องแสดงปริมาณขยะและประเภทขยะได้พร้อมกัน เมื่อกดปุ่ม ‘ยืนยันจำนวนขยะ’ ต้องไปหน้า ‘ขยะของผู้ขายขยะ’ พร้อมกับอยู่ในโหมด ’แก้ไขจำนวนขยะ’ และจำนวนขยะที่ถูกแก้ไขจะต้องถูกแก้เป็นจำนวนขยะที่ได้มาจากการของ Waste Detection Function เพื่อรอให้ผู้ใช้กดยืนยัน หลังจากการปุ่มยืนยันขยะ จำนวนขยะเหล่านั้นที่ได้จากหน้า Waste Detection Function จะต้องถูกเพิ่มเข้าไปในบัญชีผู้ขายขยะ
- สรุปการทดสอบระบบการขายขยะของผู้ใช้งาน – มีปัญหารือเรื่องความชำนาญที่เกิดขึ้นขณะดำเนินการที่เกี่ยวข้องกับการอัปเดตข้อมูลผ่านเดต้าเบส ตัวอย่างเช่น การอัปเดตข้อมูลจำนวนขยะของผู้ใช้งาน ผู้ใช้จำเป็นต้องรอ เพื่อให้ข้อมูลที่ส่งไปยังหน้าต่างๆ ของแอปพลิเคชันมีความถูกต้อง

■ การขายขยะ



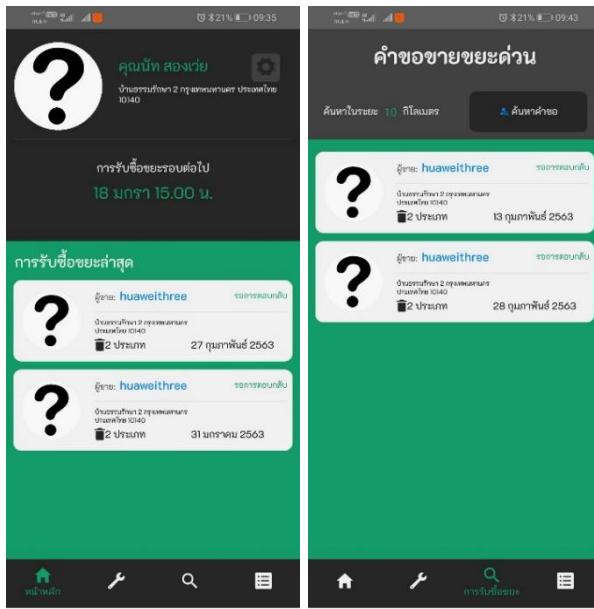
รูปที่ 8.11 – การทดสอบการแสดงผล UI และข้อมูล สามารถเกิดขึ้นได้ตามลำดับที่ออกแบบไว้ แต่มีตัวเลือกบ้างตอนโหลด Icon, การส่งค่าไปยังหน้าต่างๆ ปกติ API วันที่แสดงได้ตามปกติ

- กำหนดจำนวนขยะที่ต้องการจะขาย - เมื่อกดปุ่ม ‘Check’ ขยะที่ต้องการจะขายจะแสดงจำนวนและถูก Highlight โดยการกดครั้งแรกจะเป็นการเลือกจำนวนทั้งหมดที่มีในประเภทนั้น และกดปุ่ม Uncheck ที่แสดงหลังจากปุ่ม Check เพื่อยกเลิกประเภทขยะที่ต้องการจะขาย เมื่อกดที่ตัวเลข แป้นพิมพ์ตัวเลขจะต้องปรากฏ

- เลือกผู้รับซื้อขาย – ขยะที่ผู้รับซื้อแต่ละคนรับซื้อจะต้องถูกแสดงพร้อมกับราคารับซื้อตามประเภทของขยะที่ผู้ขายขยะเลือกขายเท่านั้น เมื่อการส่งคำขอขายขยะสำเร็จ ต้องไปยังหน้า Transaction ทั้งหมด
- กดขายด่วน – กดปุ่ม ‘ขายด่วน’ คำขอขายจะจะต้องถูกสร้างตามประเภทและปริมาณขยะทั้งหมดที่ถูกกำหนดในหน้า ‘เลือกขยะเพื่อขาย’ จากนั้นระบบนำไปสู่หน้าแสดง Transaction ทั้งหมด
- สรุปการทดสอบระบบการขายขยะของผู้ใช้งาน – มีปัญหารื่องความชำนาญในการการขาย ก็คือหลังจากเสร็จสิ้นการส่งคำสั่งไปยังเดต้าเบส ตัวอย่างเช่น การส่งคำสั่งขายขยะ ผู้ใช้จำเป็นต้องรอ เพื่อให้ข้อมูลที่ส่งไปยังหน้าต่างๆ ของแอปพลิเคชันมีความถูกต้อง และดีเลย์ที่เกิดจากการโหลด Icon

■ การรับซื้อขาย

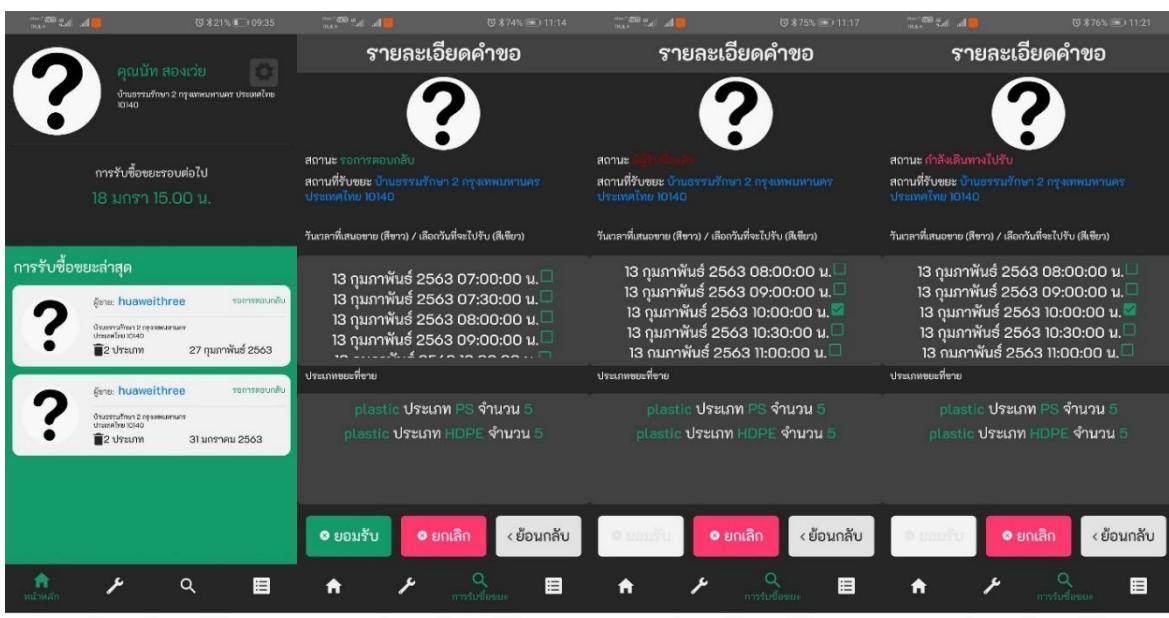
- กดรับซื้อคำขอขายขยะที่ผู้ขายขยะเลือกขายให้ผู้รับซื้อรายนี้ เมื่อกดที่คำขอขายขยะ ข้อมูลที่เกี่ยวข้องกับคำขอนั้นจะต้องแสดง พร้อมกับปุ่ม action ต่างๆ ที่ใช้ในการเปลี่ยนแปลง Status ของคำขอนั้น เช่น ปฏิเสธคำขอขายขยะ, ยอมรับคำขอขายขยะ, เสนอวันเวลาอื่น, กำลังเดินทางไปรับ โดยปุ่มเหล่านี้จะถูกแสดงแตกต่างกันตามสถานะของคำขอปัจจุบัน เมื่อกดยอมรับ สถานะของคำขอนั้นจะถูกเปลี่ยนเป็น ‘มีผู้รับซื้อแล้ว’
- กดรับซื้อคำขอขายขยะที่ผู้ขายขยะกดขายขยะแบบด่วน – ต้องทำได้เช่นเดียวกันกับการกดรับซื้อคำขอแบบแรกที่กล่าวไปข้างต้น เพิ่มเติมคือเมื่อมีการกดปุ่ม ‘ยอมรับ’ สถานะของคำขอนั้นจะถูกเปลี่ยนเป็น ‘มีผู้รับซื้อแล้ว’ พร้อมกับ ต้องมีชื่อของผู้รับซื้อรายนี้เป็น Buyer ในคำขอนั้นด้วย
- สรุปการทดสอบระบบการซื้อขายของผู้ใช้งาน – มีปัญหารื่องความชำนาญในการรับซื้อ เช่นเดียวกันกับการขาย หลังจากเสร็จสิ้นการส่งคำสั่งไปยังเดต้าเบส ตัวอย่างเช่น การรับซื้อ การเปลี่ยนสถานะของคำขอ ผู้ใช้จำเป็นต้องรอ เพื่อให้ข้อมูลที่ส่งไปยังหน้าต่างๆ ของแอปพลิเคชันมีความถูกต้อง



รูปที่ 8.12 – การทดสอบว่าข้อมูลคำร้องขอจากผู้คนขายขยะถึงผู้คนซื้อขยะ ทั้งการร้องขอขายแบบ
ด่วนและการร้องขอขายแบบเลือกผู้รับซื้อ

■ ประวัติการรับซื้อและประวัติการขาย

- ผู้รับซื้อดูประวัติการรับซื้อ-ผู้ขาย ต้องไปหน้าที่แสดง Transactions ทั้งหมด โดยแต่ละ Transaction จะมีข้อมูลและสถานะที่แตกต่างกัน ซึ่งปุ่ม Action จะต้องแตกต่างกันไปตามสถานะของ Transaction นั้นๆ ดังนี้
 - เมื่อ Status เท่ากับ ‘กำลังดำเนินการ’ ปุ่มที่กดได้ ยกเลิก
 - เมื่อ Status เท่ากับ ‘รอการยืนยันวันใหม่’ ปุ่มที่กดได้ ยอมรับเงื่อนไข
 - เมื่อ Status เท่ากับ ‘มีผู้รับซื้อแล้ว’ ปุ่มที่กดได้ ยกเลิก
 - เมื่อ Status เท่ากับ ‘กำลังไปรับ, ยกเลิกแล้ว, ดำเนินการเสร็จสิ้น’ ปุ่มที่กดได้ต้องไม่มี
- ผู้ขายดูประวัติการขาย-ผู้ซื้อ สามารถไปหน้า ‘แสดง Transaction ทั้งหมด’ เพื่อแสดง Transaction ทั้งหมด โดยแต่ละ Transaction จะมีข้อมูลและสถานะที่แตกต่างกัน ซึ่งปุ่ม Action จะต้องแตกต่างกันไปตามสถานะของ Transaction นั้นๆ ดังนี้
 - เมื่อ Status เท่ากับ ‘กำลังดำเนินการ’ ปุ่มที่กดได้ ยอมรับ ยกเลิก เสนอเวลาใหม่
 - เมื่อ Status เท่ากับ ‘รอการยืนยันวันใหม่’ ปุ่มที่กดได้ ยกเลิก
 - เมื่อ Status เท่ากับ ‘รับซื้อแล้ว’ ปุ่มที่กดได้ ยกเลิก กำลังไป
 - เมื่อ Status เท่ากับ ‘กำลังไปรับ, ยกเลิกแล้ว, ดำเนินการเสร็จสิ้น’ ต้องไม่มีปุ่มใดกดได้เลย
- สรุปการทดสอบระบบการขายขยะของผู้ใช้งาน – ในขณะที่กดปุ่มใดๆ เพื่อเปลี่ยนสถานะของ Transaction ผู้ใช้จำเป็นต้องรอ เพื่อให้ข้อมูลที่ส่งไปยังหน้าต่างๆ ของแอปพลิเคชันมีความถูกต้อง

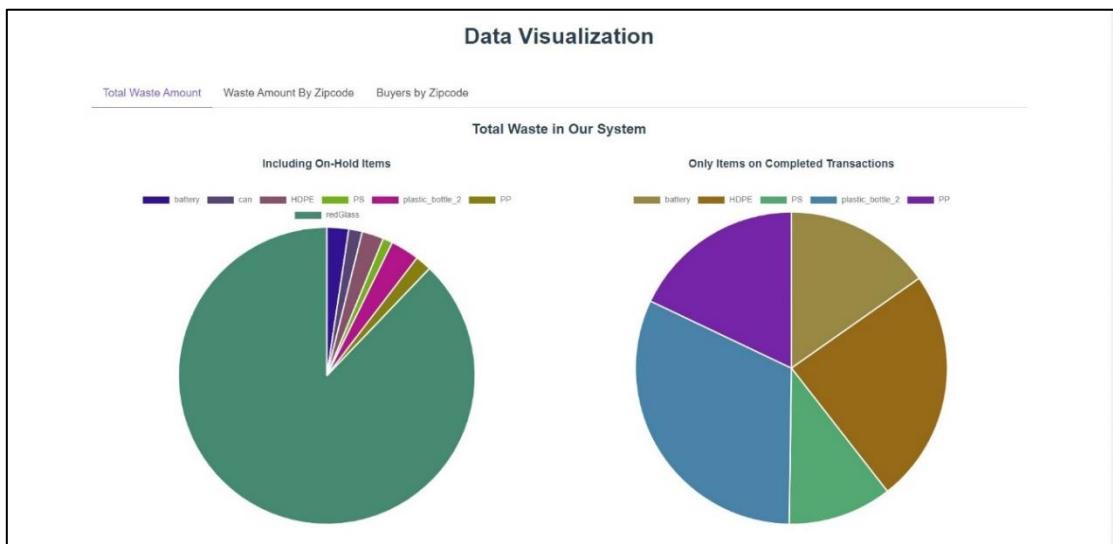


รูปที่ 8.13 – การทดสอบสถานะ Transaction ต่าง Action Button ก็จะต่างกัน

8.3. การทดสอบเว็บไซต์

■ การดึงข้อมูลของทั้งหมดภายในระบบ

- การดึงข้อมูลออกมาแสดงผลเป็น Pie Chart สามารถทำได้อย่างถูกต้องและรวดเร็ว แต่ทั้งนี้ทั้งนั้น กรณีที่ฐานข้อมูลใหญ่ขึ้น การดึงข้อมูลออกมาอาจจะช้าลงไปตามลำดับ

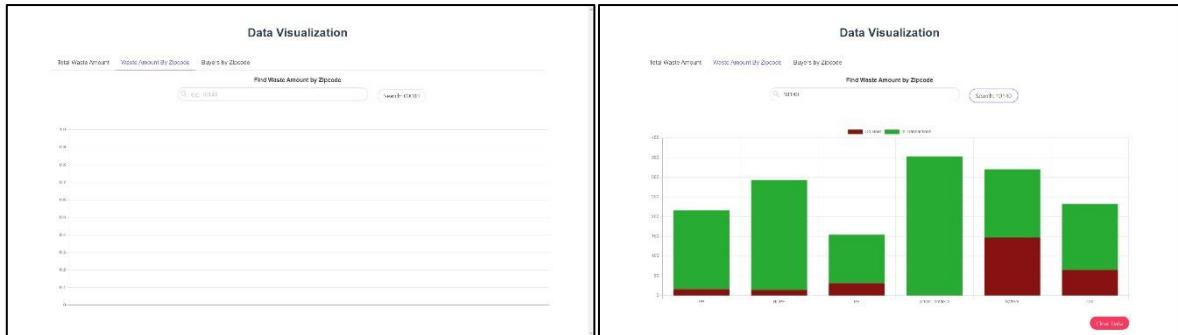


รูปที่ 8.14 – Pie Chart ที่แสดงให้เห็นถึงข้อมูลจำนวนของภัยภายในแอปพลิเคชัน

■ การดึงข้อมูลตามรหัสไปรษณีย์

- การดึงข้อมูลของมาการแสดงผลเป็น Bar Chart ตามรหัสไปรษณีย์ สามารถทำได้อย่างถูกต้องและรวดเร็ว แต่ทั้งนี้ทั้งนั้น กรณีที่ฐานข้อมูลใหญ่ขึ้น การดึงข้อมูลออกมาอาจจะช้าลงไปตามลำดับ

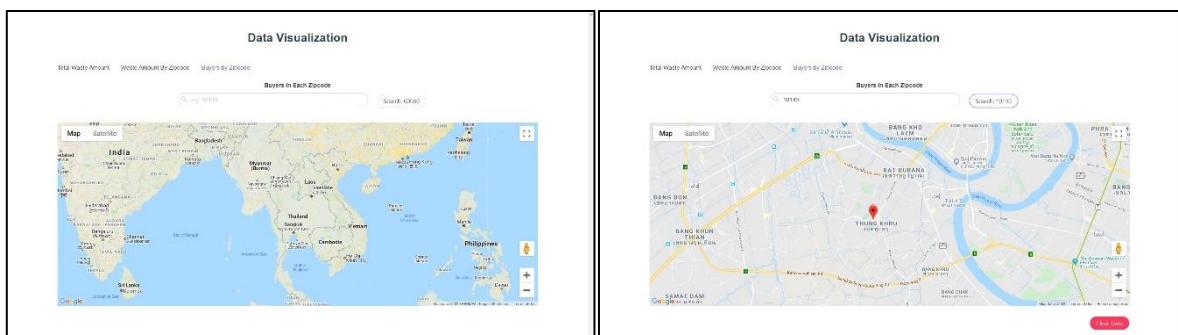
- การเปรียบเทียบข้อมูลของ 2 รหัสไปรษณีย์สามารถทำได้อย่างถูกต้องและรวดเร็ว แต่ทั้งนี้ทั้งนั้น กรณีที่ฐานข้อมูลใหญ่ขึ้น การดึงข้อมูลอกรามากก็อาจจะช้าลงไปตามลำดับ



รูปที่ 8.15 – หน้าให้กรอกรหัสไปรษณีย์เพื่อดึงข้อมูลจำนวนขยะภายในขอบเขตรหัสไปรษณีย์นั้น

■ การดึงที่อยู่ของผู้รับตามรหัสไปรษณีย์

- การดึงที่อยู่ของผู้รับซึ่งจะแสดงเป็น Marker บน Google Map ตามรหัสไปรษณีย์ สามารถทำได้อย่างถูกต้องและรวดเร็ว แต่ทั้งนี้ทั้งนั้น กรณีที่ฐานข้อมูลใหญ่ขึ้น การดึงข้อมูลอกรามากก็อาจจะช้าลงไปตามลำดับ



รูปที่ 8.16 – หน้าให้กรอกรหัสไปรษณีย์เพื่อดึงข้อมูลที่อยู่ของผู้รับซึ่งจะแสดงในเขตรหัสไปรษณีย์นั้น

9. ปัญหาและอุปสรรค

9.1. โมเดล

- ปัญหาในการใช้งาน tensorflow serving ซึ่งเป็นโปรแกรมที่ช่วยเพิ่มประสิทธิภาพสำหรับการสร้างเชิฟเวอร์การทำนายผลด้วยโมเดล เกิดบัคในการ export ข้อมูลในโมเดลที่สร้างด้วย keras เพื่อนำไปใช้งานใน tensorflow serving ทำให้ผลลัพธ์จากการทำนายที่ออกมาจากใช้โปรแกรม tensorflow serving มีค่าผิดเพี้ยนจากปกติ ส่งผลให้ทางผู้จัดใช้การ deploy แบบปกติซึ่งได้ประสิทธิภาพที่ไม่ดีเท่าการใช้โปรแกรม tensorflow serving
- เนื่องจากรูปภาพที่ผู้ใช้งานจะถ่ายและส่งเข้ามาในระบบจะเป็นรูปน่าจะเกิดจากการถ่ายรูปจากกล้องโทรศัพท์ทั้งหมด ผู้จัดทำจึงต้องทำการสร้าง dataset ที่ส่วนมากเป็นรูปที่ถ่ายจากกล้องโทรศัพท์จริง
- เนื่องจากตอนแรกมีรูปที่นำเข้ามาใช้เทรนด์ในโมเดลเป็นรูปจากโทรศัพท์มือถือที่ไม่ได้ทำการ resize เป็นจำนวนมากทำให้เกิดปัญหาการเคลื่อนย้ายไฟล์ และปัญหาการเทรนด์ข้อมูลที่ซ้ำเนื่องจากโปรแกรมต้องอ่านไฟล์รูปที่มีขนาดใหญ่ ผู้จัดทำจึงทำการ resize รูปโดยใช้อัลกอริทึมเดียวกับที่ใช้ในโมเดล ทำให้รูปมีขนาดเล็กลงและสามารถนำเข้าไปเทรนด์ได้เลย
- ผลลัพธ์จากการทำนายวัตถุชนิดที่เป็นของใส คือ แก้วขาว ขวดพลาสติก และ แก้วพลาสติก ไม่ถูกต้องบ่อยเนื่องจากมีลักษณะพื้นผิวของวัตถุที่ใกล้เคียงกันในกรณีที่รูปภาพถูกถ่ายในมุมที่ไม่ตรงกับ dataset
- ผลลัพธ์ที่ได้จากการทำนายวัตถุที่พื้นหลังของรูปนั้นเป็นสีขาวล้วน บางครั้งโมเดลจะทำนายว่าพื้นหลังนั้นเป็นกระดาษขาว
- กรณีที่มุมของรูปที่ส่งเข้าเป็นมุมที่เป็นมุมแปลงๆ หรือไม่ตรงกับ dataset(ซึ่งผู้จัดทำพยายามจะถ่ายรูปวัตถุให้ครบทุกมุม) ผลลัพธ์ที่ทำนายออกมาได้จะไม่ถูกต้อง
- การทำนายรูปที่มีความซับซ้อน เช่น รูปที่มีวัตถุหลายชิ้น หรือรูปที่มีวัตถุวางซ้อนกัน ผลลัพธ์ที่ออกมาอาจจะไม่ถูกต้อง
- การทำนายรูปภาพที่ไม่ถูกต้องเนื่องจากรูปวัตถุชนิดนั้นไม่ได้อยู่ใน dataset แต่มีลักษณะคล้ายกับรูปใน dataset

9.2. แอปพลิเคชัน

- การทดสอบแอปพลิเคชันทำได้ช้า – การทดสอบแอปพลิเคชัน ไม่ว่าจะเป็นเรื่องหน้าตา สี หรือการใช้งานฟังก์ชันต่างๆ ภายในแอปพลิเคชัน ต้องอัปโหลดไฟล์ JavaScript Bundle ขึ้น Expo Environment ก่อนทำให้เวลาที่ใช้มากขึ้น
- Third Party Library จัดการเรื่องหน้าตายาก และอาจไม่เข้ากันกับ Component ภายในบางตัว - ทำให้บาง Component ต้องเขียนเอง ซึ่งใช้เวลานานเข่นกัน
- Specific Detail ของอุปกรณ์โทรศัพท์หรือแท็ปเลต ไม่เหมือนกัน – บาง Component สามารถใช้ได้แค่บางรุ่นของอุปกรณ์ แต่ระบบปฏิบัติการหรือรุ่นอุปกรณ์ต่างกัน พฤติกรรมของ Component นั้นก็จะ

แสดงแตกต่างกัน ทำให้เป็นเรื่องยากที่จะสามารถทำให้อแอปพลิเคชันนั้นสามารถใช้งานได้บนอุปกรณ์ทุกชนิดที่มีอยู่ในท้องตลาด

- บาง library ไม่สามารถใช้ได้ จากการที่พัฒนาแอปพลิเคชันผ่าน Expo – เพราะว่าการพัฒนาโดยไม่ใช้ Expo จำเป็นจะต้อง Config ตัว Emulator ด้วยตัวเอง ทั้ง Android และ IOS ซึ่งเป็นเรื่องที่ยาก อีกทั้งคอมพิวเตอร์ที่ใช้ในการพัฒนาไม่สามารถใช้งาน Emulator ของ Apple ได้ ทำให้การทดสอบแอปพลิเคชัน จะทำได้แค่เพียงแค่ Android เท่านั้น
- การเก็บที่อยู่ของผู้ใช้งานต้องแบ่งเป็น 3 อย่าง ที่อยู่ที่ เป็นข้อความ พิกัดละติจูด - ลองจิจูด และรหัสไปรษณีย์
 - การเก็บที่อยู่ที่ เป็นข้อความนั้นเพื่อเอาไว้ใช้วิเคราะห์ภายในแอปพลิเคชัน
 - เนื่องจากการเรียกใช้หลายๆ ฟังก์ชันอย่างเช่น path optimization ต้องใช้จุดพิกัด ทำให้ต้องเก็บส่วนตรงนี้เพิ่มเข้ามา
 - เนื่องจากการใช้ที่อยู่ตาม 9ops ของมือถือได้ที่อยู่กลับมา ในหลายครั้งไม่มี เขต/อำเภอ แขวง/ตำบล ทำให้ต้องใช้รหัสไปรษณีย์ในการแยกเขตต่างๆ แทน
- จากอุปสรรคข้างต้น ทำให้เกิดปัญหาที่ ไม่มีเวลาในการ Deploy หรือนำแอปพลิเคชันขึ้นสู่ตลาดซอฟแวร์ เนื่องจากการ Deploy อาจจะก่อให้เกิดปัญหาความเข้ากันไม่ได้ของ Library ต่างๆ เช่น การใช้ Google Map ขณะนี้สามารถเรียกใช้ได้ง่ายผ่าน Expo แต่อาจจะมีปัญหาในตอนที่ Build ตัวแอปพลิเคชันเป็น Standalone App ที่ไม่มี Expo ค่อยจัดการเรื่องการ Config อุปกรณ์ให้ อีกทั้งฟังก์ชัน Notification ที่ใช้ได้ต่อเมื่อผู้ใช้งานสมัคร Expo Account เท่านั้น

9.3. เว็บไซต์

- ข้อมูลที่นำออกมาระบบแสดงผลบนเว็บไซต์ยังเป็นข้อมูลที่เกิดจากการทดสอบแอปพลิเคชัน ทำให้ข้อมูลที่แสดงผลออกมามิ่งสามารถนำมาใช้เคราะห์หอย่างจริงจังได้

10. แนวทางในการพัฒนาและประยุกต์ใช้งานร่วมกับงานอื่นๆ ในขั้นตอนไป

- เพิ่มความสามารถในการทำนายของโมเดลให้มีความคลอบคลุมชนิดของขยะให้มากขึ้น
- เพิ่มความแม่นยำในการทำนายของโมเดลให้ถูกต้องมากขึ้น
- พัฒนา API ให้ภาครัฐและเอกชนนำข้อมูลในระบบไปใช้งานได้
- พัฒนาแอปพลิเคชันให้สามารถใช้งานได้ง่ายและน่าใช้มากขึ้น
- นำตัวแอปพลิเคชันลง Google play store และ App store เพื่อให้ผู้ใช้งานสามารถโหลดมาใช้งานได้โดยไม่ต้องผ่านโปรแกรม Expo

11. ข้อสรุปและข้อเสนอแนะ

ผู้จัดทำได้ทำการพัฒนาโครงงานนี้ขึ้นมา เพื่อให้ความรู้ในการจัดการขยะอย่างมีประสิทธิภาพ และเสริมสร้างความสอดคล้องสบายนในการซื้อขายขยะ ซึ่งโครงงานนี้สามารถแบ่งออกเป็นได้ 3 ส่วน ได้แก่ ส่วนของแอปพลิเคชัน เว็บไซต์ และโมเดล ซึ่งแต่ละส่วนนั้นก็จะมีข้อบกพร่องที่แตกต่างกันออกไป ไม่เหมือนกัน

ในส่วนของการพัฒนาและทดสอบแอปพลิเคชันนั้นเป็นส่วนที่ดำเนินไปได้ราบรื่น การทดลองตามขั้นตอนต่างๆ ในตัวของแอปพลิเคชันนั้นไม่พบปัญหาใดๆ แต่ทั้งนั้นทั้งนั้น อาจจะมีข้อบกพร่องภายในแอปพลิเคชันนนั้น ยกเว้นจากที่ได้ทำการทดสอบเกิดขึ้นได้ นอกจากนี้ ในตัวแอปพลิเคชันยังไม่ถูกอัพโหลดแอปพลิเคชันขึ้น Playstore และ Appstore เนื่องจากอาจมีปัญหาที่ตามมาหลังจากการอัพโหลดขึ้นไป ทำให้ขั้นตอนนี้ถูกชะลออยู่มาก่อน อย่างไรก็ตาม การอัพโหลดแอปพลิเคชันขึ้น Playstore และ Appstore จะถูกจัดการในภายหลัง

หลังจากที่ผู้จัดทำได้ทำการเทรนโมเดลโดยใช้สถาปัตยกรรม Faster-RCNN เป็นจำนวน 360 epochs และปรากฏว่าผลลัพธ์ที่เมื่อวัดเป็นค่า mAP แล้วอยู่ที่ 0.878 ซึ่งเมื่อเทียบกับผลลัพธ์จากในเปเปอร์ Faster R-CNN แล้ว ค่า mAP ของเรามากกว่า แต่ค่าที่สูงกว่านั้นเกิดจากการที่โมเดลของคณะผู้จัดทำเกิด Overfitting ซึ่งส่งผลให้รูปที่ใช้ทดสอบบางรูป ไม่ถูกต้องเท่าที่ควรอย่างที่กล่าวไว้ในผลลัพธ์การทดลอง กล่าวคือหากรูปที่นำมาทดสอบเป็นรูปที่ไม่เคยเห็นมาก่อน โมเดลจะสามารถแยกประเภทขยะนั้นได้อย่างแม่นยำ แต่ในทางกลับกัน ถ้าหากรูปที่ใช้ทดสอบเป็นรูปที่มีวัตถุซ้อนกันหลายๆ ประเภทความแม่นยำของโมเดลก็จะลดน้อยลงไป

เนื่องจากข้อมูลในส่วนของเว็บไซต์ที่แสดงผลนั้น ขึ้นอยู่กับข้อมูลจริงภายในฐานข้อมูลของแอปพลิเคชัน ทำให้ข้อมูลที่แสดงผลออกมาต่อนี้ยังเป็นเพียงข้อมูลที่เกิดขึ้นจากการทดลองในการใช้แอปพลิเคชันเท่านั้น ส่งผลให้กราฟ หรือข้อมูลต่างๆ ที่แสดงออกมาบนเว็บไซต์ไม่สามารถนำมาระหว่างหัวใจความสัมพันธ์ หรือนำไปต่อยอดต่อไปได้ แต่ทั้งนี้ทั้งนั้น หากตัวของแอปพลิเคชันของเรายังสามารถใช้งานขึ้นจริงแล้ว ข้อมูลที่แสดงอยู่บนเว็บไซต์นั้นจะสามารถนำไปใช้เคราะห์ ทำความสัมพันธ์ต่างๆ ได้ อย่างเช่นการวิเคราะห์หาพื้นที่ที่จะก่อสร้างโรงเรือน จำกัดจำนวนข้อมูลการซื้อขายภายในแอปพลิเคชัน เป็นต้น

จากที่กล่าวไปทั้งหมด ถึงแม้ว่าส่วนต่างๆ ภายในโครงงานยังมีข้อบกพร่องบางประการ แต่ก็ข้อบกพร่องต่างๆ เหล่านี้ยังสามารถแก้ไขและพัฒนาต่อในอนาคตได้ นอกจากนี้ โครงงานนี้ก็สามารถตอบโจทย์ได้ตาม

ข้อสรุปและข้อเสนอแนะ

วัตถุประสงค์ที่ตั้งขึ้นมาไว้ในตอนแรกซึ่งก็คือการให้ความรู้ในการจัดการขยายอย่างประสิทธิภาพ โดยการระบุ ประเภทและวิธีการกำจัดขยายภายในแอปพลิเคชัน และเสริมสร้างความสัมภានด้วยการซื้อขายขยาย ด้วย พงกชั้นการซื้อขายภายในแอปพลิเคชัน ดังนั้นผู้จัดทำจึงเชื่อว่าโครงงานนี้จะก่อให้เกิดการเปลี่ยนแปลงเกี่ยวกับ การจัดการขยายในประเทศไทยในอนาคต ไม่มากก็น้อย

12. เอกสารอ้างอิง

- [1] กรมควบคุมมลพิษ กระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อม (2562). **สรุปสถานการณ์มลพิษของประเทศไทยปี 2561 [PDF File]**. กรมควบคุมมลพิษกระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อม. สืบค้นจาก <http://www.pcd.go.th/file/Thailand%20Pollution%20Report%202018_Thai.pdf>
- [2] กรมควบคุมมลพิษ กระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อม. **ปัญหาสิ่งแวดล้อมจากขยะมูลฝอย.** กรมควบคุมมลพิษ. สืบค้นเมื่อ 15 กันยายน 2562. <http://www.pcd.go.th/info_serv/waste_rubbish.htm>
- [3] Piyaporn Wongruang (2018). **SPECIAL REPORT: Alarm raised as Thailand drowns in plastic trash.** The Nation Thailand. สืบค้นเมื่อวันที่ 15 กันยายน 2562. <<https://www.nationthailand.com/national/30344702>>
- [4] S Vassanadumrongdee, S Kittipongvises (2018). **Factors influencing source separation intention and willingness to pay for improving waste management in Bangkok, Thailand.** Sustainable Environment Research, vol. 28, no. 2, pp. 90-99
- [5] AW Ando, AY Gosselin (2005). **Recycling in multifamily dwellings: does convenience matter?** Economic inquiry 43 (2), pp. 426-438
- [6] SF Sidiq, F Lopi, SV Joshi (2010). **The effects of behavior and attitudes on drop-off recycling activities.** Resources, Conservation and Recycling, vol. 54, no. 3, pp. 163-170
- [7] Anna Bernstad (2014). **Household food waste separation behavior and the importance of convenience.** Waste Management, vol. 34, no. 7, pp. 1317-1323
- [8] TTP Nguyen, D Zhu, NP Le (2015). **Factors influencing waste separation intention of residential households in a developing country: Evidence from Hanoi, Vietnam.** Habitat International, vol. 48, pp. 169-176
- [9] Dana H. Ballard; Christopher M. Brown (1982). **Computer Vision.** Prentice Hall. ISBN 978-0-13-165316-0.
- [10] Huang, T. (1996-11-19). **Vandoni, Carlo, E (ed.). Computer Vision : Evolution And Promise.** 19th CERN School of Computing. Geneva: CERN. pp. 21–25. doi:10.5170/CERN-1996-008.21. ISBN 978-9290830955.
- [11] Milan Sonka; Vaclav Hlavac; Roger Boyle (2008). **Image Processing, Analysis, and Machine Vision.** Thomson. ISBN 978-0-495-08252-1.
- [12] Hubel DH, Wiesel TN (1968). **Receptive fields and functional architecture of monkey striate cortex.** J Physiol 195:215–243

- [13] Fukushima K (1980). **Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.** Biol Cybern 36:193–202
- [14] Statcounter (2019). **Mobile Operating System Market Share Worldwide.** Statcounter. สืบค้นเมื่อ 15 กันยายน 2562. <<https://gs.statcounter.com/os-market-share/mobile/worldwide>>
- [15] John A. Calderaio (2017). **Comparing the Performance between Native iOS (Swift) and React-Native.** Medium. สืบค้นเมื่อ 27 กันยายน 2562. <<https://medium.com/the-react-native-log/comparing-the-performance-between-native-ios-swift-and-react-native-7b5490d363e2>>
- [16] Fei-Fei Li, Justin Johnson, & Serena Yeung (2017). **Lecture 9: CNN Architectures [PowerPoint slides].** Stanford. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf>
- [17] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola (2019). **Dive into Deep Learning.** D2L. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <<https://d2l.ai/d2l-en.pdf>>
- [18] Arthur Douillard (2018). **Selective Search for Object Recognition.** Arthurdouillard. สืบค้น เมื่อ 25 พฤษภาคม 2562. จาก <<https://arthurdouillard.com/post/selective-search/>>
- [19] Schuyler Smith (2014). **Selective Search for Object Recognition [PowerPoint slides].** Stanford. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <http://vision.stanford.edu/teaching/cs231b_spring1415/slides/sssearch_schuyler.pdf>
- [20] He, K., Zhang, X., Ren, S., & Sun, J. (2016). **Deep residual learning for image recognition.** In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [21] Pulkit Sharma (2018). **A Step-by-Step Introduction to the Basic Object Detection Algorithms.** Analyticsvidhya. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <<https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>>
- [22] Shilpa Ananth (2019). **Faster R-CNN for object detection.** TowardDataScience. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <<https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46>>
- [23] Shilpa Ananth (2019). **Fast R-CNN for object detection.** TowardDataScience. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <<https://towardsdatascience.com/fast-r-cnn-for-object-detection-a-technical-summary-a0ff94faa022>>

- [24] Fei-Fei Li, Justin Johnson, & Serena Yeung (2017). **Lecture 11: Detection and Segmentation [PowerPoint slides]**. Stanford. สืบค้นเมื่อ 25 พฤษภาคม 2562. จาก <http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf>
- [25] Sik-Ho Tsang (2019). **Review: YOLOv3 — You Only Look Once (Object Detection)**. TowardDataScience. สืบค้นเมื่อ 30 มกราคม 2562. จาก <<https://towardsdatascience.com/review-yolov3-you-only-look-once-object-detection-eab75d7a1ba6>>
- [26] Vaidehi Joshi (2017). **Finding The Shortest Path, With A Little Help From Dijkstra**. Medium. สืบค้นเมื่อ 30 มกราคม . จาก <<https://medium.com/basecs/finding-the-shortest-path-with-a-little-help-from-dijkstra-613149fbdc8e>>

13. สถานที่ติดต่อของผู้พัฒนาและอาจารย์ที่ปรึกษา โทรศัพท์ มือถือ โทรสาร E-mail

ผู้พัฒนา

1. ธนาสนธิ เอี่ยมยิ่งสกุล 087-681-9990 titlethanason@gmail.com
บ้านเลขที่ 99/119 ถ.ปทุมสามโคก ต.บางปรอ ก อ.เมืองปทุมธานี จ.ปทุมธานี 12000
2. นราวดิษฐ์ ทับทิมโต 096-306-1332 narawit.tub@outlook.co.th
บ้านที่เลขที่ 226/17 ถ.เทศา ต.พระปฐมเจดีย์ อ.เมืองนครปฐม จ.นครปฐม 73000
3. วัชรินทร์ ศิริเนาวกุล 099-286-7533 wchr.aun@gmail.com
บ้านเลขที่ 12 ถ.สวนผัก ช.35 แขวงฉิมพลี เขตตลิ่งชัน กรุงเทพมหานคร 10170

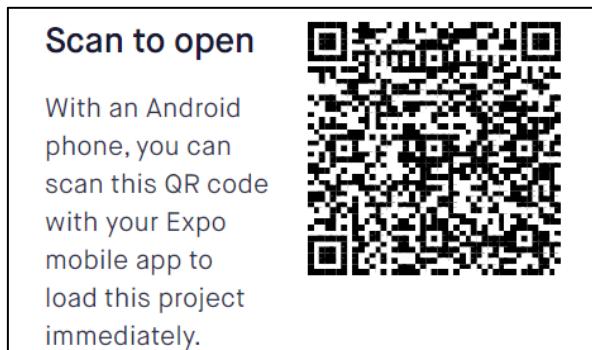
อาจารย์ที่ปรึกษา

อัญชลิสา แต้ตระกูล 086-505-6429 unchalisatae@mail.kmutt.ac.th
เลขที่ 126 ถ.ประชาอุทิศ แขวงบางมด เขตทุ่งครุ กรุงเทพมหานคร 10140

14. ภาคผนวก

คู่มือการติดตั้งแอปพลิเคชัน

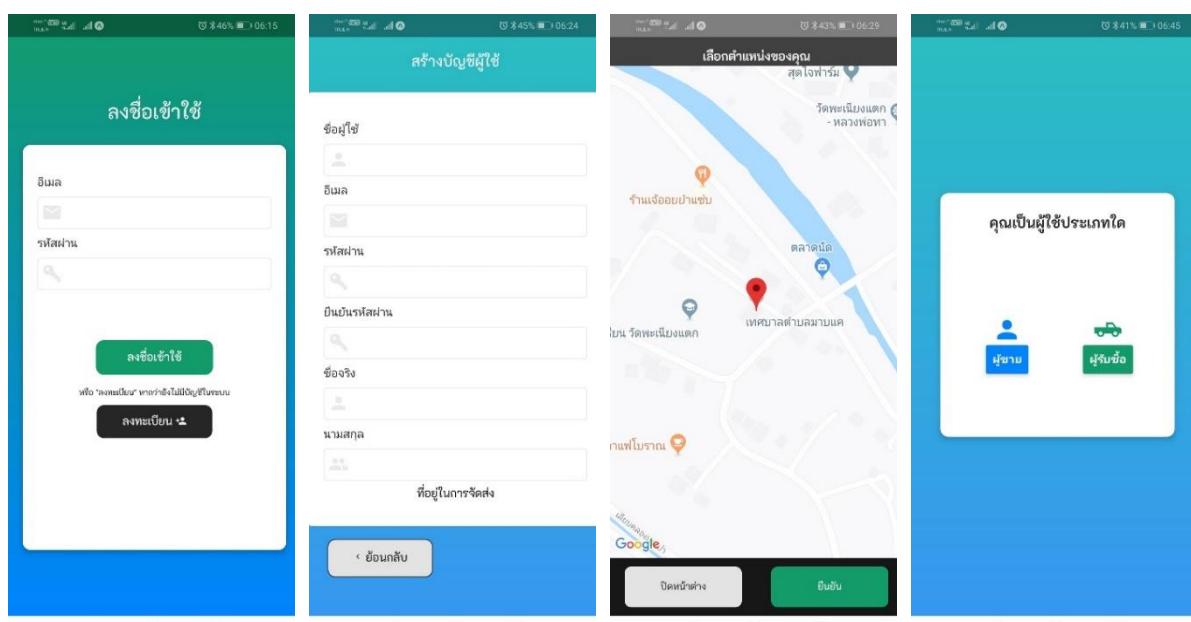
- ดาวน์โหลดและติดตั้งแอปพลิเคชัน Expo ลงบนมือถือ สามารถติดตั้งได้ทั้ง Android และ IOS
<https://play.google.com/store/apps/details?id=host.exp.exponent&hl=en>
- จากนั้นเปิดแอปพลิเคชัน Expo แล้วสแกน QR Code ต่อไปนี้



คู่มือการใช้งาน

1. แอปพลิเคชัน

■ การสมัครสมาชิก



รูปที่ 14.1

รูปที่ 14.2

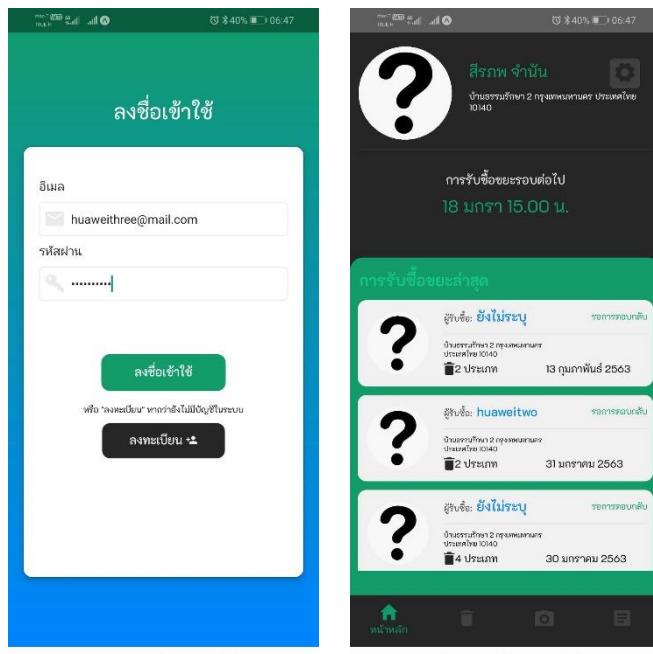
รูปที่ 14.3

รูปที่ 14.4

- กดปุ่ม 'ลงทะเบียน' ดังรูปที่ 14.1
- ดึงหน้าต่างฟอร์มในรูปที่ 14.2 กรอกข้อมูลให้ครบถ้วน
- การกรอกข้อมูลที่อยู่นั้นสามารถกรอกได้สองรูปแบบ ก็คือ
 - ใช้ตำแหน่งที่อยู่ปัจจุบันของมือถือเป็นข้อมูลที่อยู่ โดยการกด 'ใช่ที่อยู่ปัจจุบัน'

- กรอกข้อมูลลงฟอร์มที่อยู่ จากนั้นกด ‘ค้นหาสถานที่’ เพื่อเปิด Map และค้นหาตำแหน่งที่ต้องการใช้เป็นที่อยู่ ดังรูปที่ 14.3 และกดยืนยัน เพื่อบันทึกตำแหน่งที่เลือกเป็นที่อยู่
4. หลังจากเสร็จสิ้นการกรอกข้อมูลทั้งหมด กดปุ่ม ‘ยืนยันการลงทะเบียน’
5. เลือกประเภทผู้ใช้งาน มีสองแบบก็คือผู้ขายขยะกับผู้รับซื้อขยะ ดังรูปที่ 14.4

■ การลงทะเบียนเข้าใช้

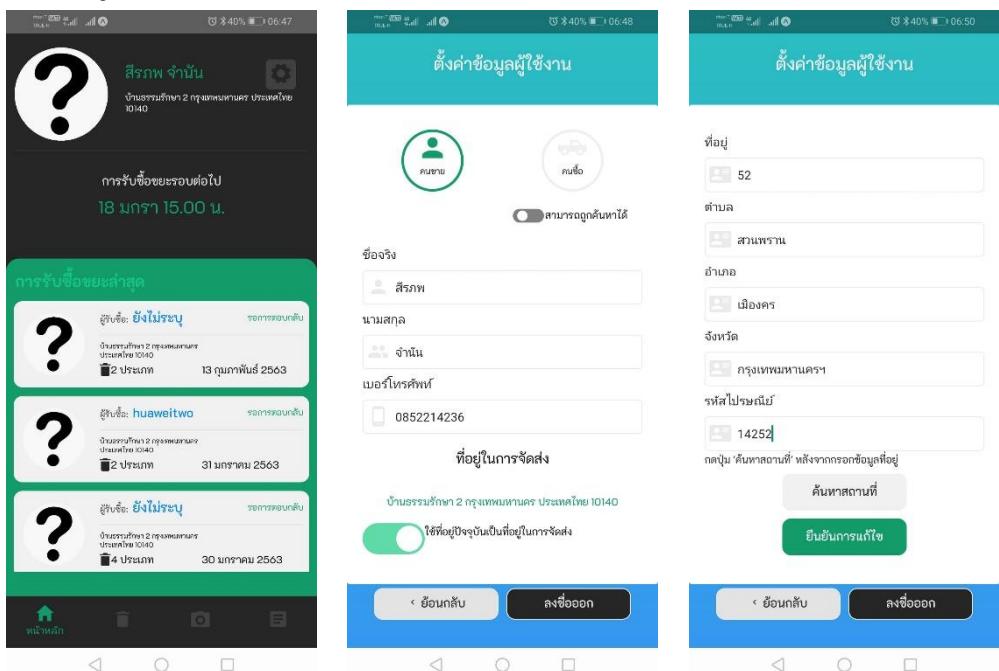


รูปที่ 14.5

รูปที่ 14.6

1. ดังรูปที่ 14.5 กรอกอีเมลและพาสเวิร์ดลงในฟอร์ม จากนั้นกด ‘ลงชื่อเข้าใช้’
2. เมื่อเสร็จสิ้น ระบบจะนำพาไปหน้าหลัก โดยเป็นหน้าหลักตามสถานะ เช่น หากเป็นผู้ขายขยะ หน้าหลักก็จะเป็นเมนูของผู้ขายขยะ ถ้าเป็นผู้รับซื้อขยะ เมนูจะเป็นแบบผู้ซื้อขยะ ดังรูปที่ 14.6 เป็นเมนูแบบผู้ขาย

■ การแก้ไขข้อมูล หรือ ลงทะเบียนออกจากระบบ



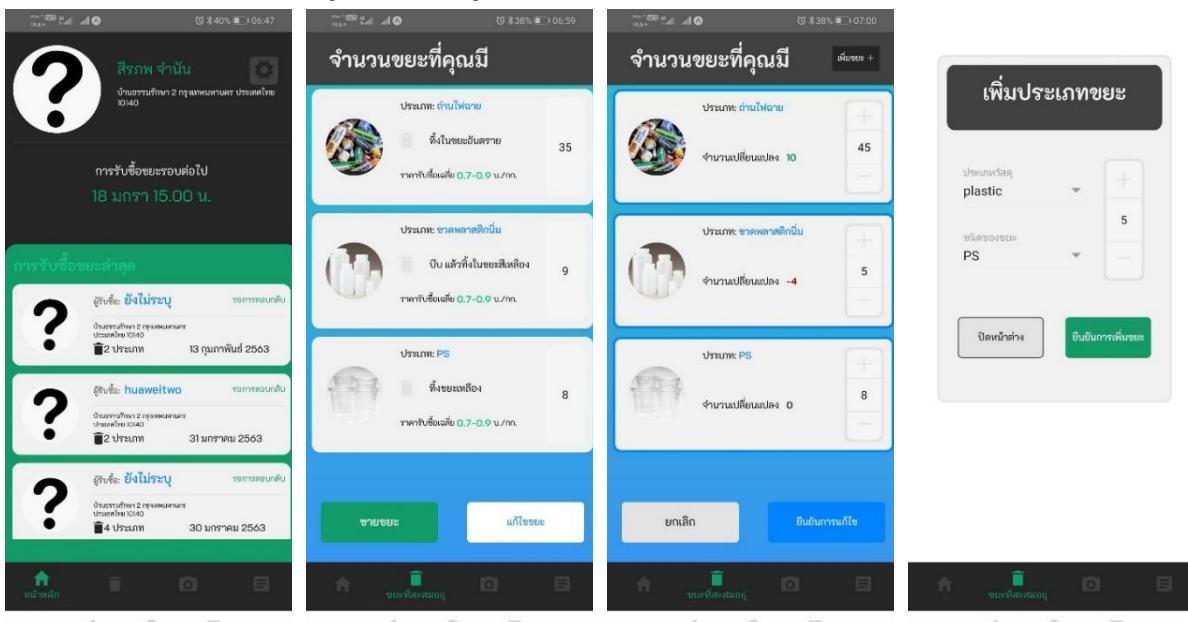
รูปที่ 14.7

รูปที่ 14.8

รูปที่ 14.9

1. ในหน้าเมนูหลัก กดปุ่มฟันเฟืองมุมบนขวาสุด ดังรูปที่ 14.7
2. กรอกข้อมูลที่จะแก้ไขลงในฟอร์ม ในหน้านี้สามารถเปลี่ยนบทบาทได้ว่าจะเป็นผู้ขายหรือผู้รับซื้อขาย ดังรูปที่ 14.8
3. เมื่อกรอกข้อมูลเสร็จสิ้น กดปุ่ม ‘ยืนยันการแก้ไข’ สีเขียว ดังรูปที่ 14.9

■ การปรับจำนวนขยะที่มีอยู่ในคลังของผู้ขาย



รูปที่ 14.10

รูปที่ 14.11

รูปที่ 14.12

รูปที่ 14.13

1. จากหน้าหลัก ให้กดปุ่มที่สองที่เป็นรูปถังขยะ ตรงแถบด้านล่าง จะปรากฏหน้าปรับจำนวนขยะ ดังรูปที่ 14.11
2. กดปุ่ม ‘แก้ไขขยะ’ เพื่อเปลี่ยนเป็นโหมดปรับจำนวนขยะ ดังรูปที่ 14.12 โดยสามารถปรับจำนวนได้โดยการกดปุ่ม บวกและลบ จากขยะแต่ละชนิด
3. เมื่อยื่นใบโหนด แก้ไขขยะ ตรงมุมบนขวาจะมีปุ่ม ‘เพิ่มขยะ’ ซึ่งเมื่อกดเข้าไป หน้าต่าง ‘เพิ่มประเภทขยะใหม่’ จะปรากฏขึ้น ดังรูปที่ 14.13 โดยต้องกรอกข้อมูลให้ครบ ทั้งประเภทวัสดุ ประเภทขยะ และจำนวน จากนั้นกดปุ่ม ‘ยืนยันการเพิ่มขยะ’ จำนวนขยะเหล่านั้นก็จะถูกเพิ่มเข้ามาให้กดปุ่ม ‘ยืนยันการแก้ไข’ อีกครั้ง เพื่อบันทึกข้อมูล

■ การตรวจสอบประเภทขยะด้วยกล้องและจัดเก็บเข้าคลังของผู้ขาย



รูปที่ 14.14



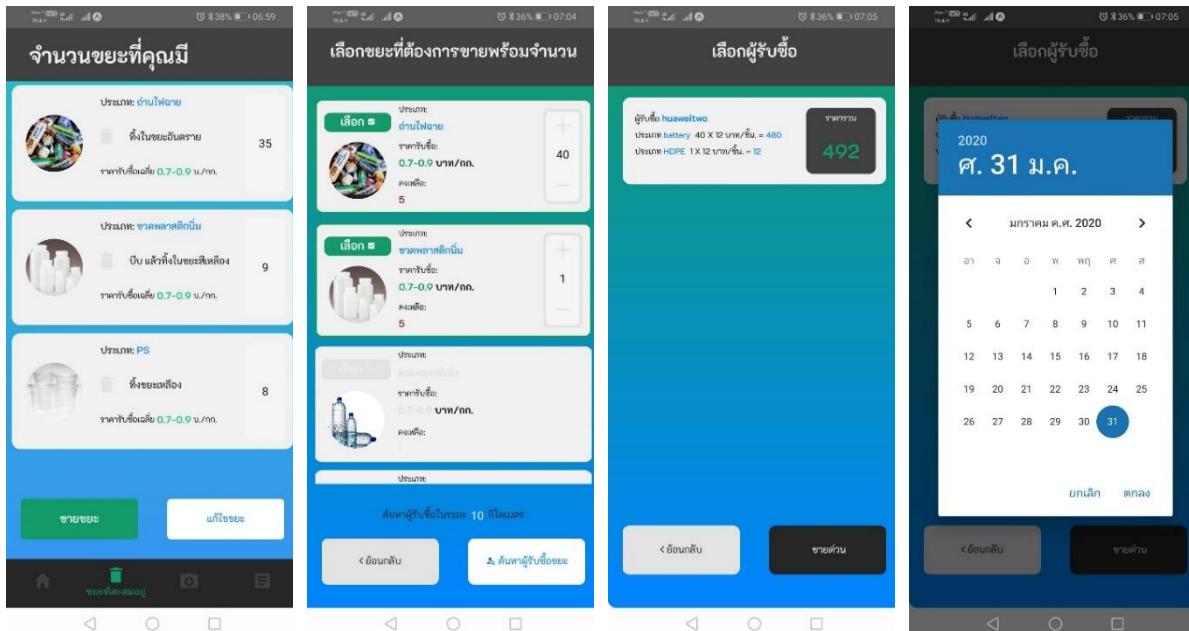
รูปที่ 14.14



รูปที่ 14.14

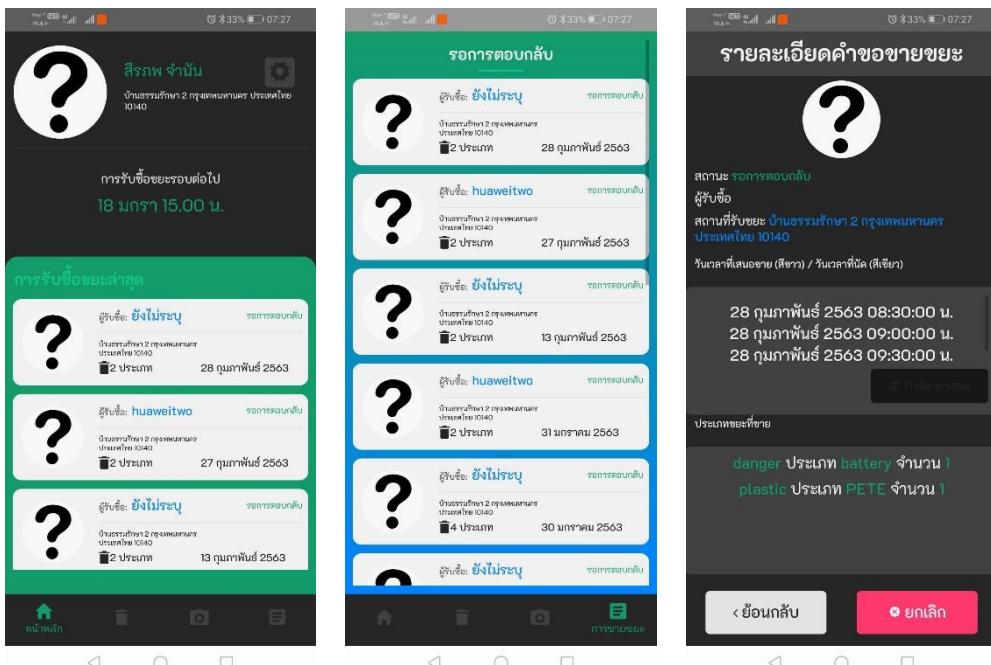
1. ในหน้าหลัก ดังรูปที่ 14.14 กดปุ่มรูปกล้องตรงแถบด้านล่าง
2. กดปุ่ม ‘ถ่ายรูป’ จากนั้นจะมีเครื่องหมาย Check มุมบนขวา กดเครื่องหมาย Check เพื่อดำเนินการต่อ หลังจากนั้นก็จะแสดงผลรูปถ่ายและผลลัพธ์ของการตรวจสอบขยะพร้อมกับจำนวน ดังรูปที่ 14.16
3. กดปุ่มเครื่องหมายบวกลบ เพื่อปรับจำนวนให้ตรงจริง หลังจากนั้นกดปุ่ม ‘ยืนยันจำนวน’

■ การกดคำสั่งขายของผู้ขาย



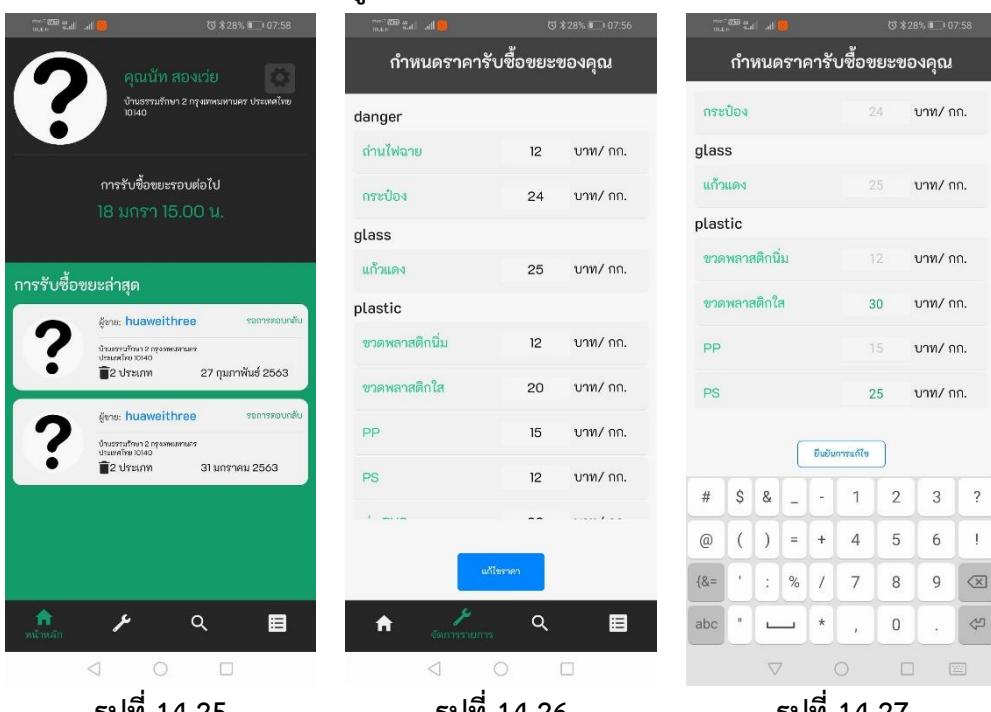
- กดปุ่มถังขยะตรงแถบล่าง ดังรูปที่ 14.17 จากนั้นกดปุ่ม ‘ขายของ’ เพื่อมาขยับหน้าเลือกของที่ต้องการขาย ดังรูปที่ 14.18
- กดปุ่ม ‘เลือก’ ตรงประเภทที่ต้องการจะขาย พร้อมระบุจำนวน ผ่านการกดปุ่มบวกลบ หลังจากนั้นกดปุ่ม ‘ค้นหาผู้รับซื้อขยะ’
- ในหน้านี้จะสามารถทำได้สองอย่าง ก็คือการขายแบบเร่งด่วน กับขายแบบเลือกผู้รับซื้อ ดังรูปที่ 14.19 เมื่อเลือกอย่างใดอย่างนึง ก็จะมีปฏิทินประภูมิให้กดเลือกวัน ดังรูปที่ 14.20
- หลังจากเลือกวัน ให้เลือกเวลาที่สะดวกในการส่งขายขยะให้กับผู้รับซื้อ ขยะ หลังจากนั้นกดปุ่ม ‘ยืนยัน’

■ การตรวจสอบคำร้องขอขายขยะของผู้ขาย



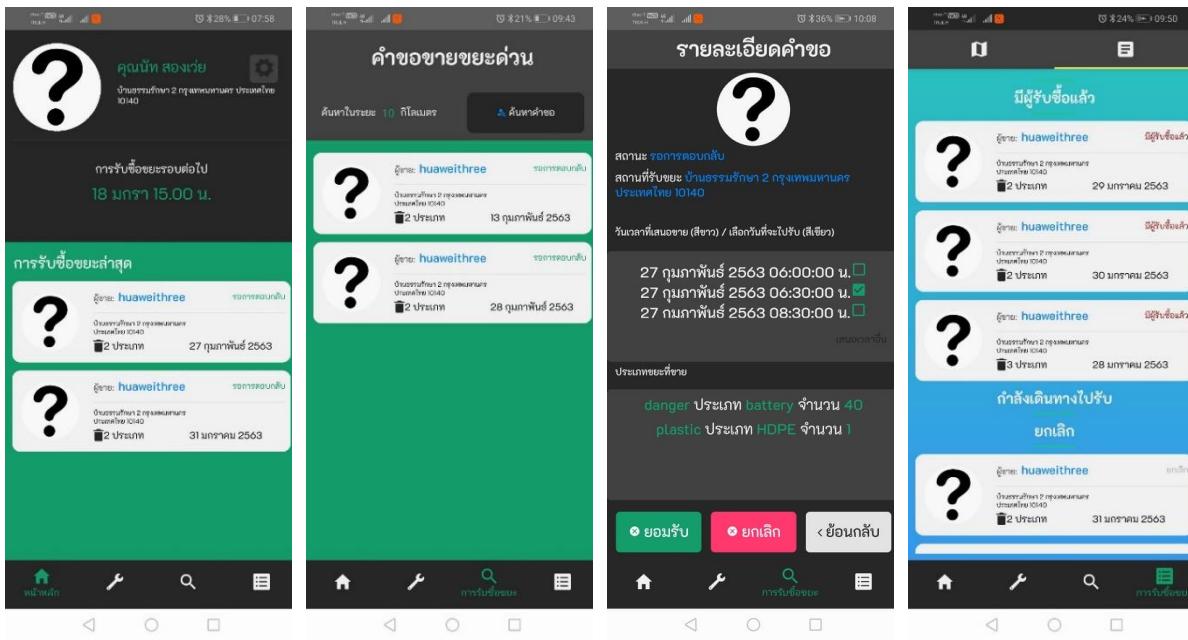
1. จากหน้าหลัก สามารถกด Transaction ‘การรับซื้อขยะล่าสุด’ เพื่อเข้าไปดูรายละเอียดของ Transaction นั้นได้ทันที ดังรูปที่ 14.24 โดยสามารถกดปุ่ม ‘ยกเลิก’ เพื่อยกเลิกคำขอขายขยะได้
2. กดปุ่มที่ 4 จากแถบด้านล่าง เพื่อไปยัง ‘การขายขยะ’ ซึ่งจะเป็นการดูข้อมูล Transaction ทุกประเภท ดังรูปที่ 14.23

■ การปรับเปลี่ยนราคารับซื้อขยะของผู้รับซื้อ



- นาทีหน้าหลักของผู้รับซื้อ กดปุ่มประจำ ปุ่มที่สองจากแถบด้านล่าง เพื่อไปยังหน้าแก้ไขข้อมูลราคา รับซื้อขาย ดังรูปที่ 14.26
- กดปุ่ม ‘แก้ไขราคา’ จากนั้นติ๊กตรงช่องตัวเลขเพื่อทำการแก้ไข
- หลังจากเสร็จสิ้นการแก้ไข กดปุ่ม ‘ยืนยันการแก้ไข’

■ การกรดรับคำร้องขายขยะแบบด่วน ในระยะที่สั้นใจ



รูปที่ 14.29

รูปที่ 14.30

รูปที่ 14.31

รูปที่ 14.32

- จากหน้าหลักดังรูปที่ 14.28 กดปุ่มรูปแวนขยาย หรือรูปที่สามจากแถบด้านล่าง เพื่อไปยังหน้า ‘คำขอขายขยะด่วน’
- ในหน้านี้ สามารถกำหนดระยะเวลาในการค้นหาคำขอขายด่วนได้ ดังรูปที่ 14.29 จะค้นหาคำขอใน 10 กิโลเมตรดังรูปที่ 14.30
- เมื่อกดเข้าไป ให้เลือกเวลาที่ต้องการจะไปรับซื้อ โดยการกดที่เวลา แล้วซอง Check สีเขียว ก็จะแสดงผลเป็น Check เพื่อให้ทราบว่าท่านได้เลือกเวลาแล้ว
- หลังจากนั้นกดปุ่มยอมรับ เพื่อยืนยันคำร้องขอว่าจะไปรับซื้อ
- สามารถตรวจสอบได้ในหน้า การรับซื้อขาย โดยดู Transaction ที่มีสถานะมีผู้รับซื้อแล้ว ดังรูปที่ 14.31

■ การกรดรับคำร้องขอขายของที่ผู้ขายขยะเลือกขายให้โดยตรง



รูปที่ 14.33



รูปที่ 14.34



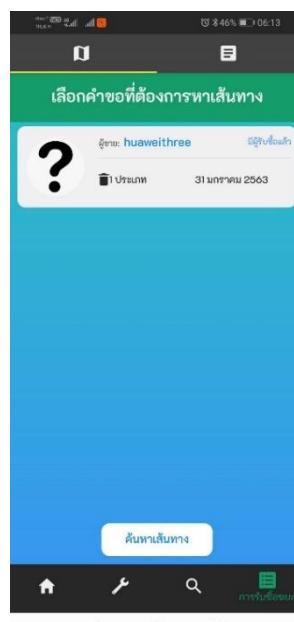
รูปที่ 14.35

1. จากหน้าหลัก สามารถกดที่ Transaction ที่มีสถานะ ‘รอการตอบกลับ’ ได้ทันที เพราะเป็นคำขอขายขยะที่ติดต่อโดยตรงมาอย่างผู้รับซื้อขยะรายนี้ ดังรูปที่ 14.32
2. จากนั้นเลือกเวลาที่ต้องการได้รับข้อ โดยการ Check ที่เวลา จนกว่าจะขึ้นเป็นสีเขียวตรงเวลาที่ต้องการ ดังรูปที่ 14.33 ได้มีการเลือกเวลา 13 กุมภาพันธ์ เวลา เจ็ดโมงครึ่ง
3. เมื่อเสร็จสิ้น ข้อมูลสามารถตรวจสอบได้ในหน้า ‘การรับซื้อขยะ’ ดังรูปที่ 14.34

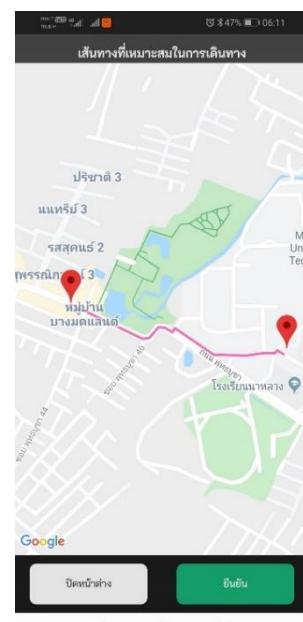
■ การค้นหาเส้นทางที่เหมาะสมที่สุดในการเดินทางไปรับขยะตามสถานที่ต่างๆ



รูปที่ 14.36



รูปที่ 14.37



รูปที่ 14.38

1. จากหน้าหลัก กดແຄບສຸດທ້າຍດ້ານລ່າງ
2. ດັ່ງຮູບທີ່ 14.36 ເລືອກຄໍາຂອທິ່ຕ້ອງການນຳໄປໃໝ່ໃນກາຣຄັນຫາເສັ້ນທາງ ພັລັງຈາກນັ້ນກົດປຸ່ມ ‘ຄັນຫາເສັ້ນທາງ’
3. ໜ້າຕ່າງ Map ຈະຖືກ pop-up ຂຶ້ນມາແສດງເສັ້ນທາງທີ່ແນະນຳໂດຍແອປພລິເຄັນ ດັ່ງຮູບທີ່ 14.37

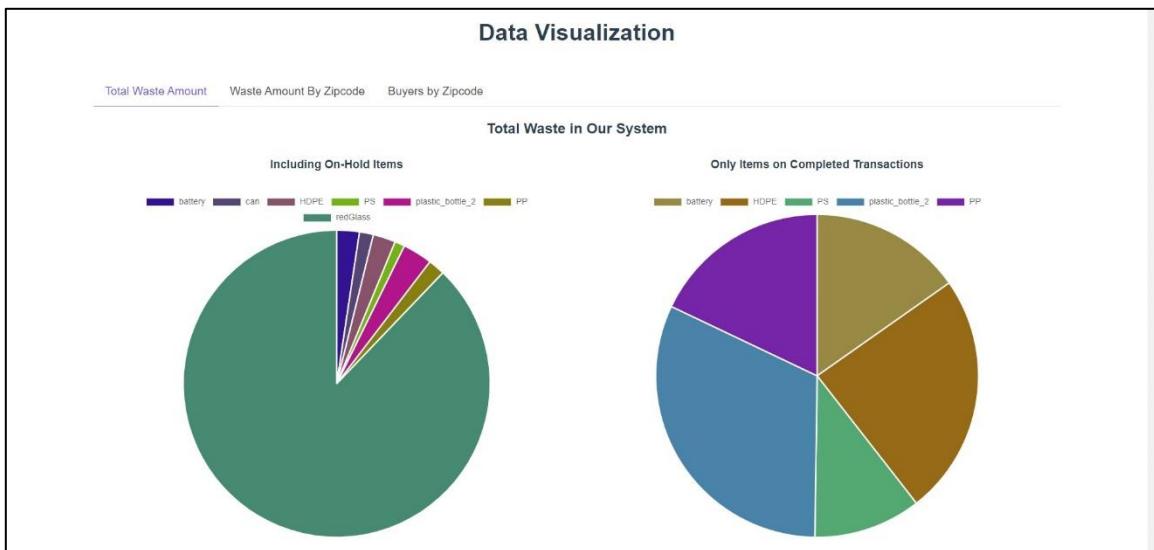
2. ເວັບໄຊຕໍ່

■ ການເຂົ້າໃໝ່ເວັບໄຊຕໍ່

1. ເຂົ້າເວັບໄຊຕໍ່ <https://senior-project-83de1.firebaseioapp.com/>

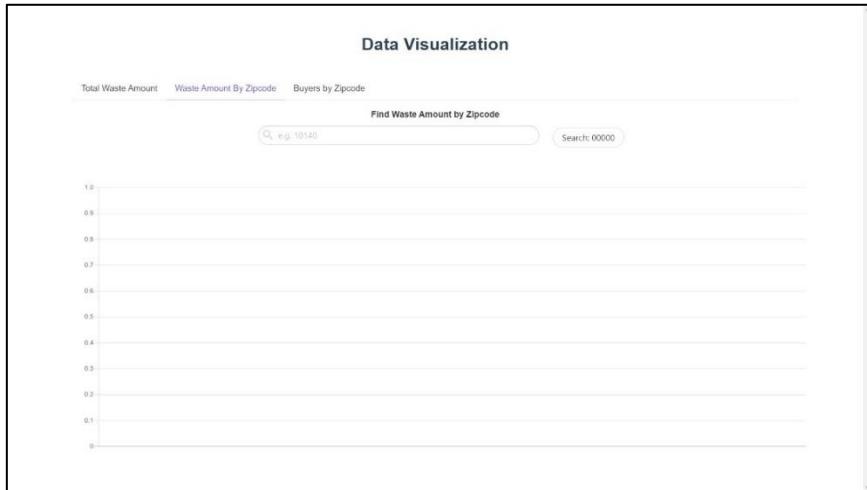
■ ກາຣດູ້ຂໍ້ມູນຈຳນວນຂະໜາດກາຍໃນແອປພລິເຄັນ

1. ເມື່ອເຂົ້າມາແລ້ວຈະເຈອກກັບໜ້າ Total Waste Amount ປຶ້ງກີ່ຄື້ອໜ້າທີ່ບໍອກຄື່ງຂໍ້ມູນຂະໜາດກາຍໃນແອປພລິເຄັນຕອນນີ້
2. Pie Chart ທາງໜ້າແສດງຄື່ງຂໍ້ມູນຂະໜາດກາຍໃນແອປພລິເຄັນ ທັ້ງຂໍ້ມູນຂະໜາດທີ່ເກີດຈາກກາຣ໌ຂໍ້ອໜ້າ ແລະ ຂໍ້ມູນຂະໜາດທີ່ຜູ້ໃຊ້ຈານຄື່ອງຮອງອຸ່ງ
3. ສ່ວນ Pie Chart ທາງໜ້າແສດງໃຫ້ເຫັນເພາະຈຳນວນຂະໜາດທີ່ເກີດຂຶ້ນຈາກກາຣ໌ຂໍ້ອໜ້າທີ່ສໍາເລັດແລ້ວ ກາຍໃນແອປພລິເຄັນ

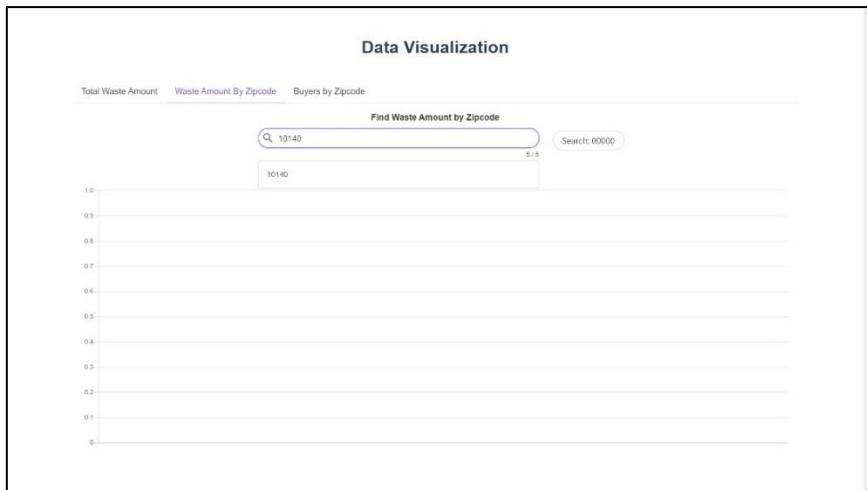


■ การคูข้อมูลจำนวนของขยะตามรหัสไปรษณีย์ภายในเว็บไซต์

1. กดเลือกที่ Waste Amount By Zipcode



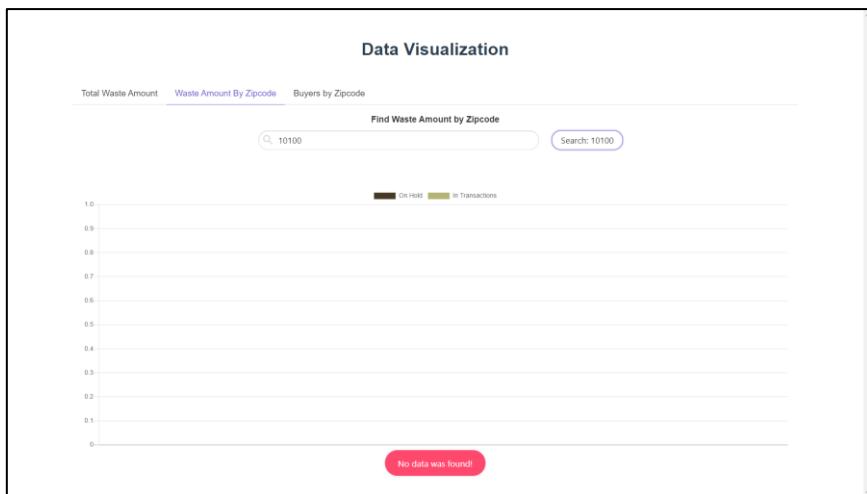
2. ทำการพิมพ์รหัสไปรษณีย์ที่ต้องการจะดูข้อมูลลงใน



a. หากภายในรหัสไปรษณีย์นั้นๆ มีข้อมูล ข้อมูลเหล่านั้นก็จะถูกนำมาแสดงเป็น Bar Chart ตามรูป

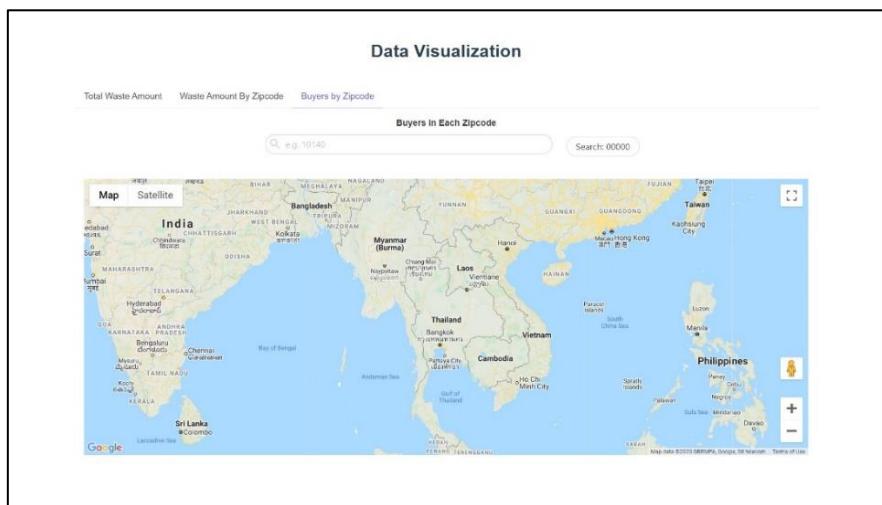


b. หากภายในรหัสไปรษณีย์นั้นไม่มีข้อมูล ก็จะขึ้นข้อความสีแดงข้างล่างว่า No data was found!



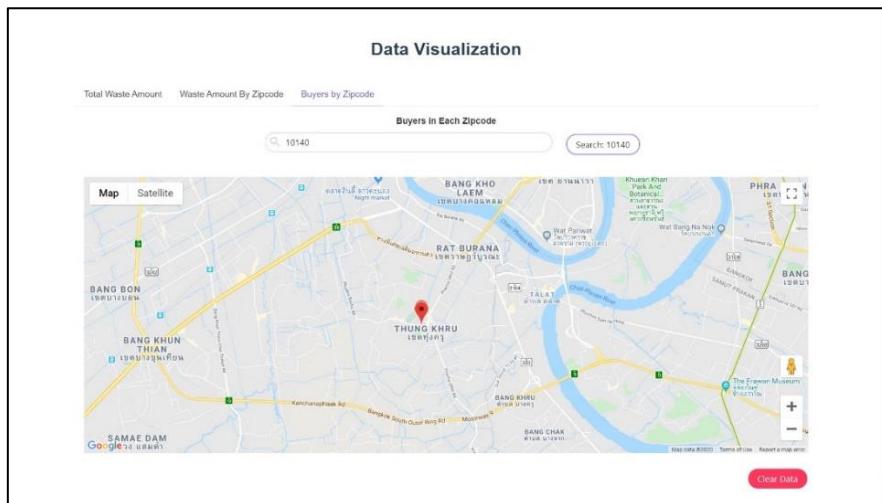
■ การเรียกขอดูข้อมูลที่อยู่ผู้รับซื้อขยะภายในแอปพลิเคชัน

1. กดเลือกที่ Buyers By Zipcode

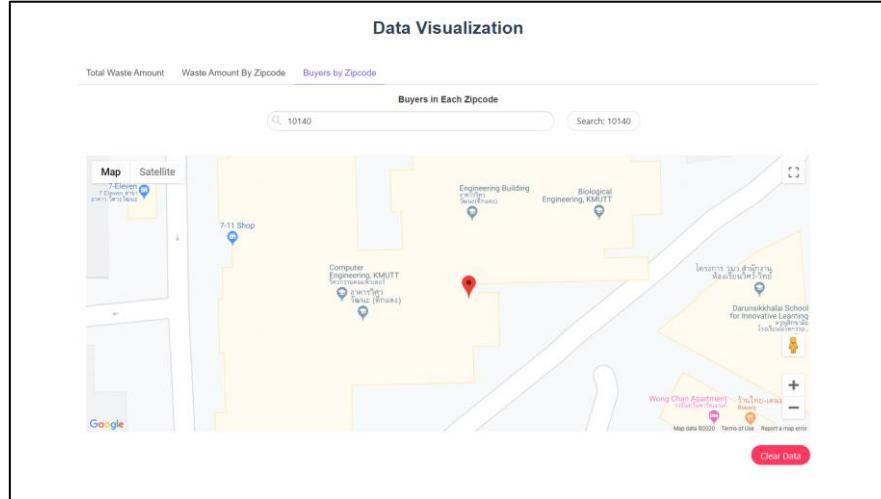


2. ทำการพิมพ์รหัสไปรษณีย์ที่ต้องการจะค้นหาผู้รับซื้อขยะลงไป

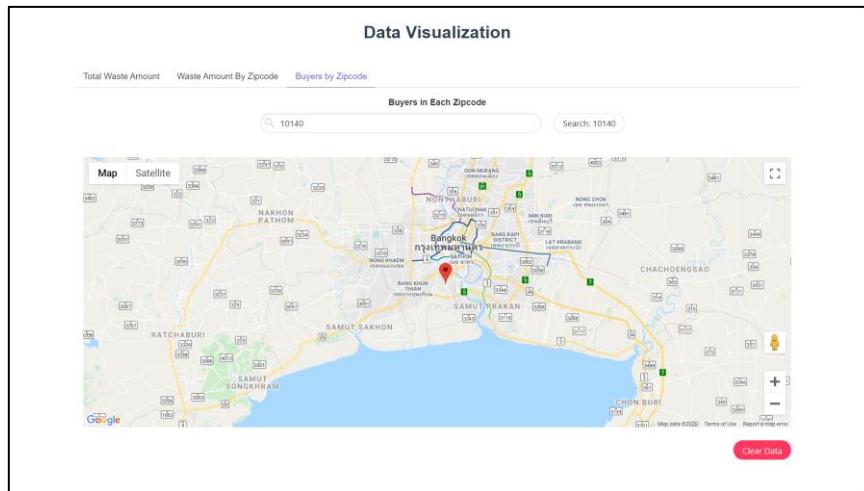
2.1. หากพบรหัสไปรษณีย์ของผู้รับซื้อขยะที่ลงทะเบียนภายในแอปพลิเคชันไว้ ก็จะแสดงเป็นหมุดสีแดงบนแผนที่ดังรูปที่



2.1.1. โดยเราสามารถซูมเข้าไปเพื่อคุ้ที่อยู่ที่ชัดเจนได้มากยิ่งขึ้นโดยการกดปุ่ม + ด้านล่างขวาของแผนที่หรือ Ctrl + Scroll เม้าส์



2.1.2. และสามารถการซูมออกมากเพื่อดูรูปที่ใหญ่ผู้รับซื้อภายในระยะเวลาเดือนนั้น โดยการกดปุ่ม – ด้านล่างขวาของแผนที่หรือ Ctrl + Scroll เม้าส์



2.2. หากภายในรหัสไปรษณีย์นั้นไม่มีข้อมูล ก็จะขึ้นข้อความสีแดงข้างล่างว่า No data was found!

