



ถังขยะรีไซเคิลอัจฉริยะรักษ์โลก

reTrash : The Smart Recycling Bin System for environment

นางสาวพัฒนิตา กาญจนสุโรจน์

Pannita Kanjanasurot

นายรัชชานนท์ แสนเสส

Ratchanon Sanses

นายวทัณญญ ลิปิกรธนางกูร

Wathunyu Lipikornthanangkul

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ปีการศึกษา พ.ศ. 2563



คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ

ชื่อหัวข้อโครงการ

ถังขยะรีไซเคิลอัจฉริยะรักษโลก

reTrash : The Smart Recycling Bin System for environment

นิสิต

นางสาวพัฒนา กาญจนสุโรจน์ 60102010572

นายรัชชานนท์ แสนเสส 60102010577

นายวทัณญญ ลิปิกรธนางกูร 60102010578

ปริญญา

วิทยาศาสตร์บัณฑิต (วท.บ.)

ภาควิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาโครงการ

อ.ดร.วีระ สอิ่ง

ลงชื่อ.....

(อ.ดร.วีระ สอิ่ง)

อาจารย์ที่ปรึกษาโครงการ

บทคัดย่อ

งานวิจัยนี้จะนำเสนอระบบถังขยะรีไซเคิลอัจฉริยะแบบใหม่ที่พัฒนาจากโครงการ Trashy ที่ใช้ IoT ระบบนี้เสนอให้มีการจัดการขยะซึ่งเป็นหนึ่งในปัญหาหลักของโลก นอกจากนี้การจัดการขยะยังนำไปสู่ปัญหามากมาย เช่น ปัญหาสุขภาพอันเนื่องมาจากเชื้อโรคและภาวะโลกร้อน ทางออกที่เป็นไปได้สำหรับปัญหานี้คือการแยกประเภทขยะแต่ละประเภทให้เหมาะสม

อย่างไรก็ตามระบบการคัดแยกขยะจำนวนมากยังไม่มีประสิทธิภาพเพียงพอและทำให้ผู้คนสับสนเกี่ยวกับวิธีการแยกประเภทของขยะ ดังนั้นงานวิจัยนี้จะเสนอโอกาสในการ reTrash เพื่อดัดแปลงโครงการต้นแบบที่สามารถแยกประเภทของขยะโดยอัตโนมัติ ได้แก่ แก้ว พลาสติก กระจก โลหะ และขยะอื่น ๆ ระบบที่นำเสนอใช้ Raspberry pi กับคอนโทรลเลอร์หลัก นอกจากนี้ระบบนี้ยังจำแนกขยะจากการเรียนรู้ภาพถ่ายก่อนหน้านี้โดยใช้แบบจำลอง resnet50 เพื่อวิเคราะห์ภาพและจำแนกขยะ จากนั้นของเสียจะถูกส่งไปยังถังขยะโดยใช้มอเตอร์ขับเคลื่อน จากผลการทดลองระบบที่นำเสนอแสดงถึงความแม่นยำถึง 98% จากการทดสอบและ 84% จากการนำไปใช้จริง

Abstract

This research will propose a new smart recycling bin system that develops from the Trashy project that is an IoT - based. This system proposed to waste management for being one of the main problems in the world. In addition, the waste management leads to many problems such as health issues due to germs and global warming. A possible solution for this problem is to properly classify each type of garbage. However, the many waste separation systems are not efficient enough and people confuse about how to separate the type of garbage. So, this research will propose a reTrash prospect to adapt the prototype project that can automatically separate the type of garbage: glass, plastic, metal cans, and other waste. The proposed system uses the Raspberry pi to the main controller. In addition, this system classifies a waste from learning previous photos by using the resnet50 model to analyze the images and classify the garbage. And then, a waste is sent to the bin by motor drive. From the experimental results, the proposed system represents the accuracy to 98% from testing and 84% from real implementation.

กิตติกรรมประกาศ

โครงการ ถังขยะรีไซเคิลอัจฉริยะรักษ์โลก (reTrash : The Smart Recycling Bin System for environment) จะเกิดขึ้นไม่ได้หากไม่ได้รับการช่วยเหลือดูแล และความร่วมมือจากหลายฝ่าย อันได้แก่

ขอขอบพระคุณ อ.ดร.วีระ สอึ้ง ที่คอยสนับสนุนความคิด และกำลังทรัพย์ เพื่อช่วยแต่งเติมโครงการให้เกิดขึ้นมาได้

ขอขอบพระคุณอาจารย์ทุกท่านที่ให้คำแนะนำ ทิชม ให้ข้อคิดเห็นในการทำงาน และให้ความรู้ต่างๆ ตั้งแต่ได้เข้ามาศึกษาจนกระทั่งถึงโครงการสุดท้าย

ขอขอบพระคุณเพื่อนๆ ภาควิชาวิทยาการคอมพิวเตอร์ที่อยู่ด้วยกัน และได้แบ่งปันช่วยเหลือในงานต่างๆ

ขอขอบพระคุณมหาวิทยาลัยศรีนครินทรวิโรฒที่เป็นสถานที่แห่งความรู้ ความสามารถ อันนำไปสู่การทำงานในภายภาคหน้า

ขอขอบพระคุณพ่อแม่ และครอบครัว ที่สนับสนุนกำลังใจ กำลังทรัพย์ เปิดโอกาสให้ได้ศึกษาเล่าเรียนในสิ่งที่สนใจ

สุดท้ายนี้ ขอขอบพระคุณทุกท่านที่มีส่วนเกี่ยวข้อง ที่ทำให้การทำงานครั้งนี้เกิดขึ้นมาได้ คณะผู้จัดทำหวังว่าโครงการถังขยะรีไซเคิลอัจฉริยะรักษ์โลก (reTrash : The Smart Recycling Bin System for environment) จะเป็นประโยชน์ต่อการศึกษาและพัฒนาต่อยอดได้อีกต่อไป

สารบัญ

บทคัดย่อ	ค
ABSTRACT	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
สารบัญภาพ	ฅ
สารบัญตาราง	ญ
บทที่ 1	1
บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2	3
องค์ความรู้ที่เกี่ยวข้อง.....	3
2.1 ทฤษฎี และแนวคิดที่ใช้ในการทำงานวิจัย	3
2.1.1 Convolutional Neural Network (CNN)	3
2.1.2 Residual Network (ResNet).....	3
2.1.3 Raspberry Pi	4
2.1.4 IR Infrared Obstacle Detection Sensor	5
2.1.5 Arduino Uno R3.....	5
2.1.6 Ultrasonic Sensor	6
2.1.7 Force Sensitive Resistor	6
2.1.8 Inter Integrate Circuit (IIC).....	7
2.2 งานวิจัยที่เกี่ยวข้อง	8

2.2.1 แทรกซี้ ถังขยะอัจฉริยะเพื่อการรีไซเคิล Trashy : An IoT - based Smart Recycling Bin System	8
2.2.2 Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling	8
2.2.3 ระบบติดตามถังขยะอัจฉริยะ Smart Trash Tracking System	8
2.2.4 RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks	9
บทที่ 3	10
วิธีการดำเนินโครงการ	10
3.1 ขั้นตอนการดำเนินงาน	10
3.1.1 ศึกษาค้นคว้าเอกสาร และงานวิจัยที่เกี่ยวข้อง	10
3.1.2 ศึกษาวิธีการทำงานของ CNN เพื่อมาประยุกต์ใช้กับชุดข้อมูลภาพขยะ.....	10
3.1.3 ศึกษาวิธีการใช้งาน Raspberry pi ในการเชื่อมต่อมอเตอร์	10
3.1.4 ศึกษาวิธีใช้งานเซนเซอร์	10
3.1.5 ออกแบบถังขยะ เพื่อให้ตรงตามการใช้งาน	10
3.1.6 ทดสอบการทำงานของถังขยะ	10
3.1.7 สรุปผลการวิจัย เสนอแนะ และจัดทำรูปเล่ม	11
3.2 แผนการดำเนินงานตลอดโครงการ	12
3.3 อุปกรณ์และเครื่องมือ	13
3.3.1 ฮาร์ดแวร์ (Hardware)	13
3.3.2 ซอฟต์แวร์ (Software).....	13
3.3.3 ภาษาที่ใช้ในการพัฒนา (Language).....	13
3.4 การออกแบบและพัฒนา.....	14
3.4.1 การเตรียมชุดข้อมูล (Data Set)	14
3.4.2 โครงสร้างของถังขยะ	15
3.4.3 ขั้นตอนการทำงานของระบบ	16
3.4.4 ปัญหาและอุปสรรค	17
บทที่ 4	18
ผลการดำเนินโครงการ.....	18

4.1 การปรับปรุงโครงสร้างของโครงการใหม่.....	18
4.1.1 การเพิ่มส่วนของการ cover	18
4.1.2 ประตูลับขยะอัตโนมัติ.....	19
4.2 จอแสดงผล	20
4.3 สรุปผลเวลาที่ใช้ในการทำงานทั้งหมดของถังขยะ	22
4.4 การทำงานของ Force Sensitive Resistor	22
บทที่ 5	25
สรุปผล อภิปรายผล และข้อเสนอแนะ	25
5.1 สรุปผลการศึกษาค้นคว้า.....	25
5.1.1 รูปร่างโครงการ.....	25
5.1.2 ส่วนแสดงผลและจอภาพ	25
5.1.3 การวัดน้ำหนักโดยใช้ Force Sensitive Resistor	25
5.2 อภิปรายผล	25
5.3 ข้อเสนอแนะ	26
บรรณานุกรม.....	27
ภาคผนวก	29
ภาคผนวก ก.....	30
คู่มือการติดตั้ง ใช้งาน	30
1. การติดตั้ง Raspberry Pi OS เพื่อใช้บน Raspberry Pi 4	30
2. การเปิดใช้งาน Raspberry Pi Camera	33
ภาคผนวก ข	34
ภาคผนวก ค.....	68
ตารางข้อมูลน้ำหนักขยะแต่ละประเภท (พลาสติก, แก้ว, โลหะ).....	68
ภาคผนวก ง.....	69
ภาพเกี่ยวกับการทำงาน	69

สารบัญภาพ

ภาพที่ 1 : สถาปัตยกรรมของ CONVOLUTIONAL NEURAL NETWORK	3
ภาพที่ 2 : สถาปัตยกรรมของ RESNET50.....	4
ภาพที่ 3 : RASPBERRY PI.....	4
ภาพที่ 4 : IR INFRARED OBSTACLE DETECTION SENSOR	5
ภาพที่ 5 : ARDUINO UNO R3	5
ภาพที่ 6 : ULTRASONIC SENSOR.....	6
ภาพที่ 7 : FORCE SENSITIVE RESISTOR (FSR).....	6
ภาพที่ 8 : การทำงานของตัวตรวจจับแรงกด FSR	7
ภาพที่ 9 : การเชื่อมต่ออุปกรณ์แบบ I ² C.....	7
ภาพที่ 10 : การออกแบบโครงสร้างถังขยะ ครั้งที่ 1.....	15
ภาพที่ 11 : ขั้นตอนการทำงานของระบบ	16
ภาพที่ 12 : รูปร่างโครงงานหลังการปรับปรุง	18
ภาพที่ 13 : การออกแบบโครงสร้างถังขยะ ครั้งที่ 2.....	19
ภาพที่ 14 : ลักษณะการใส่ขยะ	19
ภาพที่ 15 : ภาพที่แสดงบนหน้าจอขณะทำงาน 1.....	20
ภาพที่ 16 : ภาพที่แสดงบนหน้าจอขณะทำงาน 2.....	20
ภาพที่ 17 : ภาพที่แสดงบนหน้าจอขณะทำงาน 3.....	21
ภาพที่ 18 : กระบวนการทำงานของระบบ	23
ภาพที่ 19 : คำนวณน้ำหนักขยะที่เซนเซอร์รับมา	23
ภาพที่ 21 : การทดลองการใช้ FORCE SENSITIVE RESISTOR มาใช้ในการวัดน้ำหนักแทน LOADCELL.....	24
ภาพที่ 20 : การทดลองการใช้ FORCE SENSITIVE RESISTOR มาใช้ในการวัดน้ำหนักแทน LOADCELL.....	24

สารบัญตาราง

ตารางที่ 1 : แผนการดำเนินงาน	12
ตารางที่ 2 : ชุดข้อมูล TRASHNET	14
ตารางที่ 3 : ชุดข้อมูลตัวอย่างน้ำนักขยะ	15
ตารางที่ 4 : ผลเวลาที่ใช้ในการทำงานทั้งหมดของถังขยะแต่ละถัง.....	22



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันประเทศไทยมีปริมาณขยะสูงขึ้นต่อเนื่องขึ้นทุกปี จากรายงานสถานการณ์ขยะมูลฝอยชุมชนของประเทศไทย พบว่ามีปริมาณการเกิดขยะรวมกันทั้งประเทศ 27.06 ล้านตันต่อปี เทียบเท่าตึกไบฮอก 2 จำนวน 140 ตึก คิดเป็นประมาณ 74,130 ตันต่อวัน เฉลี่ยเป็นปริมาณขยะ 1.14 กิโลกรัมต่อคนต่อวัน ยังไม่รวมขยะตกค้างสะสมที่เพิ่มขึ้นทุกปีไม่ต่ำกว่าปีละ 10 ล้านตัน

จากปัญหานี้การแยกขยะที่ยังสามารถใช้ได้ใหม่อย่างการ recycle เป็นตัวช่วยที่จะสามารถลดอัตราการเพิ่มขึ้นของขยะในประเทศไทยได้ อีกทั้งช่วยในเรื่องการอนุรักษ์สิ่งแวดล้อมและลดภาวะเรือนกระจก แต่ถึงอย่างนั้นคนไทยทำการแยกขยะน้อยมากจึงทำให้ขยะที่ควรรีไซเคิลได้สูญหายไปโดยเปล่าประโยชน์

ดังนั้นกลุ่มนิสิตจึงนำเสนอโครงการถังขยะอัจฉริยะนี้เพื่อเป็นตัวช่วยในการแยกประเภทขยะรีไซเคิล โดยตัวโครงการจะทำการแยกขยะประเภทที่สามารถรีไซเคิลได้เป็นหลัก เช่น แก้ว ขวดพลาสติก กระดาษ กระป๋องโลหะ

1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 พัฒนากลังขยะที่สามารถจำแยกขยะประเภทรีไซเคิลทั้ง 4 ประเภท ได้อย่างมีประสิทธิภาพ
- 1.2.2 ศึกษาการใช้ Deep learning เพื่อมาวิเคราะห์ภาพในการจำแนกขยะ
- 1.2.3 ศึกษา Embedded system เพื่อใช้ในการควบคุมถังขยะ
- 1.2.4 ศึกษาเทคโนโลยีเซนเซอร์ที่ใช้ในการตรวจจับแรงกดของขยะ

1.3 ขอบเขตของโครงการ

- 1.3.1 สามารถรับขยะได้ 1 ประเภท ต่อ 1 รอบการทำงานของถังขยะอัจฉริยะ
- 1.3.2 สามารถจำแนกขยะที่เป็นแก้ว พลาสติก กระป๋องโลหะ และขยะอื่นๆ ได้
- 1.3.3 สามารถแสดงผลการทำงานผ่านหน้าจอแสดงผลได้
- 1.3.4 ตรวจสอบประเภทขยะโดยการจำแนกจากข้อมูลภาพ
- 1.3.5 ตรวจสอบน้ำหนักของขยะเพื่อจำแนกขยะให้มีความแม่นยำมากขึ้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ได้เป็นตัวช่วยให้สามารถแยกขยะได้ถูกประเภท
- 1.4.2 พัฒนถังขยะให้สามารถใช้ได้ในชีวิตจริง
- 1.4.3 เรียนรู้เกี่ยวกับ Machine Learning
- 1.4.4 เรียนรู้เกี่ยวกับ Embedded system



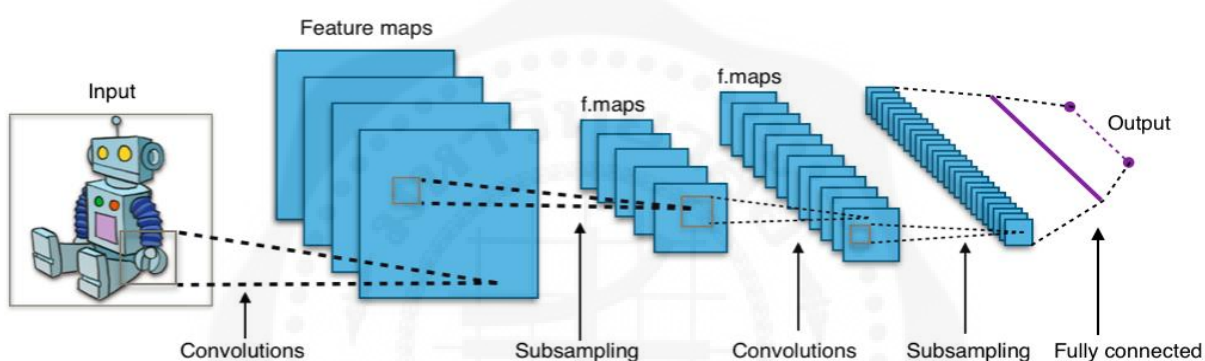
บทที่ 2

องค์ความรู้ที่เกี่ยวข้อง

2.1 ทฤษฎี และแนวคิดที่ใช้ในการทำงานวิจัย

2.1.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) คือ การวิเคราะห์รูปภาพที่มนุษย์มองเห็น โดยจะแบ่งรูปภาพออกเป็นพื้นที่ย่อยๆ เป็น Pixel แต่ละอันเพื่อทำการวิเคราะห์ Metric ของรูปภาพ โดยถ้าเป็นรูปภาพสีขาวดำ จะเป็น Metric 2×2 แต่ถ้าเป็นภาพสีจะเป็น Metric 3×3



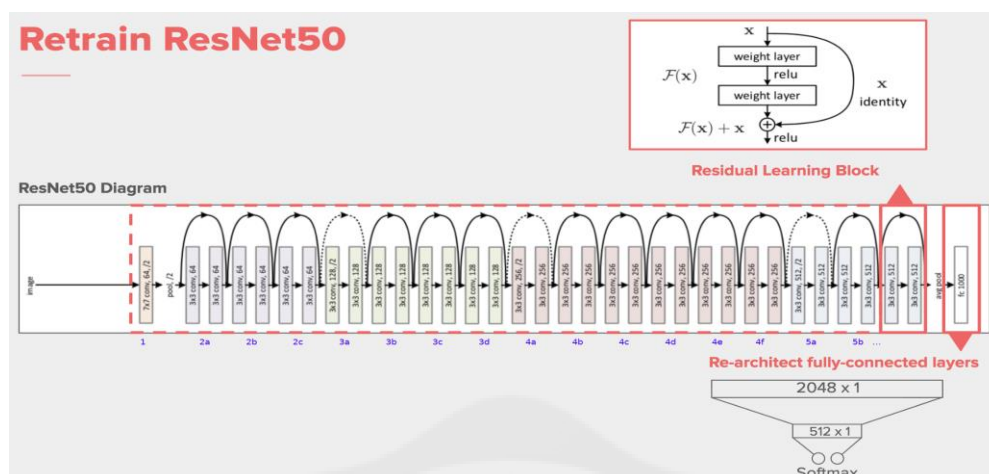
ภาพที่ 1 : สถาปัตยกรรมของ Convolutional Neural Network

2.1.2 Residual Network (ResNet)

ชื่อเต็มของ ResNet คือ Deep Residual Network ได้รับการนำเสนอครั้งแรกในงานวิจัยชื่อ Deep Residual Learning for Image Recognition ได้รับการอ้างอิง 3,700 ครั้ง ในฐานข้อมูล Scopus 41,537 ครั้ง ใน Google Scholar (Jan 2016 — Mar 2020) นำเสนอวิธีการแก้ปัญหาเรื่อง Vanishing gradient ซึ่งเกิดขึ้นกับโครงข่ายที่มีความลึกค่อนข้างมาก ด้วยการใส่ทางลัด (shortcut) ลงในโครงข่าย

โดยคณะผู้จัดทำได้เลือกใช้ ResNet50 เพราะจากผลการทดลองจากหลายๆ โมเดล โมเดลนี้มีความถูกต้องในการนำไปประยุกต์ใช้จริงมากกว่าโมเดลอื่นๆ ซึ่งโมเดลนี้มีโครงสร้างของ ResNet ที่ประกอบด้วย 50 layers การนับจำนวนชั้นก็จะต่างกันไปในแต่ละคนที่ออกแบบดังภาพที่ 2 จะเห็นได้ว่า

มีการทำงาน 50 layers แต่ละ layer จะมีการกำหนดค่าต่าง เช่น Convolution, Activation function และอื่นๆ รวมถึงการปรับค่าอย่าง Learning rate เป็นต้น



ภาพที่ 2 : สถาปัตยกรรมของ ResNet50

2.1.3 Raspberry Pi

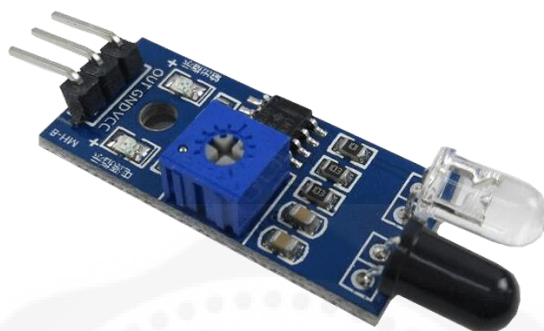
Raspberry Pi บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงการทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก รองรับระบบปฏิบัติการลินุกซ์ โดยติดตั้งบน SD Card



ภาพที่ 3 : Raspberry Pi

2.1.4 IR Infrared Obstacle Detection Sensor

เซ็นเซอร์ตรวจจับวัตถุสิ่งกีดขวางและเส้นขาดำแบบอินฟราเรด IR Infrared Obstacle Detection Sensor เซ็นเซอร์ใช้ตรวจจับวัตถุโดยใช้หลักการสะท้อนของแสงเมื่อไปชนวัตถุ (Reflective) สามารถปรับความไวในการตรวจจับได้ ใช้แสงอินฟราเรดในการตรวจจับ



ภาพที่ 4 : IR Infrared Obstacle Detection Sensor

2.1.5 Arduino Uno R3

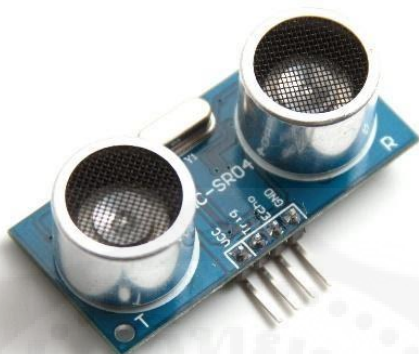
บอร์ดไมโครคอนโทรลเลอร์แบบ Open-source บนแพลตฟอร์ม Arduino โดยใช้ชิป ATmega328 รั้นที่ความถี่ 16 MHz หน่วยความจำแฟลช 32 KB แรม 2 KB บอร์ดใช้ไฟเลี้ยง 7 ถึง 12V มีระดับแรงดันไฟฟ้าในการทำงานและขาสัญญาณอยู่ที่ 5V (TTL) มี Digital Input / Output 14 ขา (เป็น PWM ได้ 6 ขา) มี Analog Input 6 ขา Serial UART 1 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบนซอฟต์แวร์ Arduino IDE และโปรแกรมผ่านพอร์ต USB



ภาพที่ 5 : Arduino Uno R3

2.1.6 Ultrasonic Sensor

Ultrasonic Sensor คือ เซ็นเซอร์ที่ใช้สำหรับตรวจจับวัตถุต่างๆ โดยอาศัยหลักการสะท้อนของคลื่นความถี่เสียง และ คำนวณหาค่าระยะทางได้จากการเดินทางของคลื่นและนำมาเทียบกับเวลา ด้วยกลไกดังกล่าวทำให้เราสามารถนำมาประยุกต์ใช้งานในรูปแบบต่างๆได้อย่างมากมาย เช่น งานวัดระดับน้ำ งานตรวจจับชิ้นงาน งานตรวจจับความหนาของวัตถุ



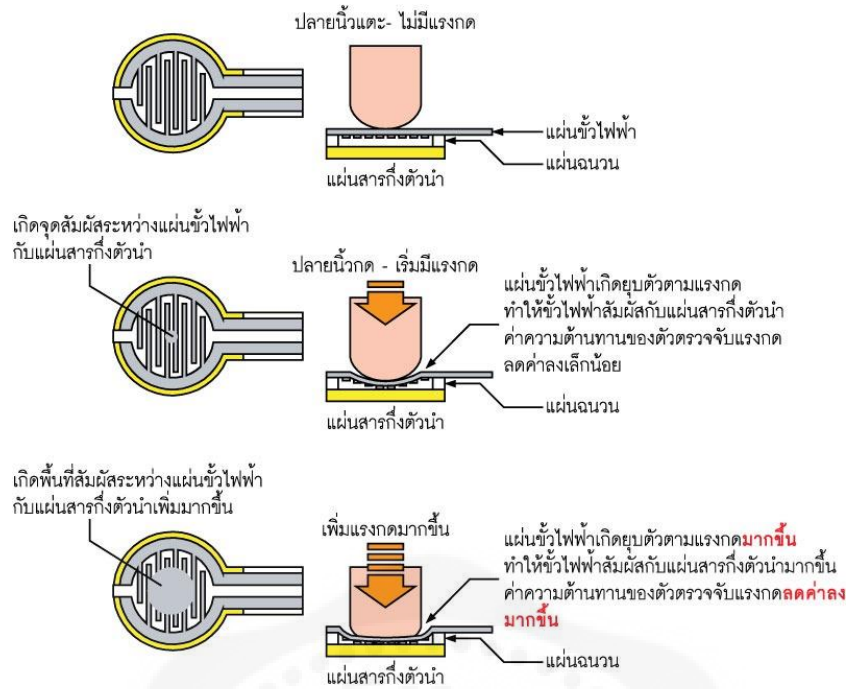
ภาพที่ 6 : Ultrasonic Sensor

2.1.7 Force Sensitive Resistor

ตัวตรวจจับแรงกดหรือ force sensor มีหลายประเภท สำหรับงานในระดับนักเรียนทดลองที่นำมาเสนอนี้เป็นตัวตรวจจับแรงกดแบบค่า ความต้านทานหรือ Force Sensing Resistor (FSR) ที่ใช้เทคโนโลยีฟิล์มโพลีเมอร์แบบหนา (Polymer Thick Film) โดยแรงดันไฟฟ้าที่ตกคร่อมตัวตรวจจับจะลดลง เมื่อมีแรงกดมากกระทำบนแผ่นตรวจจับ มีโครงสร้างของตัวตรวจจับแสดงในรูปที่ประกอบด้วยแผ่นสารกึ่งตัวนำแบบอ่อนที่เป็นตัวกำหนดค่าความต้านทานไฟฟ้าประกบ เข้ากับแผ่นขั้วไฟฟ้าแบบอ่อน โดยมีแผ่นฉนวนแบบอ่อนคั่นกลาง ทำให้เกิดค่าความต้านทานไฟฟ้าขึ้นระหว่างขาต่อใช้งาน เมื่อมีการกดลงบนแผ่นขั้วนำไฟฟ้า จะทำให้เกิดการสัมผัสระหว่างสารกึ่งตัวนำกับขั้วไฟฟ้า ส่งผลให้ค่าความต้านทานไฟฟ้าเกิดการเปลี่ยนแปลง



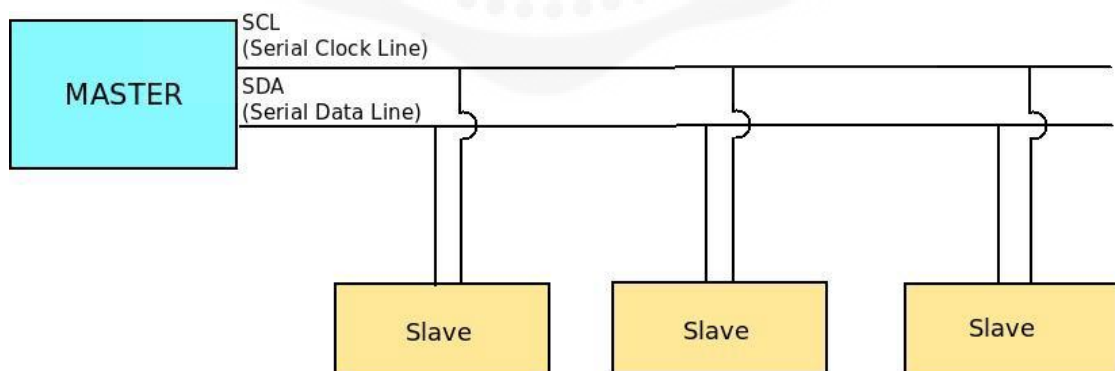
ภาพที่ 7 : Force Sensitive Resistor (FSR)



ภาพที่ 8 : การทำงานของตัวตรวจจับแรงกด FSR

2.1.8 Inter Integrate Circuit (IIC)

I²C ย่อมาจาก Inter Inegrate Circuit (IIC) แต่นิยมเรียกว่า I²C มากกว่า เป็นการสื่อสารอนุกรมแบบ synchronous เพื่อใช้สื่อสารข้อมูลระหว่างอุปกรณ์ความเร็วต่ำ I²C มีความโดดเด่นที่การเชื่อมต่อกันเป็นระบบบัสแบบรับและส่งข้อมูล ซึ่งสามารถเชื่อมต่อกับอุปกรณ์จำนวนมากได้โดยใช้สายเพียงแค่ 2 เส้น ทำให้ลดจำนวนของสายไฟและขนาดของอุปกรณ์ไปได้ จึงทำให้เป็นการสื่อสารที่นิยมสำหรับระบบฝังตัว (embedded system) ขนาดเล็ก



ภาพที่ 9 : การเชื่อมต่ออุปกรณ์แบบ I²C

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 แทรชชี ถึงขยะอัจฉริยะเพื่อการรีไซเคิล Trashy : An IoT - based Smart Recycling Bin System

ผู้แต่ง : สุภาวินี บัวทอง, พิมพ์ฝัน แสนเรืองเดช, ขวัญชนก ผนวกสุข

โครงการนี้นำเสนอถึงขยะอัจฉริยะที่สามารถแยกขยะได้ โดยเราจะทำการแยกขยะออกเป็น 4 ประเภท คือ แก้ว พลาสติก กระจกโลหะ และขยะอื่นๆ ซึ่งจะเน้นไปที่ขยะที่สามารถนำไปรีไซเคิลได้ ประสิทธิภาพเบื้องต้นของการคัดแยกประเภทขยะอยู่ที่ประมาณร้อยละ 98 มีการใช้ Ultrasonic sensor เพื่อวัดปริมาณขยะภายในถัง โดยจะมีการแจ้งเตือนให้เจ้าหน้าที่ทราบ เมื่อปริมาณขยะใกล้เต็มถึงและมีการเชื่อมต่อฐานข้อมูล AWS S3 เพื่อเก็บข้อมูลภาพหลังจากการแยกประเภทนั้นๆ

2.2.2 Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling

ผู้แต่ง : Yinghao Chu, Chen Huang, Xiaodan Xie, Bohai Tan, Shyam Kamal and Xiaogang Xiong

งานวิจัยนี้เป็นการศึกษาเกี่ยวกับการแยกขยะโดยใช้รูปภาพนำมาผ่านกระบวนการมัลติเลเยอร์ไฮบริดเลิร์นนิ่ง โดยกระบวนการจะมีเฉพาะอัลกอริทึมและ Sensor ต่างๆ ที่นำมาช่วยให้การจำแนกประสิทธิภาพในการจำแนกขยะเพิ่มขึ้น ได้แก่ Bridge sensor ช่วยในเรื่องของการวัดน้ำหนัก Inductor ใช้สำหรับวัดโลหะ

2.2.3 ระบบติดตามถังขยะอัจฉริยะ Smart Trash Tracking System

ผู้แต่ง : ทิพานัน พงษ์สุวรรณ, อนุพงษ์ ดิตะ และ ภาณุวัตร อุทัยบาล

ระบบติดตามถังขยะอัจฉริยะ เป็นระบบที่ใช้ในการจัดการปัญหาขยะที่เต็มก่อนกำหนด และมีกลิ่นเหม็นโดยมีการแจ้งเตือนส่วนกลางให้ทราบถึงถังขยะที่ใกล้จะเต็มแล้ว ขยะมีกลิ่นเหม็น หรือตรวจสอบว่าขยะที่เต็มแล้วนั้นถูกวางไว้จุดใด โดยการใช้เซนเซอร์แสงอินฟราเรดติดตั้งจุดต่างๆ เพื่อวัดปริมาณขยะในถัง การนำเซนเซอร์วัดคุณภาพอากาศเพื่อตรวจสอบกลิ่นเหม็นที่ออกมาจากขยะ จากนั้นจะส่งตำแหน่งของถังขยะ โดยใช้ระบบนำทาง (GPS) ข้อมูลปริมาณและกลิ่นของขยะผ่านทางอุปกรณ์ตรวจจับแล้วนำส่งข้อมูลไปยังเครื่องแม่ข่าย (Server) และมีระบบรายงานแจ้งเตือนทั้งทางเว็บ และแอปพลิเคชันบนมือถือ เพื่อบริหารจัดการและแก้ไขปัญหาขยะในห้องสมุด

2.2.4 RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks

ผู้แต่ง : Cenk Bircanoğlu, Meltem Atay, Fuat Beşer, Özgün Genç and Merve Ayyüce Kızrak

งานวิจัยนี้เป็นการนำ Convolution Neural Network มาใช้ในการจำแนกขยะของชุดข้อมูล TrashNet ซึ่งจากการวิจัยนี้ได้ลองทดสอบโดยใช้ CNN แบบต่างๆ มาเปรียบเทียบ แล้วได้ทำการบันทึกผล ความแม่นยำของmodel เหล่านั้น ซึ่งการวิจัยนี้ได้ให้ความเห็นว่า model เหล่านั้นมีความแม่นยำ แต่ เวลาในการทำนายค่อนข้างช้า ทางคณะผู้จัดทำจึงได้มีการนำเสนอ model แบบใหม่ที่มีความรวดเร็วในการทำนายมากขึ้นเพื่อให้เหมาะกับการใช้งานแบบเรียลไทม์



บทที่ 3

วิธีการดำเนินโครงการ

3.1 ขั้นตอนการดำเนินงาน

3.1.1 ศึกษาค้นคว้าเอกสาร และงานวิจัยที่เกี่ยวข้อง

เป็นขั้นตอนการสืบค้นเอกสาร และงานวิจัยที่เกี่ยวข้องเกี่ยวกับเรื่องที่คุณผู้จัดทำสนใจ และทำการคัดเลือกงานวิจัยที่เกี่ยวข้อง เพื่อนำมาเป็นแนวทางในการทำโครงการและกำหนดหัวข้อโครงการ

3.1.2 ศึกษาวิธีการทำงานของ CNN เพื่อมาประยุกต์ใช้กับชุดข้อมูลภาพขยะ

จากการได้ศึกษาเอกสารงานวิจัยที่เกี่ยวข้องกับโครงการ ในเอกสารงานวิจัยได้มีการใช้ CNN จึงต้องทำการศึกษา CNN เพื่อความเข้าใจและนำไปประยุกต์ใช้ในกระบวนการทำงาน

3.1.3 ศึกษาวิธีการใช้งาน Raspberry pi ในการเชื่อมต่อมอเตอร์

การศึกษารูปแบบการทำงานและ ภาษาที่ใช้ในการเขียนใน Raspberry pi จำเป็นต้องศึกษาเพราะ Raspberry pi จะเป็นตัวควบคุมการทำงานหลักของชิ้นงาน

3.1.4 ศึกษาวิธีใช้งานเซนเซอร์

ส่วนเพิ่มเติมหลักคือการใช้ Force Sensitive Resistor และ IR Infrared Obstacle Detection Sensor จึงต้องทำการศึกษาค้นคว้าและ ทดลองเซนเซอร์ต่างๆเพื่อนำมาใช้ในงานวิจัย

3.1.5 ออกแบบถังขยะ เพื่อให้ตรงตามการใช้งาน

เป็นขั้นตอนในการออกแบบรูปร่างถังและการออกแบบกระบวนการทิ้งของถังขยะ โดยคำนึงถึงสภาพแวดล้อมการทำงานจริงกับอุปกรณ์ที่ได้มีการติดตั้งให้สามารถดำเนินการได้ตามสภาพแวดล้อมต่างๆ และงบประมาณในการจัดทำถังขยะอัจฉริยะ

3.1.6 ทดสอบการทำงานของถังขยะ

การทดสอบช่วงแรก : เป็นการทดสอบประสิทธิภาพการทำงานและระบบการทำงานของ Trashy(ผลงานต้นแบบ)ว่าการทำงาน เป็นอย่างไร หรือมีปัญหาอะไรหรือไม่

การทดสอบช่วงหลัง : เป็นการทดสอบหลังจากที่ทำงานวิจัยเพิ่ม และต้องการที่จะทดสอบเซนเซอร์ Force Sensitive Resistor

3.1.7 สรุปผลการวิจัย เสนอแนะ และจัดทำรูปเล่ม



3.3 อุปกรณ์และเครื่องมือ

3.3.1 ฮาร์ดแวร์ (Hardware)

1. คอมพิวเตอร์
2. Ultrasonic sensor รุ่น HC-SR04
3. Raspberry Pi 4 model B Ram 4 GB
4. Raspberry Pi camera V2.1
5. Arduino Uno R3 ATmega328P
6. Motor
7. Relay 5V 2-CH 10A 250VAC
8. LED strip
9. DRV 8255 Stepping motor Drive
10. Force Sensitive Resistor FSR
11. Tower Pro SG90 Mini Micro Servo
12. Monitor
13. Force Sensitive Resistor
14. IR Infrared Obstacle Detection Sensor

3.3.2 ซอฟต์แวร์ (Software)

1. Jupyter Notebook
2. Arduino IDE

3.3.3 ภาษาที่ใช้ในการพัฒนา (Language)

1. Python 3.7.3
2. C

3.4 การออกแบบและพัฒนา

3.4.1 การเตรียมชุดข้อมูล (Data Set)

ตารางที่ 2 : ชุดข้อมูล Trashnet

ประเภทของขยะ	จำนวนภาพ ในแต่ละประเภทขยะ
กระดาษ	594
แก้ว	501
พลาสติก	482
กระป๋องโลหะ	410
กระดาษแข็ง	403
ขยะอื่นๆ	137
รวม	2,527

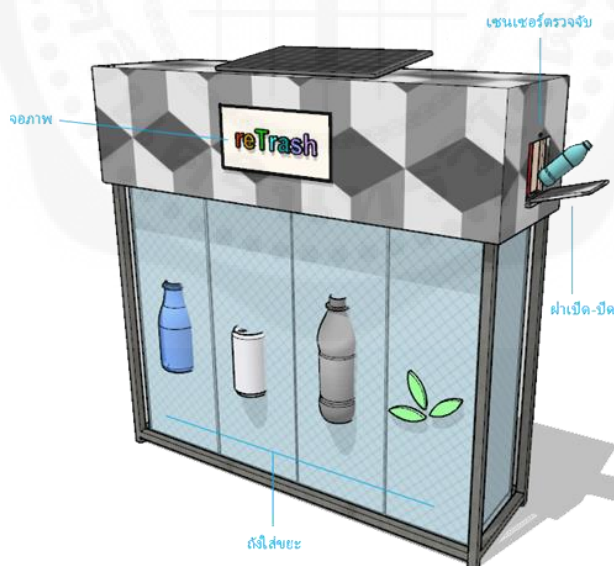
จากตารางที่ 2 เป็นชุดข้อมูล Trashnet ที่มีทั้งหมด 2,527 รูป และแบ่งเป็น 6 ประเภท ได้แก่ กระดาษ แก้ว พลาสติก กระป๋องโลหะ กระดาษแข็ง และขยะอื่นๆ ซึ่งทางคณะผู้จัดทำได้นำชุดข้อมูลดังกล่าวมาทำการดัดแปลง เพื่อให้เหมาะสมกับงานวิจัยนี้ โดยการลดประเภทขยะที่จะคัดแยกให้เหลือ 4 ประเภท ได้แก่ แก้ว พลาสติก กระป๋องโลหะ และขยะอื่นๆ และได้มีการเพิ่มภาพขยะใหม่เข้าไปบางส่วน เพื่อให้ข้อมูลภาพขยะแต่ละประเภทมีจำนวนที่สมดุลกันในทุกประเภทให้สามารถนำมาใช้งานได้อย่างมีประสิทธิภาพ โดยจะมีการจัดสภาพแวดล้อมของภาพใหม่ให้มีสภาพแวดล้อมใกล้เคียงกับภาพในชุดข้อมูล Trashnet

ตารางที่ 3 : ชุดข้อมูลตัวอย่างน้ำหนักขยะ

ประเภท	จำนวน(ชิ้น)	น้ำหนักเฉลี่ย(กรัม)
พลาสติก	20	23.25
แก้ว	7	141.8571429
โลหะ	5	12

จากตารางที่ 3 เป็นข้อมูลน้ำหนักเฉลี่ย(กรัม) ของขยะ 3 ประเภท ได้แก่ พลาสติก แก้ว และ โลหะ ที่ทางคณะผู้จัดทำสร้างขึ้น จะนำไปเป็นแนวทางทดลองและการทำงานร่วมกับ Force Sensitive Resistor

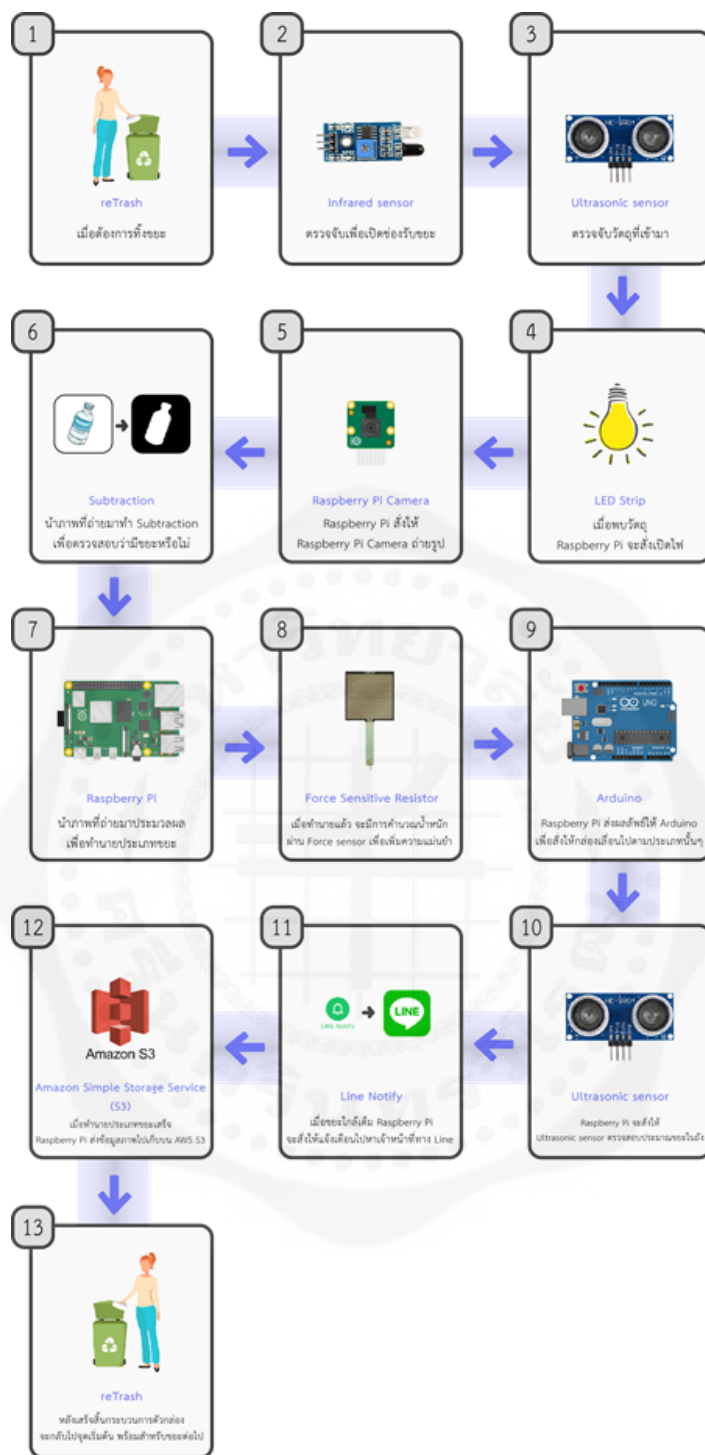
3.4.2 โครงสร้างของถังขยะ



ภาพที่ 10 : การออกแบบโครงสร้างถังขยะ ครั้งที่ 1

ทำการออกแบบโครงสร้างใหม่ให้มีแผ่นครอบรอบตัวถังขยะอัจฉริยะ ปรับปรุงถังขยะให้เป็นวัสดุที่แข็งแรงขึ้น เพิ่มหน้าจอแสดงผล และฝาถังขยะเปิด-ปิดอัตโนมัติทางด้านซ้ายของถังขยะอัจฉริยะ

3.4.3 ขั้นตอนการทำงานของระบบ



ภาพที่ 11 : ขั้นตอนการทำงานของระบบ

จากภาพที่ 10 แสดงการทำงานของถังขยะที่ติดตั้งอุปกรณ์แล้ว ภายใน 1 ถัง จะมี 4 ช่องเก็บขยะ ตามประเภท ดังนี้ แก้ว พลาสติก กระป๋องโลหะ และขยะอื่นๆ เช่น กระดาษ ถุงพลาสติก ภายในถังจะมีกล้องรับขยะไว้เป็นที่พักเพื่อรอการประมวลผล ซึ่งการทำงานของถังจะเป็นดังนี้ เมื่อมีการทิ้ง

ขยะในกล่องรับขยะ Ultrasonic sensor จะทำการตรวจสอบเบื้องต้นว่ามีขยะเข้ามาในกล่องหรือไม่ เมื่อ sensor ตรวจจับว่ามีวัตถุเข้ามา LED Strip จะเปิดไฟ และกล้อง Raspberry pi camera จะถ่ายภาพเพื่อนำภาพไปตรวจสอบอีกครั้งว่ามีวัตถุเข้ามาจริง เนื่องจากที่ต้องมีการตรวจสอบอีกครั้ง เพราะว่า Ultrasonic sensor ที่กลุ่มวิจัยใช้มีรัศมีการตรวจจับที่แคบทำให้บางจุดในกล่องรับไม่สามารถตรวจจับได้ หลังจากนั้นจะนำภาพที่ถ่ายไปทำนายผลบน Raspberry Pi 4 เมื่อได้ผลลัพธ์ของประเภทขยะมา Force Sensitive Resistor จะทำการวัดน้ำหนักของขยะอีกทีเพื่อความแม่นยำ และสุดท้ายตัวกล่องรับจะทำการเลื่อนกล่องไปตามรางแล้วนำขยะไปทิ้งตามประเภทนั้นๆ

3.4.4 ปัญหาและอุปสรรค

1. ตัวงานวิจัยเดิมมีความชำรุดเสียหาย จึงต้องใช้เวลาในการทำการซ่อมแซมส่วนที่เสียหาย เช่น สายไฟหลุดออกจากอุปกรณ์
2. การใช้งาน Force Sensitive Resistor ไม่เป็นไปตามที่คาดการณ์ไว้ ทำให้ต้องศึกษาการใช้งานเพิ่มเพื่อให้เซนเซอร์ใช้งานได้ตรงตามความต้องการ
3. เนื่องจากสถานการณ์โควิด19 ทำให้ในบางช่วงไม่สามารถเข้ามาทำงานที่มหาวิทยาลัยได้ ซึ่งอุปกรณ์ต่างๆในการทำงานวิจัยอยู่ที่มหาวิทยาลัยทั้งหมด การพัฒนาผลงานจึงมีความล่าช้า

บทที่ 4

ผลการดำเนินโครงการ

การดำเนินโครงการในครั้งนี้ผู้จัดทำได้เสนอผลการดำเนินโครงการ โดยแสดงผลลัพธ์ดังต่อไปนี้

4.1 การปรับปรุงโครงสร้างของโครงการใหม่

4.1.1 การเพิ่มส่วนของการ cover

เพื่อให้ถังขยะอัจฉริยะสามารถตั้งอยู่ภายนอกอาคารได้ เพราะแสงจากภายนอกมีผลต่อการวิเคราะห์จำแนกประเภทขยะ และยังเป็น การปกป้องแผงไฟ และระบบ IoT ภายในอีกด้วย ทั้งยังมีการวางแผนในการเสริมล้อเลื่อน เพื่อให้สะดวกต่อการขนย้ายมากยิ่งขึ้น



ภาพที่ 12 : รูปร่างโครงการหลังการปรับปรุง

4.1.2 ประตูรับขยะอัตโนมัติ

เป็นประตูที่อยู่บริเวณด้านขวาของถังขยะอัจฉริยะ ใช้ในการเปิด-ปิดเพื่อรับขยะเข้ามายังกล่องรับขยะ โดยใช้ IR Infrared Obstacle Detection Sensor ในการตรวจจับการเปิด-ปิด



ภาพที่ 13 : การออกแบบโครงสร้างถังขยะ ครั้งที่ 2



ภาพที่ 14 : ลักษณะการใส่ขยะ

4.2 จอแสดงผล

ในการนำไปใช้จริงจะต้องมีส่วนที่สามารถติดต่อ และบอกผู้ที่ใช้งานถึงการทำงานของระบบถึงขยะ ว่าการทำงานสอดคล้องต่อขยะที่ทิ้งลงไปหรือไม่ โดยจะมีจอภาพแสดงภาพที่บ่งบอกชนิดของขยะที่ประมวลผลได้ และ step ที่กำลังทำงานอยู่ แทนการแสดงของโค้ด โดยจะมีภาพดังนี้



ภาพที่ 15 : ภาพที่แสดงบนหน้าจอขณะทำงาน 1



ภาพที่ 16 : ภาพที่แสดงบนหน้าจอขณะทำงาน 2



ภาพที่ 17 : ภาพที่แสดงบนหน้าจอขณะทำงาน 3

4.3 สรุปผลเวลาที่ใช้ในการทำงานทั้งหมดของถังขยะ

จากการนำ Model ที่เราเลือกใส่ลง Raspberry Pi เพื่อให้เกิดการประมวลผลในการทำนายประเภทของขยะ แล้วเชื่อมต่อ Arduino เพื่อควบคุมการทำงานของมอเตอร์ ให้ถังขยะสามารถใช้งานได้อย่างอัตโนมัติ โดยในการทำงานตั้งแต่เริ่มต้นจนถึงสิ้นสุดของขยะแต่ละประเภท จะใช้เวลาในการทำงานที่ต่างกันไป ดังตารางที่ 4

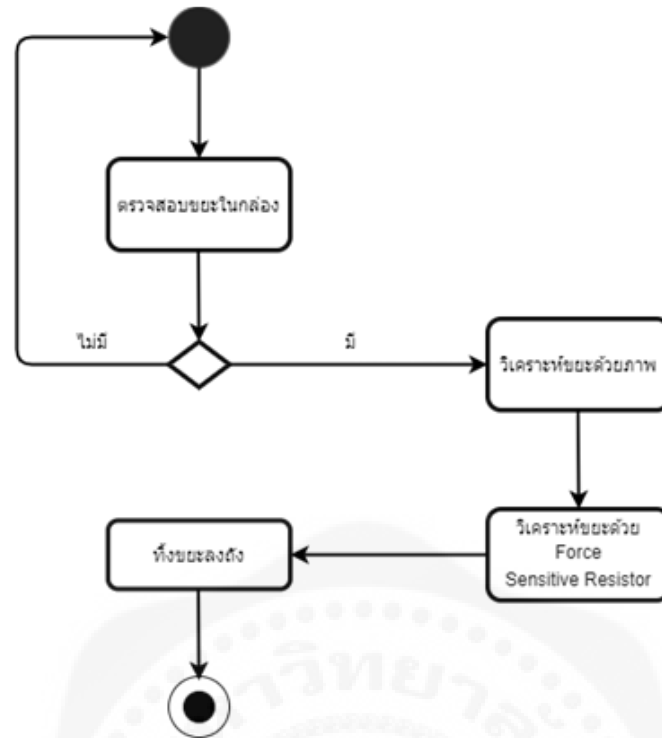
ตารางที่ 4 : ผลเวลาที่ใช้ในการทำงานทั้งหมดของถังขยะแต่ละถัง

ประเภทของขยะ	การเคลื่อนที่ไป และกลับยังถัง (วินาที)	ทั้งหมด (วินาที)
ขยะอื่นๆ	10	21
พลาสติก	18	29
กระป๋องโลหะ	28	39
แก้ว	42	53

จะเห็นได้ว่าจะใช้เวลาค่อนข้างเยอะ หากถังขยะอยู่ไกลจากตำแหน่งรับขยะมาก ซึ่งจากเวลาในการทำงานทั้งหมด จะมีการทำงานย่อยที่เวลาที่เท่ากันคือ การถ่ายภาพขยะในกล่องเป็นเวลา 3 วินาที, การทำนายผลในการจำแนกขยะเป็นเวลาประมาณ 5 - 6 วินาที, การดันขยะให้ลงถังเป็นเวลา 4 วินาที ส่วนที่เหลือคือการทำงานของมอเตอร์ที่แตกต่างกันไป ขึ้นอยู่กับระยะทางไปยังถังขยะประเภทนั้นๆ

4.4 การทำงานของ Force Sensitive Resistor

ทางคณะผู้จัดทำได้เลือก Force Sensitive Resistor ซึ่งเป็นเซนเซอร์วัดแรงกด เพื่อจะวัดน้ำหนักของขยะ โดยจะติดเซนเซอร์ไว้ภายในกล่องรับขยะ และทำการวัดน้ำหนักและประมวลผลหลังจากการประมวลผลประเภทขยะโดยรูปภาพ ดังภาพที่ 18



ภาพที่ 18 : กระบวนการทำงานของระบบ

หลักการการทำงาน คือ เมื่อ Raspberry Pi camera ทำการถ่ายภาพและส่งไปประมวลผลเรียบร้อยแล้ว จะทำนายด้วยการวัดน้ำหนัก โดยจะให้ Force Sensitive Resistor รับค่าแรงกดมา ค่าที่รับมาจะมีหน่วยเป็น Newton ผู้จัดทำจึงทำการแปลงหน่วยให้เป็นกรัม เพื่อให้สะดวกต่อการอ่านค่า และจะทำการเฉลี่ยแรงกด 3 ค่า ให้เป็นน้ำหนักของขยะชิ้นนั้น

```

----- 1.23 newton ----- Force in kg: 41.90 gram -----
----- 1.23 newton ----- Force in kg: 41.90 gram -----
----- 1.22 newton ----- Force in kg: 41.73 gram -----
41.84
  
```

ภาพที่ 19 : ค่าน้ำหนักขยะที่เซนเซอร์รับมา

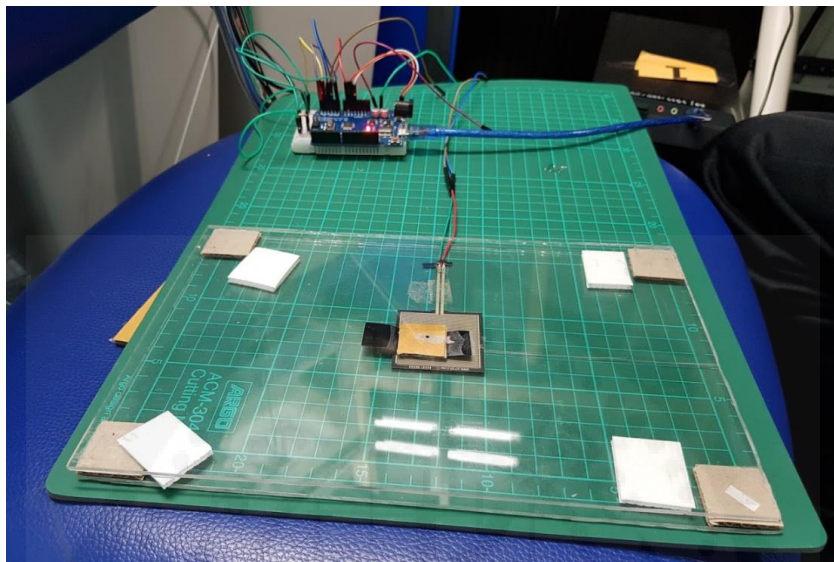
จากนั้นส่งค่าที่ได้ไปให้ Raspberry Pi หลักที่ควบคุมการทำงาน เมื่อ Raspberry Pi ได้รับค่ามา ในโปรแกรมจะมีการกำหนดช่วงน้ำหนักของขยะแต่ละประเภท ช่วงน้ำหนักที่กำหนดนั้นจะได้อาจมาจากชุดข้อมูลจริงที่ผู้จัดทำเก็บรวบรวมมา หากขยะชิ้นนั้นมีน้ำหนักตกอยู่ในช่วงใด ก็จะทำนายขยะเป็นประเภทนั้นๆ

โดยช่วงน้ำหนักของขยะแต่ละประเภทที่ผู้จัดทำเก็บรวบรวมจะอยู่ที่

ช่วงน้ำหนักของขยะประเภท กระป๋องโลหะ จะอยู่ระหว่าง 10 - 13 กรัม

ช่วงน้ำหนักของขยะประเภท ขวดพลาสติก จะอยู่ระหว่าง 13 - 34 กรัม

ช่วงน้ำหนักของขยะประเภท ขวดแก้ว จะอยู่ที่ 60 กรัมขึ้นไป



ภาพที่ 21 : การทดลองการใช้ Force Sensitive Resistor มาใช้ในการวัดน้ำหนักแทน Loadcell



ภาพที่ 20 : การทดลองการใช้ Force Sensitive Resistor มาใช้ในการวัดน้ำหนักแทน Loadcell

บทที่ 5

สรุปผล อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการศึกษาค้นคว้า

จากการพัฒนาโครงการ reTrash : The Smart Recycling Bin System for environment ได้มีการนำ IoT มาใช้ร่วมกับ AI เพื่อคัดแยกขยะ ที่สามารถนำไปรีไซเคิลได้โดยแบ่งเป็น 4 ประเภท ได้แก่ แก้ว พลาสติก กระป๋องโลหะ และขยะอื่นๆ โดยมีการเพิ่ม function อื่นของถังขยะให้ดูเหมาะแก่การใช้งานจริงได้มากยิ่งขึ้น ได้แก่

5.1.1 รูปร่างโครงการ

ได้มีการประกอบส่วนโครงชิ้นใหม่ให้รองรับกับตัวอย่างที่นำมาติดตั้งกับตัวโครงการ ให้มีความสะดวกในการเคลื่อนย้าย และสามารถย้ายไว้สถานที่อื่นได้ และได้มีการทำ cover เพื่อปกป้องกลไกการทำงานของระบบ lot ภายใน ด้วยฟิวเจอร์บอร์ด และยังมีส่วนตัวถังขยะที่สามารถเปลี่ยนไปใช้ถังจริงๆได้

5.1.2 ส่วนแสดงผลและจอภาพ

ได้ทำการต่อระบบให้รองรับจอภาพ และทำการแสดงการทำงานเมื่อรันการทำงานของระบบ เพื่อให้ผู้ใช้งานสามารถทราบถึงวิธีการที่ถังขยะทำงานอยู่

5.1.3 การวัดน้ำหนักโดยใช้ Force Sensitive Resistor

เพิ่ม Force Sensitive Resistor เพื่อให้การทำนายประเภทขยะแม่นยำยิ่งขึ้น เพราะเดิมที่จำแนกขยะด้วยภาพเพียงอย่างเดียว ขยะประเภท ขวดแก้ว และ ขวดพลาสติกมีการทำนายผิดอยู่บ่อยครั้ง เนื่องจากขยะทั้ง 2 ประเภทมีรูปร่างลักษณะคล้ายคลึงกัน

5.2 อภิปรายผล

จากการจัดทำโครงการ reTrash : The Smart Recycling Bin System for environment ในครั้งนี้ เป็นการทำงานต่อยอดจากผลงานเดิม Trashy ซึ่งชิ้นงานเกิดปัญหาและชำรุดอยู่บ่อยครั้ง จึงต้องทำการศึกษา และทำความเข้าใจกระบวนการทำงานของระบบทั้งภายในและภายนอกทั้งหมด ให้ชิ้นงานอยู่ในสภาพที่พร้อมใช้งาน เพื่อจะทำการพัฒนาชิ้นงานให้สมบูรณ์ ตอบสนองต่อความต้องการมากขึ้น โดยส่วนที่ผู้จัดทำได้ทำเพิ่มเติมขึ้นมา ได้แก่ การพัฒนารูปลักษณ์ให้เหมาะสมต่อใช้งานมากขึ้น และทำการเพิ่มระบบการวัดน้ำหนักขยะขึ้นมา เนื่องจากเดิมขยะบางชิ้นมักจะถูกทำนายผิดพลาด เพราะรูปร่างของขยะคล้ายคลึงกันมาก การ

วิเคราะห์ด้วยภาพจึงมีประสิทธิภาพไม่เพียงพอ หลังจากทำการศึกษาเพิ่มเติมพบว่าการจำแนกขยะแต่ละประเภทด้วยน้ำหนักรจะสามารถเพิ่มประสิทธิภาพได้มากขึ้น

5.3 ข้อเสนอแนะ

จากปัญหาการทิ้งขยะได้ที่ละชั้นของโครงการเป็นข้อจำกัดของโครงการนี้ และอาจทำการเพิ่มระบบข้อมูลหรือรางวัลสะสม เพื่อดึงดูดให้ผู้คนอยากทิ้งขยะกับ reTrash มากยิ่งขึ้น



บรรณานุกรม

- [1] สุภาวีนี บัวทอง, พิมพ์ฝัน แสนเรืองเดช, ขวัญชนก ผนวกสุข. (2563). **แทรชซี ถึงขยะอัจฉริยะเพื่อการรีไซเคิล Trashy : An IoT - based Smart Recycling Bin System**. สืบค้นเมื่อ 29 พฤษภาคม 2563, จาก มหาวิทยาลัยศรีนครินทรวิโรฒ
- [2] กรมควบคุมมลพิษและกระทรวงทรัพยากรธรรมชาติและสิ่งแวดล้อม. (2559). **แผนแม่บท การบริหารจัดการขยะมูลฝอยของประเทศ พ.ศ. 2559 – 2564**. สืบค้นเมื่อ 5 มิถุนายน 2563, จาก <http://infofile.pcd.go.th/waste/WasteMasterPlan.pdf?CFID=2748363&CFTOKEN=86290581>
- [3] Aggregated news around AI and co. (2561). **มาลองดูวิธีการคิดของ CNN กัน**. สืบค้นเมื่อ 5 มิถุนายน 2563, จาก <https://mc.ai/มาลองดูวิธีการคิดของ-cnn-ก/>
- [4] diysolarcell. **วิธีการคำนวณ สูตรคำนวณ แบบง่ายๆ แผงโซล่าเซลล์ แบตเตอรี่ คอนโทรลชาร์จ และ อินเวอร์เตอร์ calculate solar cell system**. สืบค้นเมื่อ 5 มิถุนายน 2563, จาก <https://www.diysolarcell.com/calculate-solar-cell-system-%E0%B8%84%E0%B8%B3%E0%B8%99%E0%B8%A7%E0%B8%93%E0%B9%82%E0%B8%B%E0%B8%A5%E0%B9%88%E0%B8%B2%E0%B9%80%E0%B8%8B%E0%B8%A5%E0%B8%A5%E0%B9%8C/>
- [5] Thaieasyelec. (2563). **Arduino Uno R3**. สืบค้นเมื่อ 6 มิถุนายน 2563, จาก <https://www.thaieasyelec.com/arduino-uno-r3.html>
- [6] Chatchawan Niyomthum. (2562). **Convolution Neural Network คืออะไร**. สืบค้นเมื่อ 6 มิถุนายน 2563, จาก <https://medium.com/@pradyasin/what-is-convolution-neural-network-bf2e525089f5>
- [7] Thaieasyelec. (2563). **Load Cell**. สืบค้นเมื่อ 6 มิถุนายน 2563, จาก <https://www.thaieasyelec.com/article-wiki/review-product-article/how-to-use-load-cell-and-hx711-amplifier-module.html>
- [8] Amazon Web Services. (2545). **การประมวลผลบนระบบคลาวด์ด้วย AWS**. สืบค้นเมื่อ 6 มิถุนายน 2563, จาก <https://aws.amazon.com/th/what-is-aws/>
- [9] Amazon Web Services. (2545). **Amazon S3**. สืบค้นเมื่อ 6 มิถุนายน 2563, จาก https://aws.amazon.com/th/s3/?nc2=h_ql_prod_st_s3

- [10] ThaiEasyElec. (2559). **บทความการพัฒนาโปรแกรมบน Raspberry Pi ด้วย Qt.** สืบค้นเมื่อ 6 มิถุนายน 2563, จาก <https://www.thaieasyelec.com/article-wiki/embedded-electronics-application/raspberry-pi-programming-with-qt-ch1.html>
- [11] Gump.in.th. **มาทำความรู้จักชนิดต่างๆ ของแผงโซลาร์เซลล์กัน.** สืบค้นเมื่อ 6 มิถุนายน 2563, จาก <https://www.gump.in.th/article/535>
- [12] netway.co.th. (2562). **Cloud Computing.** สืบค้นเมื่อ 7 มิถุนายน 2563, จาก <https://netway.co.th/kb/blog/Cloud%20%26%20Managed%20Services/360029895251-Cloud%20Computing%20%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3%20>
- [13] Kritsada Jaiyen. **Force sensor with UNICON.** สืบค้นเมื่อ 20 พฤษภาคม 2563, จาก <http://doc.inex.co.th/force-sensor-with-unicon/>
- [14] Gus. **How to Setup a Raspberry Pi Pressure Pad (FSR).** สืบค้นเมื่อ 20 พฤษภาคม 2563, จาก <https://pimylifeup.com/raspberry-pi-pressure-pad/>

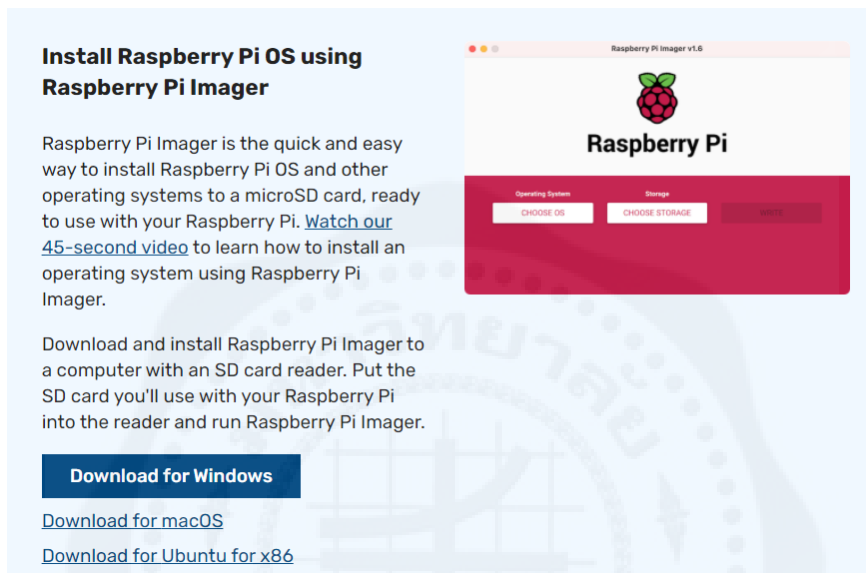


ภาคผนวก ก

คู่มือการติดตั้ง ใช้งาน

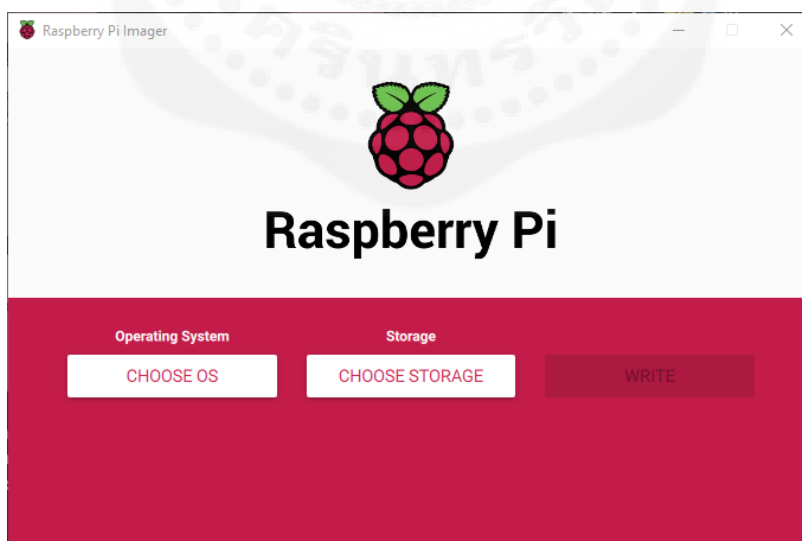
1. การติดตั้ง Raspberry Pi OS เพื่อใช้บน Raspberry Pi 4

1.1 เข้าเว็บไซต์ <https://www.raspberrypi.org/downloads/> ดาวน์โหลดไฟล์ Raspberry Pi Imager ลงในเครื่องคอมพิวเตอร์ ดังภาพที่ ก - 1



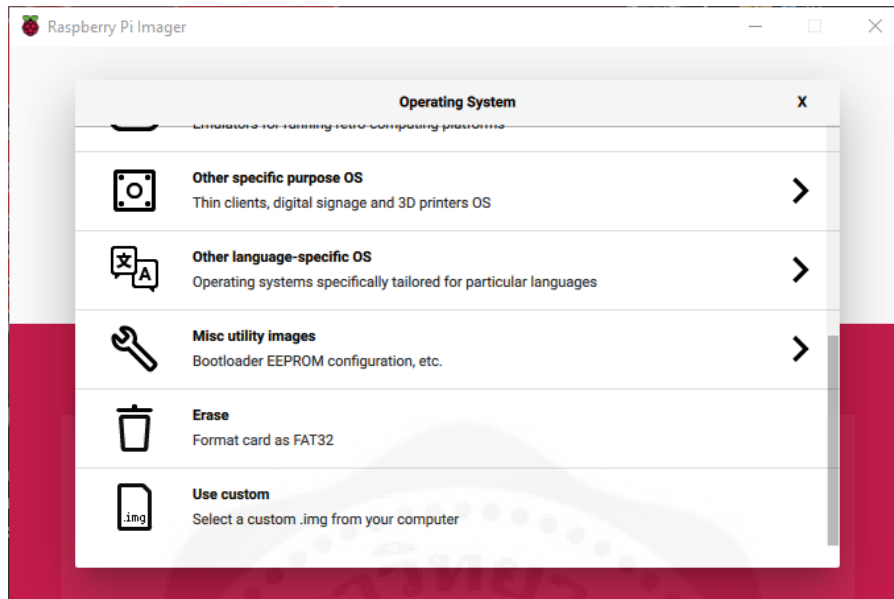
ภาพที่ ก - 1.1 หน้าดาวน์โหลดโปรแกรม

1.2 เสียบ SD Card เข้าเครื่องคอมพิวเตอร์



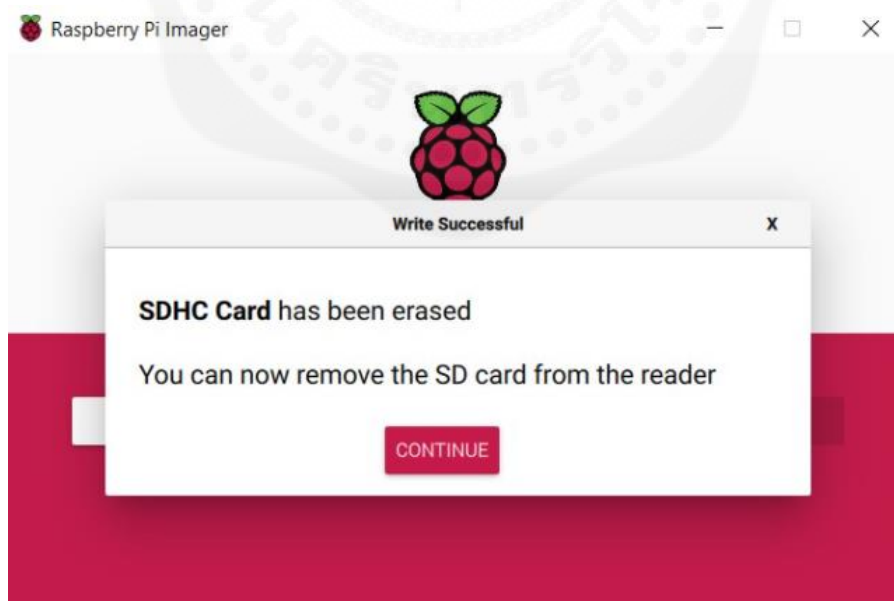
ภาพที่ ก - 1.2 หน้าโปรแกรม Raspberry Pi Imager

1.3 ทำการ format sd card ก่อนใช้งาน โดยเปิดโปรแกรม Raspberry Pi Imager เลือก “CHOOSE OS” เป็น “Erase”



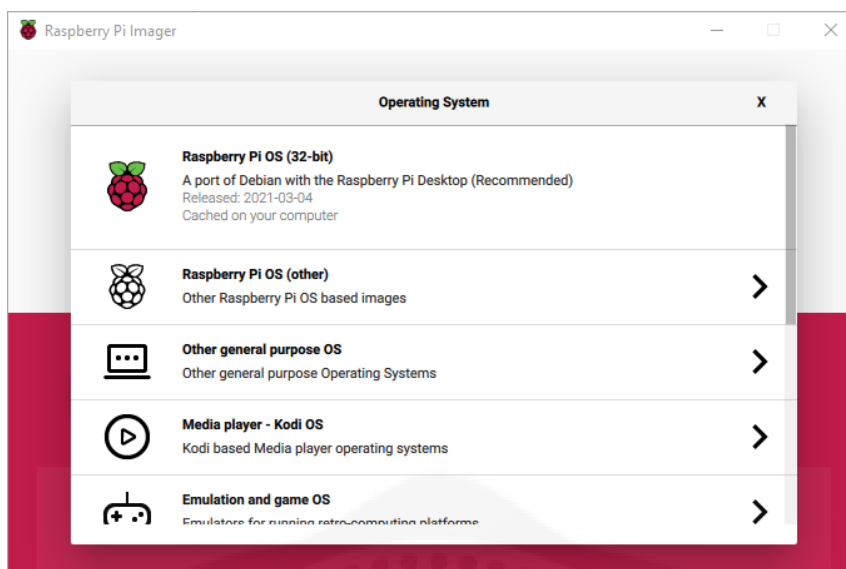
ภาพที่ ก - 1.3 ตัวเลือกสำหรับการ format

1.4 “CHOOSE SD CARD” เลือก sd card ที่เราต้องการ format จากนั้นกด “WRITE” รอจนเสร็จสิ้น



ภาพที่ ก - 1.4 format เสร็จสิ้น

1.5 กลับไปที่ “CHOOSE OS” เลือกติดตั้ง Raspberry Pi OS (32 -bit)



ภาพที่ ก - 1.5 ตัวเลือกสำหรับการติดตั้ง Raspberry Pi OS

1.6 “CHOOSE SD CARD” เลือก sd card ที่เราต้องการติดตั้งจากนั้นกด “WRITE” รอจนเสร็จสิ้น

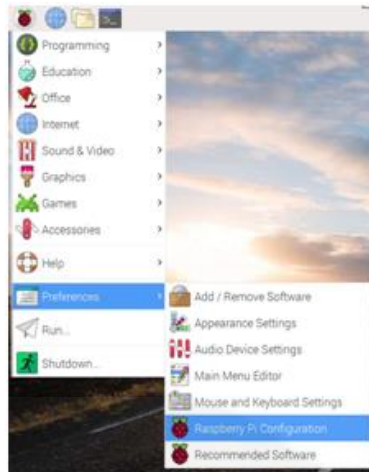


ภาพที่ ก - 1.6 เสร็จสิ้นการติดตั้ง Raspberry Pi OS

1.7 SD Card พร้อมสำหรับการใช้งานบน Raspberry Pi 4

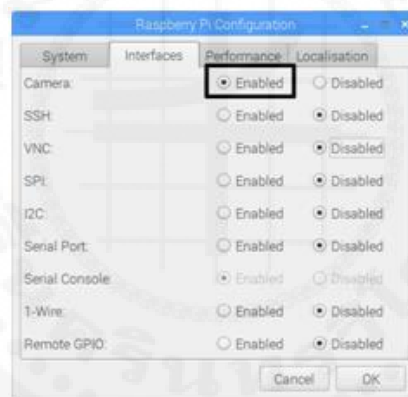
2. การเปิดใช้งาน Raspberry Pi Camera

2.1 ที่ Raspberry Pi 4 ไปที่ Preferences > Raspberry Pi Configuration



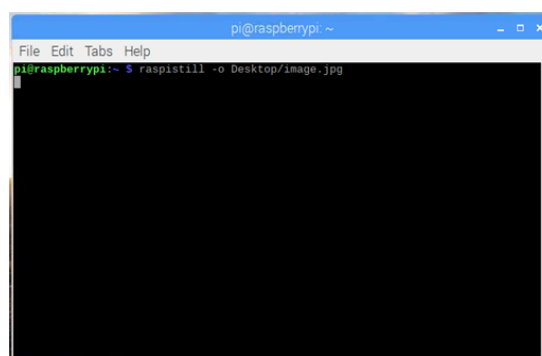
ภาพที่ ก - 2.1 เข้า Raspberry Pi Configuration

2.2 ที่แท็บ Interfaces > Camera เลือก Enabled จากนั้น กด OK



ภาพที่ ก - 2.2 ทำการ Enabled Camera

2.3 ทดสอบถ่ายภาพโดยเปิด Terminal พิมพ์คำสั่ง `$ raspistill -o Desktop/image.jpg`



ภาพที่ ก - 2.3 ทดสอบถ่ายภาพ

ภาคผนวก ข

ชื่อไฟล์ : TrainModel_ResNet50.ipynb

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดการทำงานในการสร้าง model โดยให้เกิดการเรียนรู้จากภาพแล้วสามารถนำมาจำแนกได้ โค้ดจะเริ่มตั้งแต่การนำเข้าข้อมูลภาพ การแบ่งไฟล์ การเรียนรู้ของภาพ การประเมินประสิทธิภาพ และการส่งออกไฟล์ในรูปแบบที่สามารถไปใช้งานได้ในการทำนายเพื่อไม่ต้องเสียเวลาในการเรียนรู้หลายครั้ง

```
#การนำเข้า Module อื่น ๆ เพื่อนำมาเรียกใช้งานในโปรแกรม
%reload_ext autoreload
%autoreload 2
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
from fastai.vision import *
from fastai.metrics import error_rate
from pathlib import Path
from glob2 import glob
from sklearn.metrics import confusion_matrix , accuracy_score,precision_score,recall_score
import pandas as pd
import numpy as np
import os
import zipfile as zf
import shutil
import re
import seaborn as sns

files = zf.ZipFile("data-set.zip",'r') # แดกไฟล์ .zip
files.extractall()
files.close()

def split_indices (folder,seed1,seed2): #การแบ่ง folder Train, Test, Validation
```

```

n = len(os.listdir(folder))
full_set = list(range(1,n+1))
random.seed(seed1)
train = random.sample(list(range(1,n+1)),int(.8*n))
remain = list(set(full_set)-set(train))
random.seed(seed2)
valid = random.sample(remain,int(.5*len(remain)))
test = list(set(remain)-set(valid))
return(train,valid,test)

def get_names (waste_type,indices):
    file_names = [waste_type+" (" +str(i)+").jpg" for i in indices]
    return(file_names)

def move_files (source_files,destination_folder):
    for file in source_files:
        shutil.move(file,destination_folder)

subsets = ['train','valid']
waste_types = ['glass','metal','plastic','trash']
for subset in subsets: # จัดการ directory
    for waste_type in waste_types:
        folder = os.path.join('data',subset,waste_type)
        if not os.path.exists(folder):
            os.makedirs(folder)

if not os.path.exists(os.path.join('data','test')):
    os.makedirs(os.path.join('data','test'))

for waste_type in waste_types:
    source_folder = os.path.join('data-set',waste_type)
    train_ind, valid_ind, test_ind = split_indices(source_folder,1,1)

```

```

train_names = get_names(waste_type,train_ind)
train_source_files = [os.path.join(source_folder,name) for name in train_names]
train_dest = "data/train/"+waste_type
move_files(train_source_files,train_dest)

valid_names = get_names(waste_type,valid_ind)
valid_source_files = [os.path.join(source_folder,name) for name in valid_names]
valid_dest = "data/valid/"+waste_type
move_files(valid_source_files,valid_dest)

test_names = get_names(waste_type,test_ind)
test_source_files = [os.path.join(source_folder,name) for name in test_names]
move_files(test_source_files,"data/test")

path = Path(os.getcwd())/ "data" # ย้าย path ไปที่ เราจะเรียกใช้

tfms = get_transforms(do_flip=True,flip_vert=True) # เพิ่มข้อมูลโดยการหมุนภาพแนวตั้งและนอน
data = ImageDataBunch.from_folder(path,test="test",ds_tfms=tfms,bs=16)

learn = create_cnn(data,models.resnet50,metrics=error_rate) # เลือกใช้ model "ResNet50"

learn.lr_find() # การหา learning rate
learn.recorder.plot(suggestion=True)

learn.fit_one_cycle(40,max_lr=7.41E-04) # การ Training 40 รอบการทำงาน

interp = ClassificationInterpretation.from_learner(learn) # การแสดงภาพที่ไม่ถูกต้องที่สุด
losses,idxs = interp.top_losses()
interp.plot_top_losses(9, figsize=(15,11))
interp.plot_top_losses
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
interp.most_confused(min_val=2)

```

```

preds = learn.get_preds(ds_type=DatasetType.Test) # ทำนายกับชุด Test
max_idx = np.asarray(np.argmax(preds[0],axis=1))
yhat = []
for max_idx in max_idx:
    yhat.append(data.classes[max_idx])

y = [] # ดึงค่าจริงของภาพมาเทียบความถูกต้อง
for label_path in data.test_ds.items:
    y.append(str(label_path))
pattern = re.compile("[a-z]+\s{1}([0-9]+)")
for i in range(len(y)):
    y[i] = pattern.search(y[i]).group(1)

cm = confusion_matrix(y,yhat) # ดูประสิทธิภาพ model ในรูป confusion matrix
df_cm = pd.DataFrame(cm,waste_types,waste_types)
plt.figure(figsize=(11,8))

ax = sns.heatmap(df_cm,annot=True,fmt="g",cmap="YlGnBu")
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)

plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

correct = 0
for r in range(len(cm)):
    for c in range(len(cm)):
        if (r==c):
            correct += cm[r,c]
accuracy = correct/sum(sum(cm))

```

```
accuracy # แสดงเปอร์เซ็นต์ความถูกต้อง
```

```
learn.export('TrashyNet.pkl') # Export model เพื่อนำไปใช้ต่อ
```



ชื่อไฟล์ : Automatrashsimple.ino

ภาษาที่ใช้ : C

คำอธิบาย : โค้ดการทำงานของถังขยะในส่วนที่ควบคุมมอเตอร์หลังจากที่มีการประมวลผลแล้วว่าขยะเป็นประเภทอะไร โดยโค้ดนี้จะทำงานบน Arduino ซึ่งจะทำงานควบคู่กับการใช้ Pin ที่มีการกำหนดค่าแรงดันไฟใน Raspberry Pi ให้ตรงกัน จึงสามารถควบคุมการทำงานของถังขยะให้ทั้งลงช่องที่ต้องการได้ ประกอบไปด้วยการเซตค่า การเคลื่อนที่ต่าง ๆ ของตัวกล่องรับขยะ และตัวดันขยะ

```
// กำหนดการเชื่อมต่อมอเตอร์ตามขั้นตอนการทำงาน
// ตั้งค่าการเคลื่อนที่ของตัวกล่องรับขยะ
#define dirPin1 2
#define stepPin1 3

// ตั้งค่าการเคลื่อนที่ของตัวดันขยะ
#define dirPin2 4
#define stepPin2 5

// การปรับเปลี่ยนระยะการเคลื่อนที่ของกล่องรับถัง ให้ไปยังถังนั้น ๆ
#define stepsPerRevolution1 200 ///----- Step distance Trash A (Trash)
#define stepsPerRevolution2 5000 ///----- Step distance Trash B (Plastic)
#define stepsPerRevolution3 10700 ///----- Step distance Trash C (Metal)
#define stepsPerRevolution4 15870 ///----- Step distance Trash D (Glass)

// การปรับระยะตัวดันขยะ
#define stepsArm 3100

// การปรับระยะให้อยู่ในตำแหน่งเริ่มต้น
#define stepsPerRevolutionHome 1

// กำหนดค่านำเข้า
const int Ain1 = 9;
const int Ain2 = 10;
```

```
const int Ain3 = 11;
const int Ain4 = 12;

// ค่าคงที่ที่ไม่มีมีการเปลี่ยนแปลง โดยค่านี้จะเป็นการกำหนด Pin
const int MoveHome = 8; // Pin ของตัวกล่องรับขยะ
const int ArmHome = 6; // Pin ของตัวถังขยะ

// ตัวแปรที่จะมีการเปลี่ยนแปลง
int MoveHomeState = 0; // ตัวแปรเพื่อดูสถานะของปุ่มเช็คการกักจุดเริ่มต้นของตัวกล่องรับขยะ
int ArmHomeState = 0; // ตัวแปรเพื่อดูสถานะของปุ่มเช็คการกักจุดเริ่มต้นของตัวถังขยะ

// กำหนดตัวแปรกำหนดสถานะการนำเข้า
int StateA1 = 0;
int StateA2 = 0;
int StateA3 = 0;
int StateA4 = 0;

void setup () {
  // ประกาศ Pin เป็นการนำเข้า
  pinMode (Ain1, INPUT);
  pinMode (Ain2, INPUT);
  pinMode (Ain3, INPUT);
  pinMode (Ain4, INPUT);

  // ประกาศ Pin เป็นการส่งออก
  pinMode (stepPin1, OUTPUT);
  pinMode (dirPin1, OUTPUT);

  pinMode (stepPin2, OUTPUT);
  pinMode (dirPin2, OUTPUT);

  pinMode (MoveHome, INPUT);
```

```

pinMode (ArmHome, INPUT);
Serial.begin(9600);
Serial.println("Automatrash system V1.0");
delay (200);

// เซตให้ทุกครั้งก่อนการทำงานอยู่ที่ตำแหน่งเริ่มต้น เพื่อป้องกันกรณีไฟดับขณะใช้งาน และ ค้าง
movetrayhome ();
ArmtoHome ();
}

// เงื่อนไขการเคลื่อนที่ไปยังถึงที่ต้องการ โดยจะควบคุมแรงดันไฟของค่า 4 Pin โดยไปยังถึงที่ Pin มีค่าเป็น LOW
void loop () { // อ่านค่าจาก Raspberry Pi
    StateA1 = digitalRead (Ain1); // ให้ตัวแปรสามารถอ่านค่าสถานะแรงดันไฟได้ โดยมีค่าเป็น LOW หรือ HIGH
    delay (10);
    StateA2 = digitalRead (Ain2);
    delay (10);
    StateA3 = digitalRead (Ain3);
    delay (10);
    StateA4 = digitalRead (Ain4);
    delay (10);

    // กำหนดเงื่อนไข ที่ตัวแปร StateA1 ตรวจสอบสถานะได้เป็น LOW
    if (StateA1 == LOW && StateA2 == HIGH && StateA3 == HIGH && StateA4 == HIGH) {
        Serial.println("Trash-1");
        delay (500);
        movetrayforward1();
        delay (500);
        Serial.println("Arm-push");
        movearm ();
        delay (500);
        ArmtoHome ();
        Serial.println("Back-home");
    }
}

```

```
movetrayhome ();
delay (500);
Serial.println("Finish");
delay (1000);
}

// กำหนดเงื่อนไข ที่ตัวแปร StateA2 ตรวจสอบสถานะได้เป็น LOW
else if (StateA1 == HIGH && StateA2 == LOW && StateA3 == HIGH && StateA4 == HIGH) {
Serial.println("Trash-2");
delay (500);
movetrayforward2();
delay (500);
Serial.println("Arm-push");
movearm ();
delay (500);
ArmtoHome ();
Serial.println("Back-home");
movetrayhome ();
delay (500);
Serial.println("Finish");
delay (1000);
}

// กำหนดเงื่อนไข ที่ตัวแปร StateA3 ตรวจสอบสถานะได้เป็น LOW
else if (StateA1 == HIGH && StateA2 == HIGH && StateA3 == LOW && StateA4 == HIGH) {
Serial.println("Trash-3");
delay (500);
movetrayforward3();
delay (500);
Serial.println("Arm-push");
movearm ();
delay (500);
```

```
ArmtoHome ();
Serial.println("Back-home");
movetrayhome ();
delay (500);
Serial.println("Finish");
delay (1000);
}

// กำหนดเงื่อนไข ที่ตัวแปร StateA4 ตรวจสอบสถานะได้เป็น LOW
else if (StateA1 == HIGH && StateA2 == HIGH && StateA3 == HIGH && StateA4 == LOW) {
Serial.println("Trash-4");
delay (500);
movetrayforward4();
delay (500);
Serial.println("Arm-push");
movearm ();
delay (500);
ArmtoHome ();
Serial.println("Back-home");
movetrayhome ();
delay (500);
Serial.println("Finish");
delay (1000);
}

else {
Serial.println("Wait-command");
}
delay (1000);
}

// การเคลื่อนที่ของตัวกล่องรับไปยังช่องถังขยะที่ 1
void movetrayforward1() {
```

```
digitalWrite (dirPin1, LOW); // กำหนดการทำงานของ Pin ในสถานะ LOW หรือ HIGH
for (int i = 0; i < stepsPerRevolution1; i++) {
    digitalWrite (stepPin1, HIGH);
    delayMicroseconds (500);
    digitalWrite (stepPin1, LOW);
    delayMicroseconds (500);
}
}

// การเคลื่อนที่ของตัวกลองรับไปยังช่องถึงขยยะที่ 2
void movetrayforward2() {
    digitalWrite (dirPin1, LOW);
    for (int i = 0; i < stepsPerRevolution2; i++) {
        digitalWrite (stepPin1, HIGH);
        delayMicroseconds (500);
        digitalWrite (stepPin1, LOW);
        delayMicroseconds (500);
    }
}

// การเคลื่อนที่ของตัวกลองรับไปยังช่องถึงขยยะที่ 3
void movetrayforward3() {
    digitalWrite (dirPin1, LOW);
    for (int i = 0; i < stepsPerRevolution3; i++) {
        digitalWrite (stepPin1, HIGH);
        delayMicroseconds (500);
        digitalWrite (stepPin1, LOW);
        delayMicroseconds (500);
    }
}

// การเคลื่อนที่ของตัวกลองรับไปยังช่องถึงขยยะที่ 4
```

```
void movetrayforward4() {  
    digitalWrite (dirPin1, LOW);  
    for (int i = 0; i < stepsPerRevolution4; i++) {  
        digitalWrite (stepPin1, HIGH);  
        delayMicroseconds (500);  
        digitalWrite (stepPin1, LOW);  
        delayMicroseconds (500);  
    }  
}
```

// การเคลื่อนที่ของตัวกล่องรับไปให้กลับมายังตำแหน่งเดิม

```
void movetrayhome () {  
    do {MoveHomeState = digitalRead (MoveHome);  
        digitalWrite (dirPin1, HIGH);  
        for (int i = 0; i < stepsPerRevolutionHome; i++) {  
            digitalWrite (stepPin1, HIGH);  
            delayMicroseconds (500);  
            digitalWrite (stepPin1, LOW);  
            delayMicroseconds (500);  
        }  
    }  
    while (MoveHomeState == LOW);  
}
```

// การเคลื่อนที่ของตัวคันชโยให้มีการดัน

```
void movearm () {  
    digitalWrite (dirPin2, LOW);  
    for (int i = 0; i < stepsArm; i++) {  
        digitalWrite (stepPin2, HIGH);  
        delayMicroseconds (500);  
        digitalWrite (stepPin2, LOW);  
        delayMicroseconds (500);  
    }  
}
```

```
}  
}  
// การเคลื่อนที่ของตัวต้นขยับให้กลับมายังตำแหน่งเดิม  
void ArmtoHome () {  
    do {ArmHomeState = digitalRead (ArmHome);  
        digitalWrite (dirPin2, HIGH);  
        for (int i = 0; i < stepsPerRevolutionHome; i++) {  
            digitalWrite (stepPin2, HIGH);  
            delayMicroseconds (500);  
            digitalWrite (stepPin2, LOW);  
            delayMicroseconds (500);  
        }  
    }  
    while (ArmHomeState == LOW);  
}
```

ชื่อไฟล์ : main.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดการทำงานของถังขยะในส่วนที่ควบคุมการรับภาพถ่ายจากขยะจริงเพื่อนำไปประมวลผลผลลัพธ์ว่าเป็นขยะประเภทไหน และกำหนดแรงดันไฟให้ตรงเพื่อส่งการทำงานต่อให้มอเตอร์ที่ถังขยะตามขยะนั้นๆ ที่จำแนกได้ตามผลลัพธ์ โดยทำตามขั้นตอนการทำงานของโค้ดที่ Arduino ต่อไปโดยโค้ดส่วนนี้เป็นตัวดำเนินการหลักของขั้นตอนการทำงานทั้งหมดตั้งแต่ต้นจนจบการทำงาน

```
import RPi.GPIO as GPIO
import time
import os

from TestModel import *
from Photo import *
import numpy as np
import cv2
from Ultrasonic import *
from Motor import *
GPIO.setmode(GPIO.BCM) #Type Connect Pin
picture for show
img1 = cv2.imread('png/reTrash.png')
img2 = cv2.imread('png/ready.png')
img3 = cv2.imread('png/not ready.png')
img4 = cv2.imread('png/analysis.png')

#PinconnectwithUltrasonicOnThebox
TRIG = 21 # Pin GPIO21
ECHO = 20 # Pin GPIO20

#pin LED
pinled = 26
```

```
pin_ready = 19 #Green
pin_working = 13 #Red

#Important
time.sleep(0.001)
GPIO.setwarnings(False)
GPIO.setup(pinled,GPIO.OUT)
GPIO.setup(pin_ready,GPIO.OUT)
GPIO.setup(pin_working,GPIO.OUT)

testmodel = TestModel()
photo = Photo()
ultrasonic = Ultrasonic()
motor = Motor()
print ("Distance Measurement In Progress")

# Set output trig and input echo
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)
GPIO.setwarnings(False)

try:
    cv2.imshow('reTrash',img1)
    cv2.waitKey(2000)
    cv2.destroyWindow('reTrash')
    cv2.imshow('reTrash',img2)
    cv2.waitKey(1000)
    while True:

        GPIO.output(pin_ready,GPIO.HIGH) #OpenGreen
```

```
GPIO.output(TRIG, False)

print ("Waiting For Sensor To Settle")
time.sleep(0.25)
GPIO.output(TRIG, True)
#time.sleep(0.00001)
time.sleep(0.0001)
GPIO.output(TRIG, False)

while GPIO.input(ECHO)==0: #Don't have Trash
    time_start = time.time()

while GPIO.input(ECHO)==1:#Have Trash
    time_end = time.time()

#Calculate Distance
time_duration = time_end - time_start
distance = time_duration * 17150
distance = round(distance, 2)
print ("Distance:",distance,"cm")

#Sensor detect a object
if distance < 27:
    time_start_all = time.time()
    GPIO.output(pin_ready,GPIO.LOW)
    GPIO.output(pin_working,GPIO.HIGH)
    print("_____")
    print("have object")
    print ("LED on")
    GPIO.output(pinled,GPIO.LOW) #OpenLED
#    time.sleep(20)
```

```
#call def detect from photo class

##Capture
photo.capture()

##Crop
photo.crop()

#Subtract
photo.CheckTrash()

#if(real have)
cv2.destroyWindow('reTrash')
cv2.imshow('reTrash',img4)
cv2.waitKey(1000)

if (photo.CheckTrash() == True):

    from testw import *
    ##call def predict
    predict = testmodel.predict()
    print(predict)

    #3#
    acc = testmodel.acc()
    print (acc)
    if (acc < 0.5):
        pridict1 = weigth.check()
        print((predict)+"-->"+(pridict1))
        predict = pridict1

    #call def move from motor class
    motor.arduino(predict)

    ##call def rename from photo class
    photo.rename(predict)

    ##call def uploadtos3 from photo class
```

```
photo.uploadtos3(predict)

    ##call def measure and def noti from ultrasonic class
    ultrasonic.checkfull(predict)

    #call def backtobasic from motor class
    print ("LED off")

    GPIO.output(pinled,GPIO.HIGH)

    GPIO.output(pin_working,GPIO.LOW)

    time_end_all = time.time()

    print("time all : " + str(time_end_all - time_start_all))

    print("_____")

else:

    os.remove('/home/pi/Desktop/test.jpg')

    print("Remove test.jpg")

    GPIO.output(pin_working,GPIO.LOW)

    print ("LED on")

    GPIO.output(pinled,GPIO.HIGH)

    time_end_all = time.time()

    print("time all : " + str(time_end_all - time_start_all))

continue

cv2.destroyWindow('reTrash')

cv2.imshow('reTrash',img2)

cv2.waitKey(1000)

except KeyboardInterrupt:

    GPIO.output(pin_ready,GPIO.LOW)

    print("Cleaning up!")

    GPIO.cleanup()
```

finally:

```
GPIO.output(pin_ready,GPIO.LOW)
```

```
GPIO.cleanup()
```



ชื่อไฟล์ : Photo.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดฟังก์ชันการทำงานของส่วนที่มีความเกี่ยวข้องกับการใช้ภาพถ่าย ที่มีการเชื่อมกับ Raspberry Pi Camera เป็นกล้องที่ใช้ในการถ่ายภาพอยู่ภายในตัวถัง และ Amazon Simple Storage Service (S3) ในการเก็บข้อมูลภาพขณะที่ย้ายไป โดยจะมีฟังก์ชันย่อยๆ ดังนี้ คำสั่งถ่ายภาพ ตัดขอบภาพ ตรวจสอบวัตถุในภาพ เปลี่ยนชื่อไฟล์ภาพ อัปโหลดไฟล์ภาพ

โดยโค้ดส่วนนี้เป็นตัวดำเนินการย่อย ที่โปรแกรมหลักจะเรียกใช้งาน

#การนำเข้า Module อื่น ๆ เพื่อนำมาเรียกใช้งานในโปรแกรม

```
from picamera import PiCamera
```

```
from time import sleep
```

```
import sys, os, glob, time
```

```
import os
```

```
import datetime
```

```
import RPi.GPIO as GPIO
```

```
from boto. s3. connection import S3Connection
```

```
from boto.s3.key import Key
```

```
import glob
```

```
from PIL import Image, ImageDraw, ImageFilter
```

```
import cv2
```

```
import numpy as np
```

```
from TestModel import *
```

```
testmodel = TestModel ()
```

```
pin = 16
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(pin, GPIO.OUT)
```

```
class Photo:
```

```
    def __init__(self): # การกำหนดเพื่อใช้ในการเข้าถึง S3
```

```

self.directory = '/home/pi/Desktop/'
self.AWS_ACCESS = "AKIA4JSFZIFQHABSYHEQ"
self.AWS_SECRET = "PMFsTOAT5p9a2YyDw4QdfbBq4wJnWEuQ3sGuePv"
self.conn = S3Connection(self.AWS_ACCESS,self.AWS_SECRET)
self.bucket = self.conn.get_bucket('trashy2')

def capture(self): # การถ่ายภาพขยะ
    start = time.time ()
    camera = PiCamera ()
    camera.resolution = (1280,960)
    camera.start_preview()
    sleep(3)
    camera.capture ('/home/pi/Desktop/test.jpg')
    camera.stop_preview()
    camera.close ()

def crop(self): # การตัดขอบภาพ
    im = Image.open('/home/pi/Desktop/test.jpg')
    thumb_width = 711
    thumb_height = 568
    img_width, img_height = im.size
    im_thumb = im.crop (((img_width - thumb_width) // 2, (img_height - thumb_height) // 2,
(img_width + thumb_width) // 2, (img_height + thumb_height) // 2))
    im_thumb.save ('/home/pi/Desktop/test.jpg')

def rename(self,folder): # การเปลี่ยนชื่อไฟล์ภาพ
    Current_Date = datetime.datetime.now ()
    os.rename (r'/home/pi/Desktop/test.jpg',r'/home/pi/Desktop/'+ folder + '-' + str(Current_Date) +
'.jpg')

def percent_cb(self,complete, total):

```

```
sys.stdout.write('.')
sys.stdout.flush()

def getFiles(self, dir):
    self.dir = dir
    return [os.path.basename(x) for x in glob.glob(str(self.dir) + '*.jpg')]

def uploadtos3(self, folder): # อัปโหลดไฟล์ภาพขึ้น S3
    filenames = self.getFiles (self.directory)
    path_s3 = folder + '/'
    for f in filenames:
        print(f)
        k = Key (self.bucket)
        k.key = path_s3 + f
        GPIO.output(pin, GPIO.HIGH)
        k.set_contents_from_filename (self.directory + f, cb=self.percent_cb, num_cb=10)
        GPIO.output(pin, GPIO.LOW)
        os.remove (self.directory + f)

def CheckTrash(self): # ตรวจสอบวัตถุในภาพ โดยแปลงภาพเป็นภาพขาวดำก่อน
    bg = cv2.imread('/home/pi/Desktop/TestSubImg/bg.jpg')
    bg_grey = cv2.cvtColor(bg, cv2.COLOR_RGB2GRAY)

    frame = cv2.imread('/home/pi/Desktop/test.jpg')
    frame_grey = cv2.cvtColor(frame, cv2.COLOR_RGB2GRAY)

    difference = cv2.absdiff(bg_grey, frame_grey)
    _, final = cv2.threshold(difference, 20, 255, cv2.THRESH_BINARY)

    final = cv2.medianBlur(final,5)
    final = cv2.GaussianBlur(final, (3,3),0)
```

```
arr2 = np.asarray (final)
image_size = bg_grey.shape [0] * bg_grey.shape [1]

final2 = arr2 / 255.0
image_detect_size = np.sum(final2)

if (image_detect_size > (0.02*image_size)):
    return True; # มีวัตถุ
else:
    return False; # ไม่มีวัตถุ
```



ชื่อไฟล์ : TestModel.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดฟังก์ชันการทำงานของส่วนในการทำนายผลลัพธ์การแยกประเภทขยะจากภาพถ่ายโดยจะทำการดึงไฟล์ที่มีชื่อว่า **Trashy_ResNet50_Model.pkl** เป็นไฟล์ที่ได้จากการ Export หลังการ Train model เรียบร้อยแล้ว เพื่อให้สามารถนำมาใช้ได้รวดเร็วมากขึ้น

โดยโค้ดส่วนนี้เป็นตัวดำเนินการย่อย ที่โปรแกรมหลักจะเรียกใช้งาน

```
#การนำเข้า Module อื่น ๆ เพื่อนำมาเรียกใช้งานในโปรแกรม
from fastai.vision import *
import time

class TestModel:
    def __init__(self): # การกำหนดที่อยู่ไฟล์ที่จะใช้งาน
        self.path = "/home/pi/Desktop/Trashy"
        self.learn = load_learner(self.path,'Trashy_ResNet50_Model.pkl')

    def predict (self): # การนำภาพเพื่อมาทำนายผลลัพธ์แล้วนำค่าเก็บในตัวแปร predict
        start = time.time ()
        data = open_image ('/home/pi/Desktop/test.jpg')
        data = data.resize ((3,384,512))
        predicted_class, predicted_index, outputs = self.learn.predict (data)
        predict = str (predicted_class)
        end = time.time()

        print ("time to predict : " + str(end - start))
        return predict

    def acc(self):
        data = open_image('/home/pi/Desktop/test.jpg')
        data = data.resize((3,384,512))
        predicted_class, predicted_index, outputs = self.learn.predict(data)
```

```
# print (outputs)
# print (max(outputs))
# print("-----")
list1 = []
for num in outputs:
    i = "{:.10f}".format(num)
    list1.append(float(i))

# print (type(outputs))
a = sorted(list1,key = float)
# print (a)
return (a[3]-a[2])

#def accuracy(self):
# data = open_image('/home/pi/Desktop/test.jpg')
#data = data.resize((3,384,512))
# predicted_class, predicted_index, outputs = self.learn.predict(data)

#print (outputs)

# test All time ~ 25 sec
```

ชื่อไฟล์ : Motor.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดฟังก์ชันการทำงานของส่วนที่จะทำการเคลื่อนกล่องรับขยะไปยังตำแหน่งของถังขยะที่ทำนายผลได้ มีการตั้งเงื่อนไขในกรณีผลลัพธ์แบบต่างๆ โดยจะตั้งค่าสถานะแรงดันไฟที่ LOW เพื่อให้สามารถสื่อสารกันได้ระหว่างโค้ดนี้กับโค้ดใน Arduino ที่ได้มีการตั้งค่าไว้แล้ว ให้มอเตอร์ทำงานตามต้องการ โดยโค้ดส่วนนี้เป็นตัวดำเนินการย่อย ที่โปรแกรมหลักจะเรียกใช้งาน

```
import RPi.GPIO as GPIO
import time
import numpy as np
import cv2
picture for show

img1 = cv2.imread('png/t1.png')
img2 = cv2.imread('png/t01.png')
img3 = cv2.imread('png/t2.png')
img4 = cv2.imread('png/t02.png')
img5 = cv2.imread('png/t3.png')
img6 = cv2.imread('png/t03.png')
img7 = cv2.imread('png/t4.png')
img8 = cv2.imread('png/t04.png')

GPIO.setmode(GPIO.BCM)

from Ultrasonic import *

ultrasonic = Ultrasonic()

GPIO.setwarnings(False)
```

```
pintrash = 24
pinplast = 23
pinmetal = 17
pinglass = 27

#one way
time_ch1 = 6   #trash go
time_ch2 = 10  #plastic go
time_ch3 = 16  #metal go
time_ch4 = 21  #glass go

class Motor():
    cv2.destroyAllWindows()

    def arduino(self,typebin):

        if (typebin == 'trash') :
            print ("Motor - Hello trash IMTRASH")
            cv2.imshow('trash',img1)
            cv2.waitKey(3000)
            cv2.destroyWindow('trash')
            cv2.imshow('trash',img2)
            cv2.waitKey(1000)
            GPIO.setup(pinplast,GPIO.OUT)
            GPIO.setup(pintrash,GPIO.OUT)
            GPIO.setup(pinmetal,GPIO.OUT)
            GPIO.setup(pinglass,GPIO.OUT)
            GPIO.output(pinplast,GPIO.HIGH)
            GPIO.output(pintrash,GPIO.LOW)
            GPIO.output(pinmetal,GPIO.HIGH)
            GPIO.output(pinglass,GPIO.HIGH)
```

```
time.sleep(time_ch1) #go
cv2.destroyWindow('trash')
ultrasonic.checkfull(typebin)
time.sleep(time_ch1-3) #back
GPIO.output(pintrash,GPIO.HIGH)

if (typebin == 'plastic') :
    print ("Motor - Hello plastic IMTRASH")
    cv2.imshow('plastic',img7)
    cv2.waitKey(3000)
    cv2.destroyWindow('plastic')
    cv2.imshow('plastic',img8)
    cv2.waitKey(1000)
    GPIO.setup(pinplast,GPIO.OUT)
    GPIO.setup(pintrash,GPIO.OUT)
    GPIO.setup(pinmetal,GPIO.OUT)
    GPIO.setup(pinglass,GPIO.OUT)
    GPIO.output(pinplast,GPIO.LOW)
    GPIO.output(pintrash,GPIO.HIGH)
    GPIO.output(pinmetal,GPIO.HIGH)
    GPIO.output(pinglass,GPIO.HIGH)
    time.sleep(time_ch2)
    cv2.destroyWindow('plastic')
    ultrasonic.checkfull(typebin)
    time.sleep(time_ch2-3)
    GPIO.output(pinplast,GPIO.HIGH)

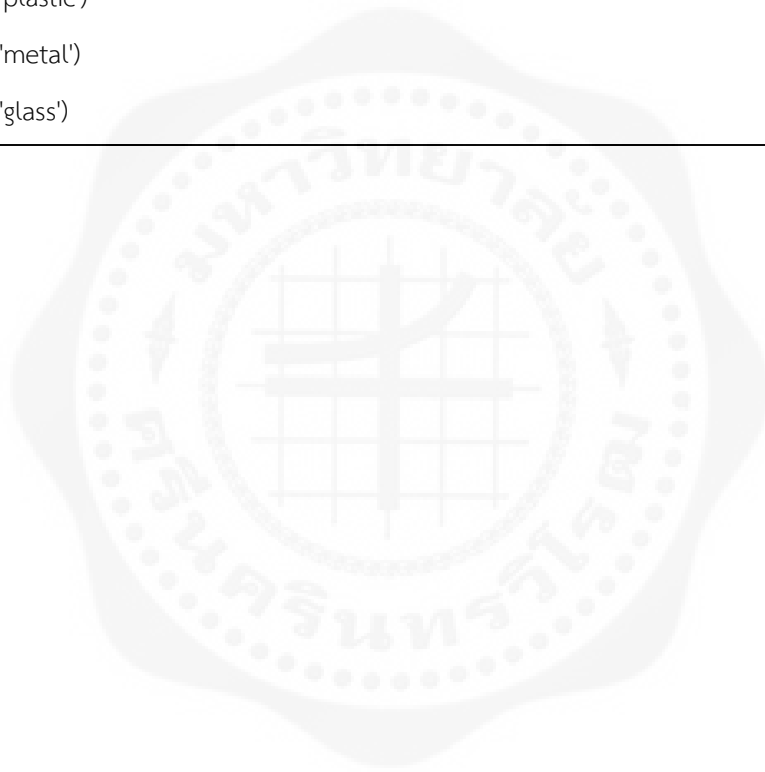
if (typebin == 'metal') :
    print ("Motor - Hello metal IMTRASH")
    cv2.imshow('metal',img5)
    cv2.waitKey(3000)
```

```
cv2.destroyAllWindows('metal')
cv2.imshow('metal',img6)
cv2.waitKey(1000)
GPIO.setup(pinplast,GPIO.OUT)
GPIO.setup(pintrash,GPIO.OUT)
GPIO.setup(pinmetal,GPIO.OUT)
GPIO.setup(pinglass,GPIO.OUT)
GPIO.output(pinplast,GPIO.HIGH)
GPIO.output(pintrash,GPIO.HIGH)
GPIO.output(pinmetal,GPIO.LOW)
GPIO.output(pinglass,GPIO.HIGH)
time.sleep(time_ch3)
cv2.destroyAllWindows('metal')
ultrasonic.checkfull(typebin)
time.sleep(time_ch3-3)
GPIO.output(pinmetal,GPIO.HIGH)

if (typebin == 'glass') :
    print ("Motor - Hello glass IMTRASH")
    cv2.imshow('glass',img3)
    cv2.waitKey(3000)
    cv2.destroyAllWindows('glass')
    cv2.imshow('glass',img4)
    cv2.waitKey(1000)
    GPIO.setup(pinplast,GPIO.OUT)
    GPIO.setup(pintrash,GPIO.OUT)
    GPIO.setup(pinmetal,GPIO.OUT)
    GPIO.setup(pinglass,GPIO.OUT)
    GPIO.output(pinplast,GPIO.HIGH)
    GPIO.output(pintrash,GPIO.HIGH)
    GPIO.output(pinmetal,GPIO.HIGH)
    GPIO.output(pinglass,GPIO.LOW)
```

```
time.sleep(time_ch4)
cv2.destroyAllWindows('glass')
ultrasonic.checkfull(typebin)
time.sleep(time_ch4-3)
GPIO.output(pinglass,GPIO.HIGH)

if __name__ == '__main__':
    mm = Motor()
#    mm.arduino('trash')
#    mm.arduino('plastic')
#    mm.arduino('metal')
#    mm.arduino('glass')
```



ชื่อไฟล์ : Ultrasonic.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดฟังก์ชันการทำงานของส่วนที่จะตรวจสอบว่าปริมาณขยะในถังเต็มหรือไม่ โดย Ultrasonic Sensor เช่นเดียวกับการตรวจสอบว่ามีขยะเข้ามาในกล่องรับหรือยัง แต่จะใช้ในกรณีที่ถังขยะใกล้เต็ม หรือเต็ม จะมีการแจ้งเตือนผ่านทาง Line Notify เพื่อแจ้งให้เจ้าหน้าที่มาทำการเก็บขยะ

โดยโค้ดส่วนนี้เป็นตัวดำเนินการย่อย ที่โปรแกรมหลักจะเรียกใช้งาน

```
#การนำเข้า Module อื่น ๆเพื่อนำมาเรียกใช้งานในโปรแกรม
import RPi.GPIO as GPIO
import time
import requests

GPIO.setmode(GPIO.BCM)
TRIG = 8
ECHO = 7
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)
global msg

class Ultrasonic:
    def checkfull (self,typebin): # การกำหนดเพื่อใช้ในการเข้าถึง Line Notify และ เงื่อนไขถังขยะใกล้เต็ม/เต็ม
        url = 'https://notify-api.line.me/api/notify'
        token = 'RuVYrUg3xGwxSgeesncMkKbRTUeJ5JCwyaNwdy5i5QL'
        headers = {'content-type':'application/x-www-form-urlencoded','Authorization': 'Bearer '+token}
        GPIO.output(TRIG, False)
        time.sleep (0.01)
        GPIO.output(TRIG, True)
        time.sleep (0.0001)
        GPIO.output(TRIG, False)

        while GPIO.input(ECHO)==0:
```

```
time_start = time.time()
while GPIO.input(ECHO)==1:
    time_end = time.time()

time_duration = time_end - time_start
distance = time_duration * 17150
distance = round (distance, 2)
    if (distance < 5): # เจ็อนไซเมื่อถึงขยะเต็ม
        msg = 'The ' + typebin + ' bin is full. And distance : ' + str (distance) + ' cm'
        r = requests.post(url, headers=headers, data = {'message':msg})

elif (distance < 10): # เจ็อนไซเมื่อถึงขยะใกล้เต็ม
    msg ='The ' + typebin + ' bin is almost full. And distance : ' + str (distance) + ' cm'
    r = requests.post (url, headers=headers, data = {'message':msg})
else:
    pass
```

ชื่อไฟล์ : testw.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดฟังก์ชันการทำงานของส่วนที่จะตรวจสอบน้ำหนักของขยะ ซึ่งจะมีการกำหนดช่วงของน้ำหนักแต่ละประเภทไว้ ซึ่งจะเป็นตัวช่วยเพิ่มความแม่นยำในการทำนายขยะแต่ละชิ้นให้มีประสิทธิภาพมากขึ้น โดยโค้ดส่วนนี้เป็นตัวดำเนินการย่อย ที่โปรแกรมหลักจะเรียกใช้งาน

```

from i2c import *
import time

i2c = i2c()
w = float(data)
print(w)
time.sleep(3)
class weighth :
    def __init__(self):
        self.path = "/home/pi/Desktop/Trashy"
        self.learn = load_learner(self.path,"Trashy_ResNet50_Model.pkl")

    def check():
        if w >= 60.0 :
            # glass น้ำหนักมากกว่า 60.0
            print ("glass")
            return glass
        elif w <= 31.0 and w >= 13.0 : # plastic น้ำหนักอยู่ระหว่าง 13.0 ถึง 31.0
            print ("plastic")
            return plastic
        elif w <= 10.0 and w >= 0.1 : # metel น้ำหนักอยู่ระหว่าง 0.1 ถึง 10.0
            print ("metel")
            return metel
        else :
            # trash อื่นๆ
            print ("trash")
            return trash

```

ชื่อไฟล์ : I2C.py

ภาษาที่ใช้ : Python

คำอธิบาย : โค้ดฟังก์ชันการทำงานในการส่งข้อมูลจาก Arduino uno ไปยัง Raspberry Pi ผ่าน I²C

```
#!/usr/bin/python
import smbus
import time

bus = smbus.SMBus(1)
address = 0x04
check = chr(bus.read_byte(address))
#print(check)
data = ""
if check!="0":
    for i in range(int(check)):
        data += chr(bus.read_byte(address))

    #print ((data)+" g")
else:
    print("no")
#time.sleep(9)
```

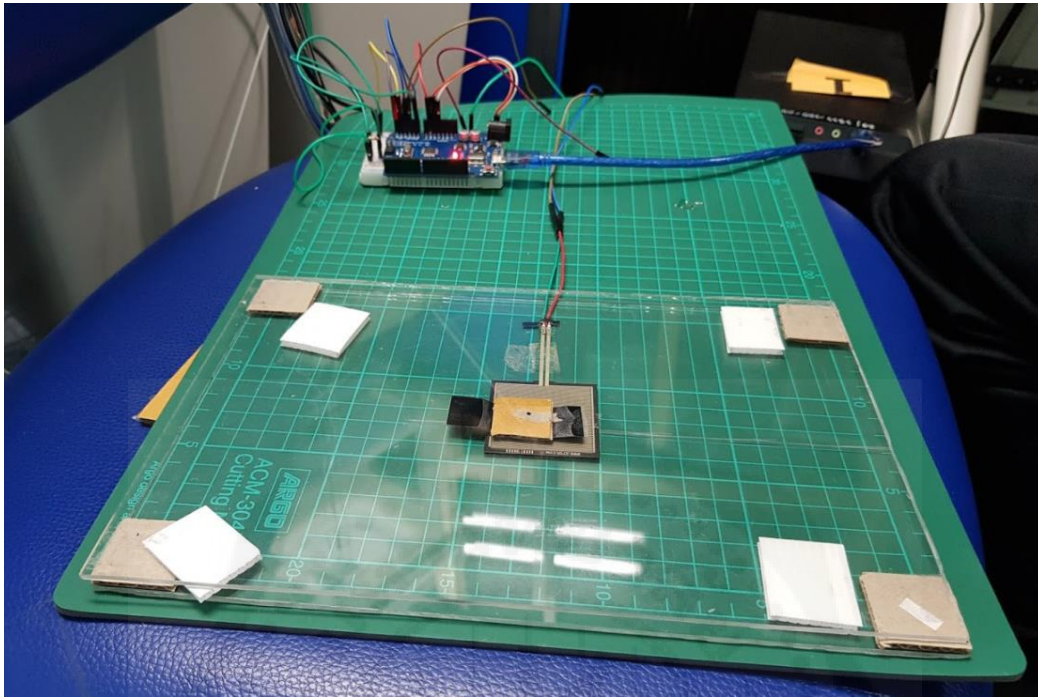
ภาคผนวก ค

ตารางข้อมูลน้ำหนักขยะแต่ละประเภท (พลาสติก, แก้ว, โลหะ)

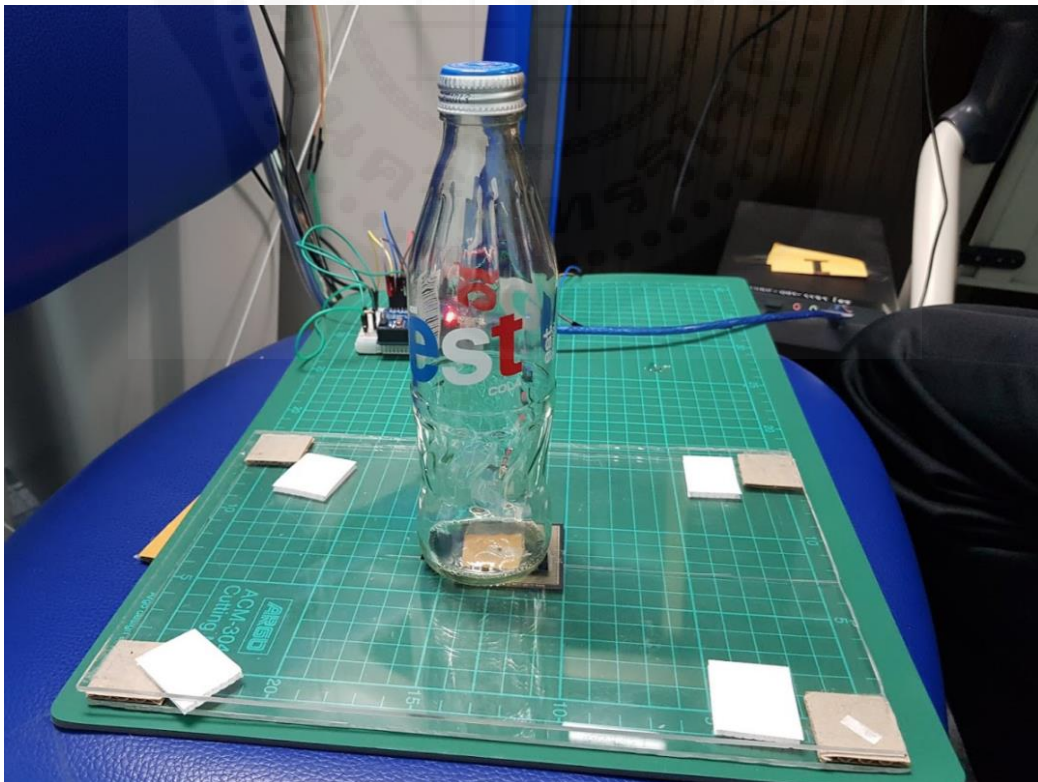
ขวดพลาสติก		ขวดแก้ว		กระป๋องโลหะ	
ชื่อ	น้ำหนัก (กรัม)	ชื่อ	น้ำหนัก (กรัม)	ชื่อ	น้ำหนัก (กรัม)
น้ำดื่ม select7 ขนาด 600 มล.	17	เอส โคล่า ขนาด 250 ซม3	190	คาลพิสแลคโตะ 245 มล.	11
น้ำดื่ม น้ำทิพย์ ขนาด 550 มล.	14	Hi vitamin C 200% ขนาด150 มล.	127	สปอนเซอร์ 325 มล.	13
น้ำดื่ม nestle ขนาด 600 มล.	15	แบรนต์ ซุปไก่สกัด ขนาด 42 มล.	61	Qizz น้ำลิ้นจี่ 230 มล.	11
น้ำดื่ม สิงห์ ขนาด 600 มล.	16	สปอนเซอร์ ขนาด 250 มล.	153	เป๊ปซี่ แมกซ์เทสต์ 245 มล.	11
น้ำดื่ม ออรั่า ขนาด 500 มล.	17	คาราบาวแดง ขนาด 150 มล.	138	เบียร์ ลีโอ ขนาด 320 ซม3	14
น้ำดื่ม vit a day ขนาด 480 มล.	28	M150 ขนาด150 มล.	141		
น้ำดื่ม man some ขนาด 480 มล.	31	มินิทเมต สแปลช ขนาด 250 มล.	183		
น้ำส้ม มินิทเมต พลัส 290 มล.	28				
เป๊ปซี่ แมกซ์ ขนาด 550 มล.	28				
ฟูจิชา กรีนที ขนาด 500 มล.	34				
โค้ก ไม่มีน้ำตาล ขนาด 500 มล.	27				
อิซิดัน กรีนที ขนาด 420 มล.	29				
นม Dutch Mill ขนาด 400 มล.	30				
Bifesta Cleansing 90 มล.	26				
ครีมอาบน้ำ Shokubutsu 100 มล.	20				
paracap paracetamol 100 เม็ด	18				
โออิชิ อุ่นเคียวไฮ 350 กรัม	30				
กาโตะ 320 กรัม	20				
น้ำดื่ม มินาเร่ ขนาด 0.75 ลิตร	24				
น้ำดื่ม tesco ขนาด 330 มล.	13				
ค่าเฉลี่ยของขวดพลาสติก (กรัม)		ค่าเฉลี่ยของขวดแก้ว (กรัม)		ค่าเฉลี่ยของกระป๋องโลหะ (กรัม)	
23.25		141.85		12	

ภาคผนวก ง

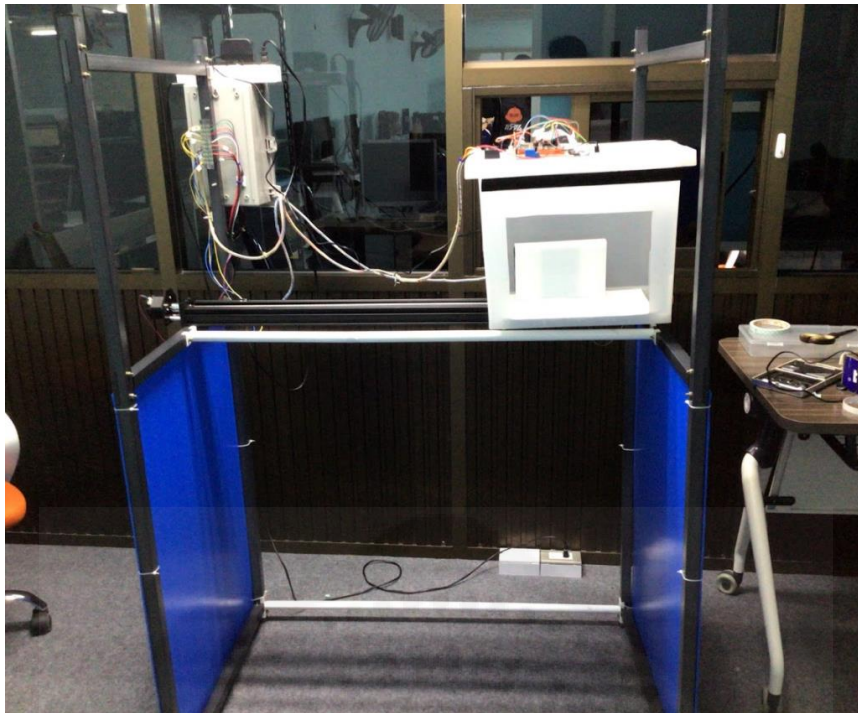
ภาพเกี่ยวกับการทำงาน



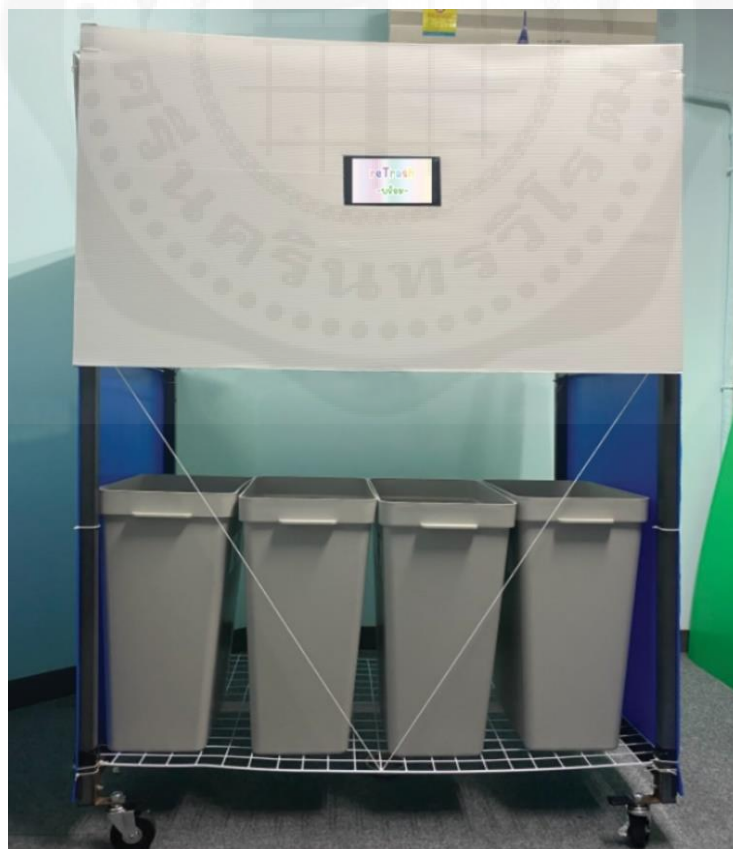
ภาพที่ ง - 1 การทดลองการใช้ Force Sensitive Resistor มาใช้ในการวัดน้ำหนักแทน Loadcell



ภาพที่ ง - 2 การทดลองการใช้ Force Sensitive Resistor มาใช้ในการวัดน้ำหนักแทน Loadcell



ภาพที่ 3 - 3 Trashy ต้นแบบถังขยะที่เรานำมาต่อยอด



ภาพที่ 4 - 4 หลังนำมาต่อยอดเป็น reTrash ในปัจจุบัน