

CP Capstone Final Project
รายงานโครงงานวิศวกรรมคอมพิวเตอร์

เรื่อง
ระบบติดตามและวัดผลมะม่วงทางการเกษตร
Smart mango farm's products detection system

โดย

- | | |
|-------------------------------|---------------------|
| 1. นาย วีรวงศ์ ศรีรัตนสมบุญ | รหัสนิติ 6130639821 |
| 2. นาย เสรี เยี่ยงสกุลไพศาล | รหัสนิติ 6130640321 |
| 3. นาย ชนภัทร กิริติเชาวนากุล | รหัสนิติ 6131006721 |

อาจารย์ที่ปรึกษา
อ.ดร.พรรณราย ศิริเจริญ
ผศ.ดร.ปารเมศ วรเศยานนท์

รายงานนี้เป็นส่วนหนึ่งของวิชา 2110489 โครงงานวิศวกรรมคอมพิวเตอร์พื้นฐาน

บทคัดย่อ

โครงการระบบติดตามและวัดผลมะม่วงทางการเกษตรมีวัตถุประสงค์เพื่อสร้างระบบในการติดตามผลผลิตทางการเกษตรของสวนมะม่วง เพื่อช่วยเหลือเกษตรกรในการวางแผนเก็บเกี่ยวผลผลิตได้อย่างมีประสิทธิภาพ โดยระบบดังกล่าวประกอบด้วย 3 ส่วนหลัก ได้แก่ machine learning model ที่ใช้โครงข่ายประสาทเทียมเชิงลึก เช่น mask R-CNN model สำหรับวิเคราะห์ข้อมูลรูปภาพต้นมะม่วงน้ำดอกไม้ เพื่อบันทึกจำนวนผลไม้บนต้นและคาดการณ์ระดับความสุกของผลไม้ในช่วงเวลาต่างๆ ระบบ IOT ที่ควบคุมผ่าน Arduino และ Sensor ต่างๆ สำหรับใช้ในการเก็บข้อมูลรูปภาพต้นมะม่วงในแต่ละช่วงเวลา พร้อมทั้งส่งผลลัพธ์ไปยัง cloud database มีการใช้ solar cell ในการให้พลังงานตัวบอร์ด และ เก็บไฟใน battery เพื่อใช้ในระยะเวลาสั้น เมื่อทำการประมวลผลผ่าน machine learning เสร็จแล้ว โดยข้อมูลรูปมะม่วงจะถูกส่งจากระบบ IOT ที่ถูกควบคุมโดย Arduino ไปยัง AWS Api Gateway หลังจากนั้นจะเปลี่ยนเป็น JSON เพื่อส่งไปเก็บยัง S3 storage เพื่อประมวลผล Mask R-CNN ผ่าน EC2 instance และนำผลลัพธ์ที่ได้ไปเก็บใน google sheet เพื่อเชื่อมต่อกับ power BI ให้สามารถอ่านค่า-นำเสนอผลลัพธ์ได้ใน schedule รอบรายวัน

Abstract

This project smart mango farm's products detection system using deep learning algorithm, has been created to develop a real time mango tracking system by using Mask R-CNN technique, with the goal to help tracking the number of mangoes and for fruit farmers. The system consists of 3 parts. In the first part, we'll build a system that will be able to take a photo and correct other parameters on the field using an Arduino Kit that can send data via a wireless network. The system also needs to sustain itself by powering from a solar system that we attached it to so that it can operate independently of a conventional power supply for a continuous period. In the second part, we will use a machine learning model to process that has been collected the mango that matches quality our stakeholder's need. Finally, we'll then send the result to AWS API Gateway then change the data to JSON in order to keep it in S3 Storage where it'll be processed via machine learning. The result will be forwarded and kept in Google Sheet which are directly linked to Power BI to show a visualised result of the data. The result from this product should show the number of mangoes with ripeness while making it easier for farmers to keep track of their field that can help the farmers plan an efficient transportation and reduce product cost.

บทที่ 1 : บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ประเทศไทยเป็นประเทศทางเกษตรกรรมที่มีรายได้หลักจากการขายผลผลิตทางการเกษตรโดยเฉพาะผลไม้ส่งออก โดยในปัจจุบันยังมีสวนผลไม้หลายแห่งที่มีวิธีการผลิตแบบล้าสมัย คือ ไม่มีระบบติดตามผลผลิตที่ชัดเจน ส่งผลให้มีผลผลิตเน่าเสียก่อนการเก็บเกี่ยวเป็นจำนวนมาก ทางกลุ่มได้เห็นความสำคัญของประเด็นนี้ จึงต้องการจัดทำโครงการที่จะนำความรู้ทางวิศวกรรมศาสตร์มาช่วยพัฒนาระบบติดตามผลผลิตทางการเกษตร เพื่อช่วยเหลือเกษตรกรในการวางแผนการเก็บเกี่ยว, ลดการสูญเสียในระบบผลผลิต และเพิ่มโอกาสในการเก็บเกี่ยวผลไม้นอกฤดูกาล

1.2 วัตถุประสงค์โครงการ

เพื่อพัฒนาระบบติดตามผลผลิตทางการเกษตร สำหรับช่วยเหลือเกษตรกรในการวางแผนการเก็บเกี่ยวผลผลิตได้อย่างมีประสิทธิภาพ โดยระบบดังกล่าวจะมีคุณสมบัติ ได้แก่ 1) ถูกพัฒนานับอุปกรณ์ภาคสนามชุดทดลอง สำหรับติดตั้งเพื่อเก็บข้อมูลในสวนผลไม้(สวนมะม่วง) 2) สามารถติดตามความเปลี่ยนแปลงของผลผลิตได้ 3) มีระบบแสดงผลที่ชัดเจน เข้าใจง่าย

1.3 ภาพรวมของโปรเจค และผู้รับผิดชอบในแต่ละส่วน

โครงการนี้จะเป็นการพัฒนาพัฒนาระบบติดตามผลผลิตทางการเกษตรทดลองบนสวนมะม่วงฟาร์มอ.เอช โดยระบบจะประกอบด้วย 3 ส่วนหลัก ได้แก่

1. Machine learning model: ใช้ model โครงข่ายระบบประสาทเชิงลึก เช่น mask R-CNN model ในการวิเคราะห์รูปภาพต้นมะม่วงน้ำดอกไม้สีทอง ได้แก่ นับจำนวนผลมะม่วงบนต้น และคาดการณ์ระดับความสุกของผลมะม่วงในช่วงเวลาต่างๆ โดย model จะถูก train แยกบน Google colab จากนั้นจึงจะทำการดาวน์โหลดมาประมวลผลบน EC2 instance ของ cloud application [ผู้รับผิดชอบหลัก: นาย ชนภัทร กิริติเชาวนากุล]
2. ระบบ IOT: เป็นชุดอุปกรณ์ที่ถูกติดตั้งที่สวนเพื่อเก็บรูปมะม่วงน้ำดอกไม้สีทองในแต่ละวัน จากนั้นจึงส่งรูปไปเก็บบน S3 cloud storage ผ่านทางอินเทอร์เน็ต เพื่อนำรูปไปนับจำนวนมะม่วงหรือใช้เป็น dataset สำหรับ train machine learning model ต่อไป[ผู้รับผิดชอบหลัก : นาย เสรี เยี่ยงสกุลไพศาล]
3. Cloud application: เป็น application platform ที่ประกอบด้วย services ย่อยต่างๆ โดยสามารถอธิบายเป็นขั้นตอนได้ดังนี้คือ รับรูปภาพจากระบบ IOT มาเก็บใน S3 storage จากนั้น ใช้ machine learning model ที่รันบน EC2 นำรูปจาก S3 เพื่อไปนับจำนวนมะม่วง แล้วจึงส่งผลลัพธ์ไปเก็บยัง google slide ที่เชื่อมต่อกับ Power BI เพื่อนำไปใช้สรุปผลเป็น dashboard และนำเสนอต่อไป [ผู้รับผิดชอบหลัก: นาย วีรวัศ ศรีรัตนสมบูรณ์]

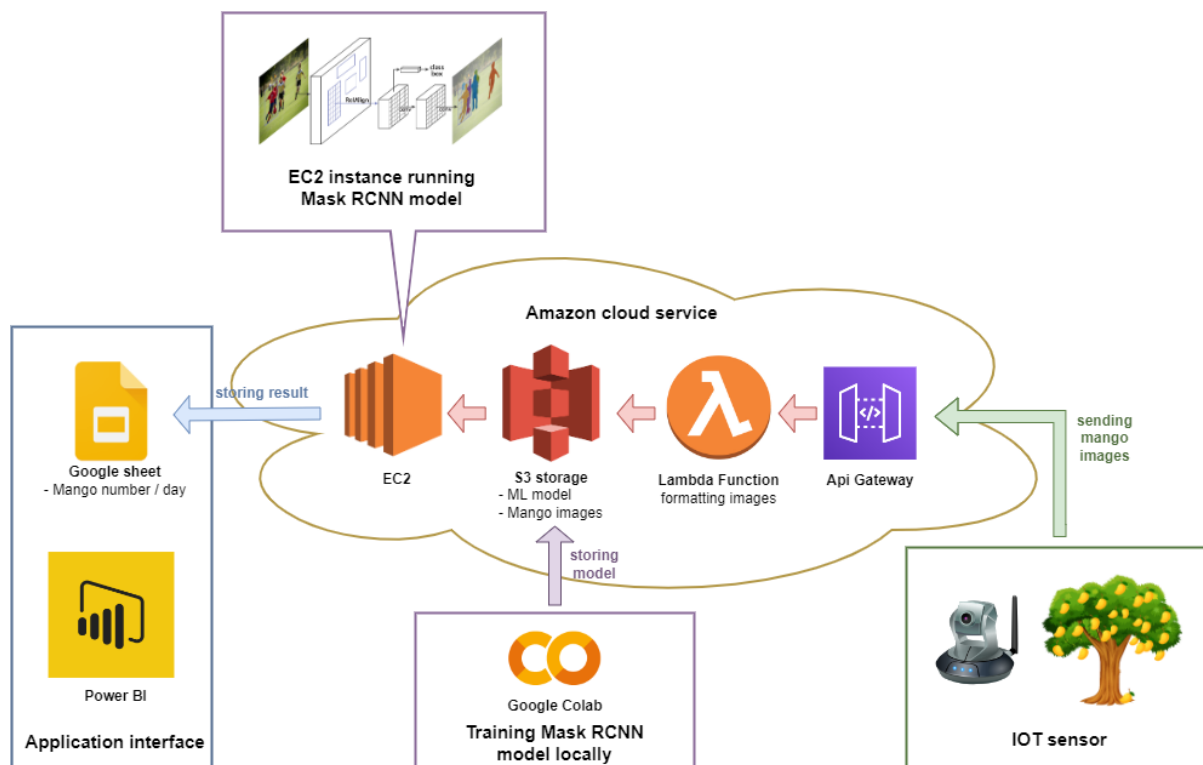


FIGURE 0: Overall Diagram of mango detection system

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากโครงการนี้ คือ สร้างระบบที่สามารถติดตามผลผลิตของสวนมะม่วง ได้แก่ จำนวนผลไม้และระดับความสุกของผลมะม่วงในช่วงเวลาต่างๆ เพื่อให้เกษตรกรสามารถนำข้อมูลมาวางแผนเพื่อเพิ่มประสิทธิภาพการเก็บเกี่ยวและขนส่งผลผลิต เพื่อลดการสูญเสียจากระบบผลผลิต

บทที่ 2 : ทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีเกี่ยวกับ Machine learning

2.1.1 Image segmentation

เป็นเทคนิคในศาสตร์ computer vision ในการวิเคราะห์รูปภาพโดยการจำแนกรูปภาพในระดับ pixel ออกเป็นส่วนต่างๆ(segment) เพื่อทำการวิเคราะห์ต่อไปว่าแต่ละ segment มีคุณสมบัติอย่างไร โดยสามารถแบ่งได้เป็น 2 ประเภทหลักได้แก่

1. Semantic segmentation คือ การวิเคราะห์ pixel บนภาพ เพื่อค้นหาประเภทของวัตถุ(class)ที่สนใจ โดยจะสร้าง mask บนรูปเพื่อแสดงตำแหน่ง pixel ที่มีวัตถุนั้นอยู่
2. Instance segmentation เป็นขั้นถัดไปของ semantic segmentation โดยจะทำการวิเคราะห์ pixel ที่ถูก mask เพื่อแบ่ง object ของ class ในนั้น เช่น แบ่ง class “คน” ที่ปรากฏบนรูป ออกเป็น object ของบุคคลเดี่ยวหลายๆคน

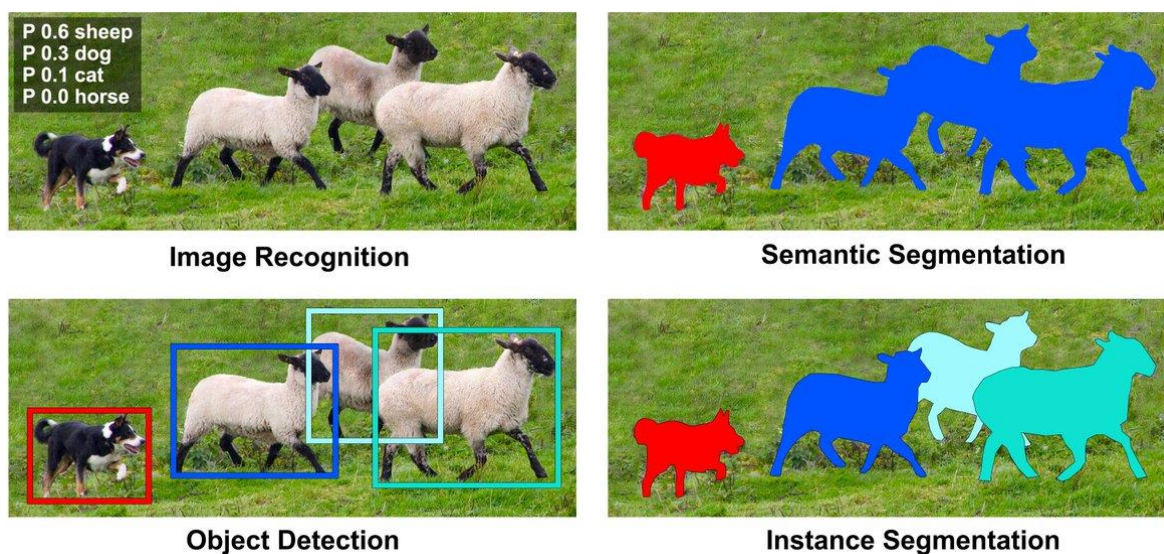


FIGURE 1: Compare between each machine learning model

โดยในโครงการนี้มีจุดประสงค์ในการใช้ machine learning model ในการวิเคราะห์รูปภาพเพื่อนับจำนวนผลมะม่วงและจำแนกความสุก จึงเจาะจง machine learning model ที่สามารถใช้เทคนิค instance segmentation ได้^[1]

2.1.2 Convolutional Neural Network

เป็น deep neural network architecture ที่ถูกใช้อย่างแพร่หลายที่สุดในงาน computer vision โดยสามารถใช้ในการทำ image recognition, Video recognition, Recommender systems หรือแม้แต่ natural language processing ตัวอย่างของ model ที่ใช้สถาปัตยกรรมนี้ เช่น AlexNet, VGGNet, ResNet เป็นต้น ประกอบไปด้วย 3 layer หลัก ได้แก่

1. Convolutional layer: ใช้ในการจำแนก input image ผ่าน filter และ kernel

2. Pooling layer: ใช้ในการย่อ sample feature map โดยการสรุปยอดรวมของ feature มัน ลง feature map
3. Fully connected layer: เชื่อม neuron เข้าด้วยกันทั้งหมดไปยังอีก layer ของมัน

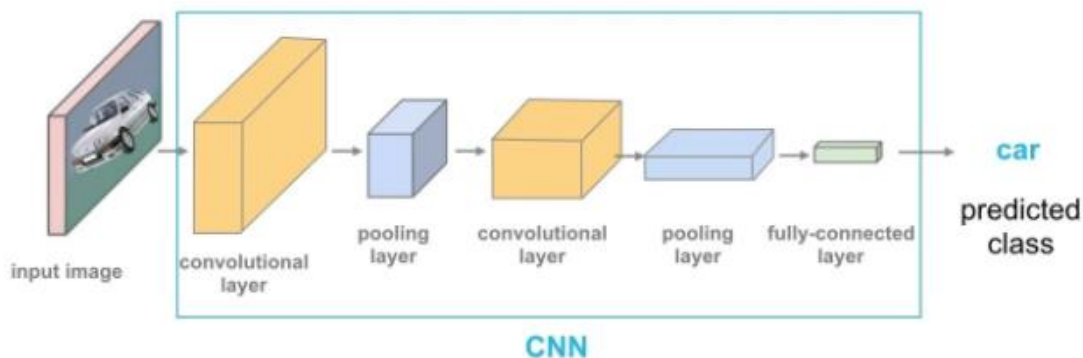


FIGURE 2: CNN MODEL PROCESS

พอเรารวม layer ของ convolutional network เข้าด้วยกัน ก็สามารถ จำแนก object ที่เราสนใจได้ แต่มีข้อเสียคือ มันเหมาะกับ รูปที่มีแค่ object เดียวๆ เท่านั้น เราเลยหยิบ mask R-CNN เข้ามาช่วยในการจำแนก object ที่ต้องการหลายๆชิ้นในรูปเดียวกัน^[19]

2.1.3 Mask R-CNN

เป็น Deep Neural Network เพื่อใช้ในการ Solve instance segmentation problem ใน machine learning และ computer vision ซึ่งถูกพัฒนามาจาก Faster R-CNN โดย จะทำการ ใช้ semantic segmentation เข้ากับรูปโดยรวมเพื่อจับ object ที่ shape ใกล้เคียงกับ สิ่งที่เราอยากได้ แล้วค่อยทำ instance Segmentation เพื่อจำแนก object ออกมาให้ชัดเจนขึ้นอีกที

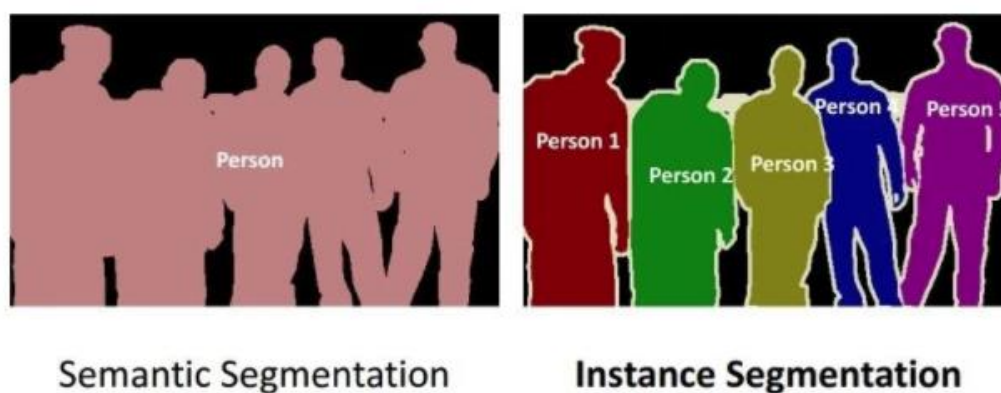
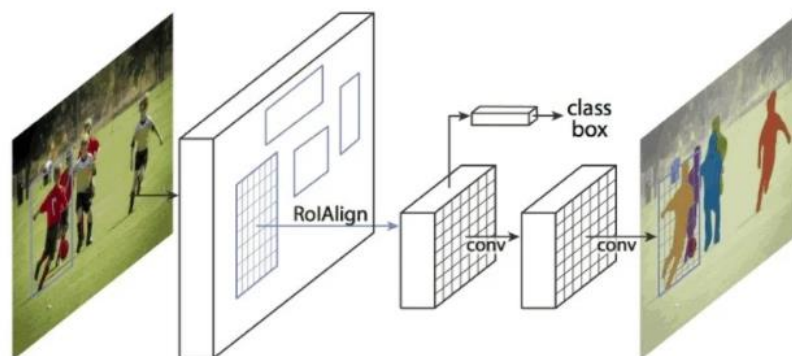


FIGURE 3: Difference between Instance and Semantic Segmentation

จาก Design, RCNN ทำขึ้นมาเพื่อ แก้ปัญหา image detection โดยใช้ bounding boxes crop รอบ ROI (Region of Interest) เพื่อหา object ที่ต้องการ รอบ บริเวณนั้นอีกที. Mask R-CNN จะทำงานคล้ายๆ กับ RCNN แต่เร็วกว่ามาก โดย

การเพิ่ม branch ของ mask output เพิ่มอีก 1 ชุด เพื่อ predict เป็น parallel ไปกับ RCNN ทั่วไป และ ลด overhead ของ algorithm ด้วย^[20]



Mask R-CNN – The Mask R-CNN Framework for Instance Segmentation

FIGURE 4: Mask R-CNN Framework for instance Segmentation

2.1.4 Evaluation metrics

การตรวจความแม่นยำของ model สามารถทำได้โดยใช้ mean Average Precision หรือ mAP

ขั้นตอนแรก ได้แก่ การหาค่า Intersection over Union (IoU) หรือ Jaccard Index คำนี้นหาได้จาก บริเวณ overlap ระหว่าง prediction และ ground truth / บริเวณ union ของ prediction และ ground truth หากมีค่า IoU สูงกว่าค่าๆหนึ่ง จะจัดว่าเป็น True Positive ไม่เช่นนั้นจัดว่าเป็น False Positive หาก ground truth หนึ่ง มีหลาย detection แล้ว detection ที่มี IoU สูงที่สุดจะถูกเลือกเป็น True Positive

หลังจากนั้น ทำการหา Precision และ Recall ของทุก detection โดยที่ $\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$

และ $\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$

ค่า Precision และ Recall เหล่านี้ ถูกนำมา plot เป็น PR หรือ Precision Recall curve มีพื้นที่คือค่า AP โดย mAP หาได้จากค่าเฉลี่ยของ AP ระหว่างทุก object category^[21]

2.1.5 Transfer Learning

เป็น Technique ในการถ่ายทอดความรู้จาก ตัวที่เคย Train มาแล้วจาก task หนึ่ง มาให้อีก model หนึ่งที่ทำคนละ task แต่ Dataset ที่ Train นั้นมีความใกล้เคียงกัน เพื่อลดระยะเวลาในการ Train โดยการนำ weight ของ model ที่ train มาแล้วมาใช้ initialize ค่า weight ของ model ที่เราจะ train ใหม่

2.1.6 Fine-tuning

เป็นเทคนิคคล้ายกับ transfer learning โดยการ train model จาก pre-trained model อื่น โดยการตัด layer สุดท้ายสำหรับแสดงผลของ model ทิ้ง จากนั้นจึงทำการเพิ่ม layer ใหม่สำหรับ class ชนิดใหม่ที่เรากำลังต้องการ แล้วจึงทำ

การ train dataset ชุดใหม่เข้าไปเพื่อให้ model สามารถจำแนก class ของวัตถุชนิดใหม่ที่ไม่เคยถูก train มาก่อนได้ (ความแตกต่างจาก transfer learning คือ การ train dataset ชุดใหม่จะส่งผลต่อ feature extraction layer ของ pre-trained model ที่นำมาใช้ด้วย ทำให้ weight ของ pre-trained model เปลี่ยนแปลงได้)

2.2 ทฤษฎีเกี่ยวกับระบบ IOT

โดยรวมแล้วเป็นการสร้างอุปกรณ์ที่จะต้องสามารถ ถ่ายรูป จับอุณหภูมิ ถ่ายภาพ ตามช่วงระยะเวลาที่กำหนดได้ รวมถึง สามารถ ทำงานได้เป็นระยะเวลานาน ไม่ต้องพึ่ง พลังงานภายนอก เราจะจำแนกจาก แนวนี้ออกมาเป็นชิ้นอุปกรณ์ย่อย

2.2.1 Nodemcu V3 (ECH 8027)

ใช้ต่อสัญญาณเชื่อมระหว่าง Arduino กับ Web Server (Code ผ่าน Arduino)

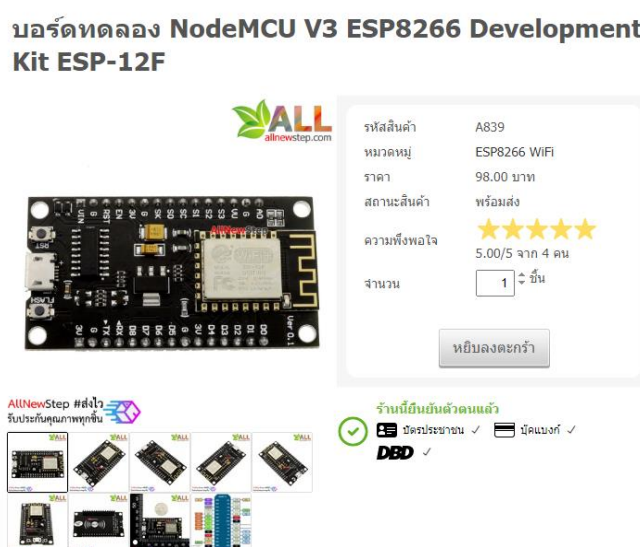


FIGURE 5: Example of NodeMCU V3 ESP8266 Board that are available in shop

2.2.2 Arduino Mega R3 Board (2560)

ใช้ต่อสัญญาณเชื่อมระหว่าง Arduino กับ Web Server (Code ผ่าน Arduino)



Arduino Mega (R3) Board

FIGURE 6: Arduino Mega (R3) Board

เราใช้ board Arduino mega แทนที่จะเป็น uno r3 (ตามปกติ) เพราะว่า เราต้องการ port i/o เพิ่ม โดย ถ้าเป็น uno เราจะทำ receive/transmit ได้แค่ 3 อุปกรณ์ กับ uno เท่านั้น แต่ ถ้า mega จะได้ถึง 9 ตัว รวมถึงมี memory เก็บสูง กว่า board ทั่วไป



FIGURE 7: Example of Arduino Mega R3 Board that are available in shop

2.2.3 ESP32 CAM OV2640 module

ตัวกล้อง พร้อมสัญญาณ wifi ติดมาด้วย คุณภาพกล้อง spec นี้เป็น OV2640 module 12 MP ได้ resolution เยอะสุดคือ 1600*1200 (pixel - 200 W)

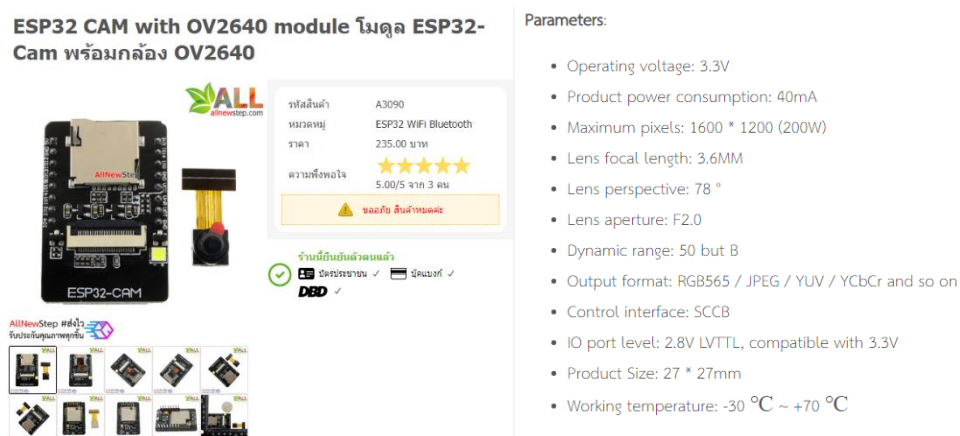


FIGURE 8: Example of ESP32 CAM with OV2640 that are available in shop

2.2.4 Solar Cell Charger Controller & solar panel 18V 0.55A 10W 270*345mm & LIPO Battery

ตัวจ่ายไฟให้กับ ตัวบอร์ดของเรา เพื่อที่จะไม่พึ่ง การเสียบปลั๊กไฟให้ลงภาคสนามได้ ใช้ solar cell เต็มไฟเข้ากับ เข้ากับ LIPO Battery เพื่อให้เก็บไฟใช้ตลอดทั้งวัน

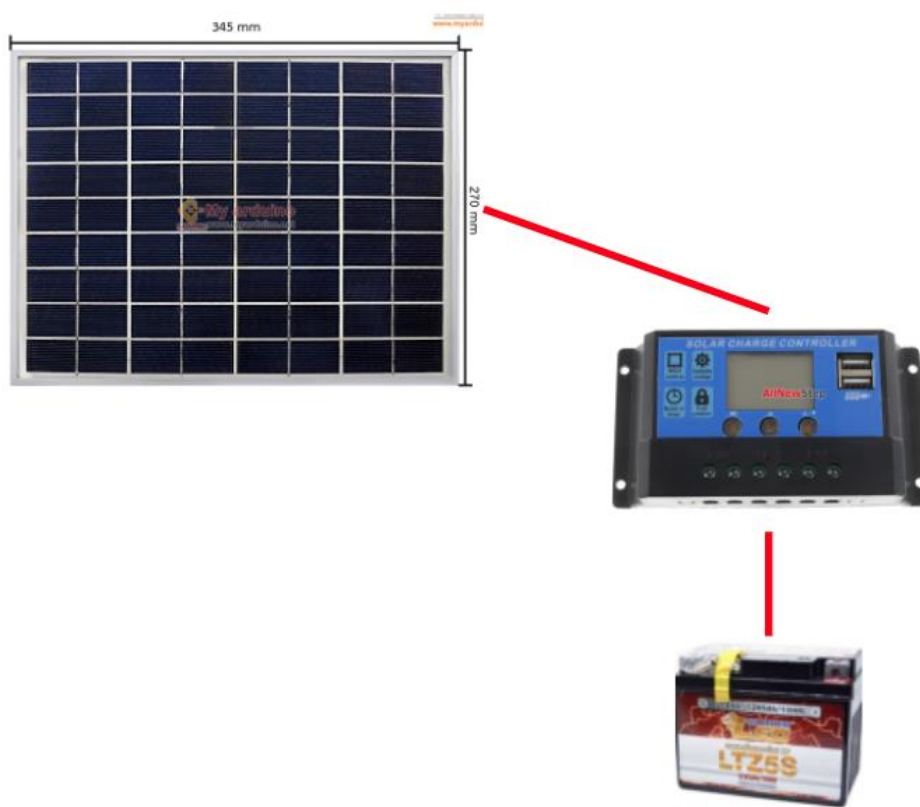


FIGURE 9: Simple Wire of Power module System

2.3 ทฤษฎีที่เกี่ยวข้องกับ Cloud application

2.3.1 Serverless architecture

คือ การรัน application บน cloud server โดยผู้ให้บริการ cloud จะช่วยจัดการงานด้าน virtual environment สำหรับรัน application เช่น การจัดการทรัพยากรสำหรับรัน application, พื้นที่จัดเก็บข้อมูล, entry point สำหรับรับส่ง request รวมถึง network security ต่างๆ โดยในการใช้บริการ cloud server จะต้องจ่ายเงินตามเวลาและจำนวนทรัพยากรที่ใช้รัน application ซึ่งสามารถ scale ได้ตามความเหมาะสมของขนาด application และจำนวนผู้ใช้งาน

Traditional vs Serverless Architecture

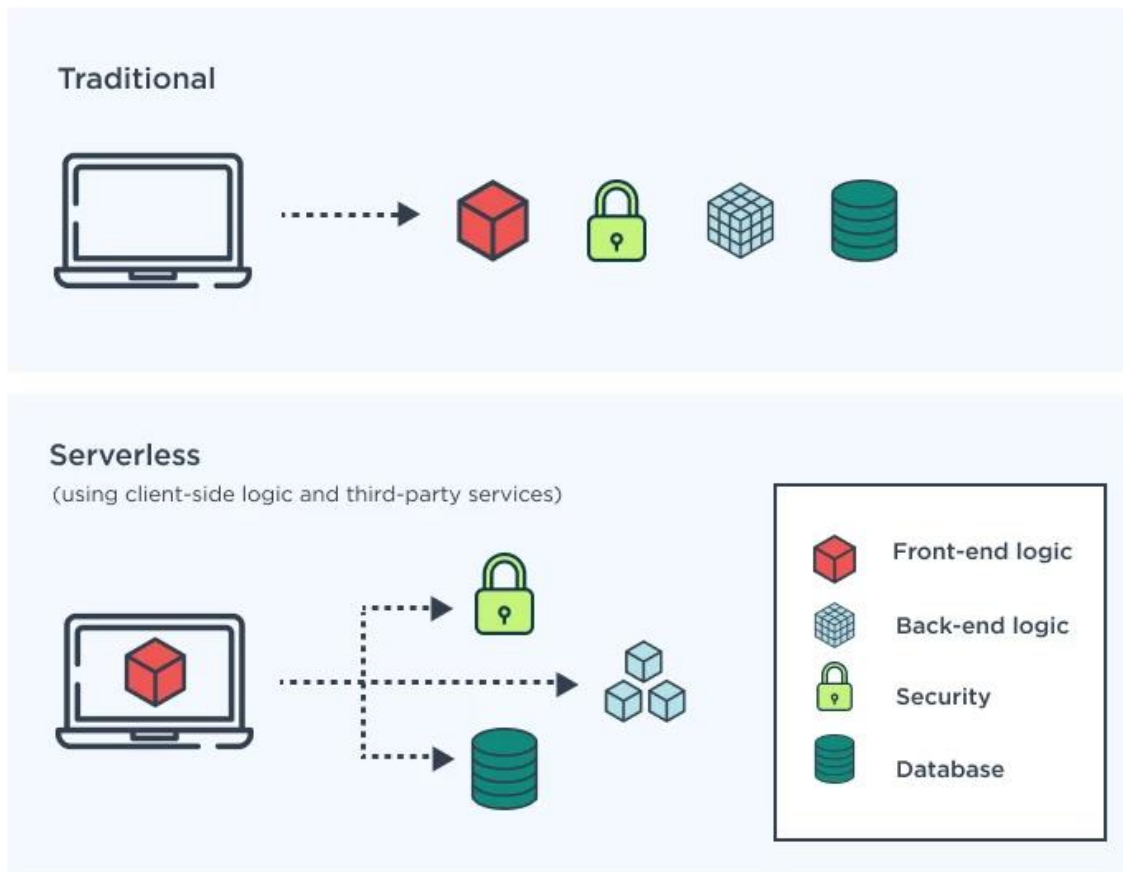


FIGURE 10: Graph compare Traditional & serverless architecture

2.3.2 Software as a Service (SaaS)

คือ ประเภทหนึ่งของ cloud platform ที่ให้บริการ software ประเภทต่างๆผ่านสถาปัตยกรรม cloud ซึ่งส่งผลให้ผู้ให้บริการสามารถเข้าใช้บริการของ software ประเภทนี้ได้ผ่านทางอินเทอร์เน็ต โดยไม่จำเป็นต้องดาวน์โหลดและติดตั้ง software บนอุปกรณ์ของตนเอง ตัวอย่างของ SaaS เช่น AWS, Google drive, Google workspace ต่างๆ เป็นต้น

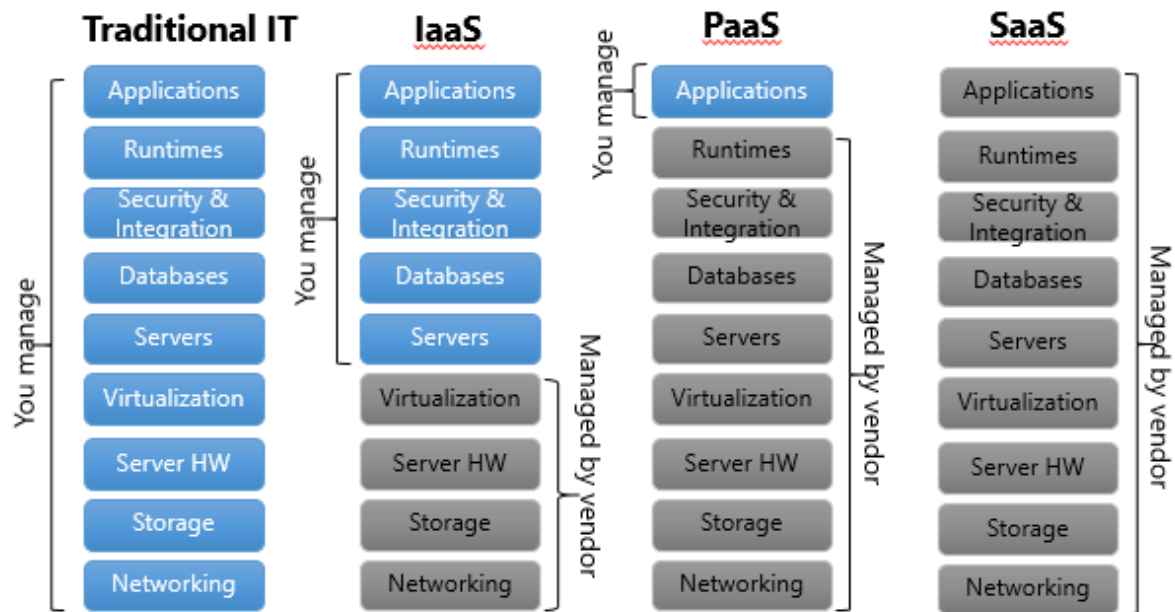


FIGURE 11: Graph compare Traditional & serverless architecture

2.3.3 Amazon web services (AWS)

เป็น platform จากเครือ Amazon ที่ให้บริการ cloud computing ประเภทต่างๆ เช่น พื้นที่จัดเก็บข้อมูล, server สำหรับรัน application เป็นต้น โดยในโปรเจกต์นี้ได้มีการหยิบเอา services ของ AWS มาใช้ดังนี้

- **API Gateway:** เป็น entry point สำหรับการรับ request จากภายนอก AWS server โดย service นี้จะช่วยจัดการกับ request ที่ได้รับอย่างเหมาะสม ได้แก่ การส่งต่อ request ไปยัง route สู่ services อื่นๆ, การดูแล traffic ในการรับ request รวมถึงการสร้าง policy สำหรับดูแลความปลอดภัยและตรวจสอบสิทธิ์ในการเข้าถึง services ต่างๆ ของผู้ทำการส่ง request
- **Lambda function:** เป็นบริการประมวลผลแบบ serverless ซึ่งสามารถอัปโหลดโค้ดเพื่อรัน backend logic ควบคุมการทำงานของ application ได้ (เข้าถึงผ่าน API gateway) รวมถึงสามารถใช้เป็น environment สำหรับรัน machine learning model ที่มีขนาดไม่ใหญ่มากได้อีกด้วย (แต่ในกรณีของโปรเจกต์นี้ model มีขนาดใหญ่เกินไป ทำให้เลือกใช้วิธี deploy บน EC2 แทน) โดยในโปรเจกต์นี้ได้ใช้ service ตัวนี้สำหรับ format รูปภาพที่ได้จากอุปกรณ์ IOT ให้เป็น ascii เพื่อนำไปเก็บใน S3 storage ต่อไป
- **S3 storage:** เป็น cloud storage ง่ายสำหรับใช้เก็บข้อมูลบน AWS server โดยในโปรเจกต์นี้ในการเก็บข้อมูล ได้แก่ รูปภาพมเมว่งและข้อมูลที่ส่งมาจากระบบ IOT, Machine learning model (train locally) และ dependency modules ต่างๆ

- **EC2:** เป็นบริการสำหรับสร้าง Image instance เพื่อใช้รัน application บน operating system ต่างๆ โดยในโปรเจกต์นี้ได้ใช้ service นี้สำหรับ deploy machine learning model เพื่อทำการ detect จำนวนมะม่วงจากรูปภาพ
- **IAM role:** เป็น service สำหรับแบ่ง role ให้กับ user ประเภทต่างๆเพื่อจำกัดสิทธิ์ในการเข้าถึง service ที่ต่างกัน โดยสามารถใช้งานได้ด้วยวิธี login เข้า AWS ผ่าน IAM user โดยตรงหรือใช้ access key ของ IAM role ก็ได้

2.3.4 Google Sheet: เป็น Google workspace อย่างง่ายสำหรับเก็บข้อมูลในรูปแบบของ excel หรือ csv โดยในโปรเจกต์นี้ได้ใช้ service ตัวนี้ในการเก็บข้อมูลจำนวนมะม่วงที่ได้จากการประมวลผลของ machine learning model ที่รันบน EC2 instance

2.3.5 Power BI: เป็นอุปกรณ์สำหรับการวิเคราะห์ข้อมูลและนำเสนอผลลัพธ์ โดยจะรับข้อมูลในรูปแบบต่างๆ เช่น ไฟล์ excel, csv, database หรือ website เพื่อนำมาวิเคราะห์และนำเสนอผลลัพธ์ในรูปแบบของกราฟหรือ dashboard ต่างๆ นอกจากนี้ยังสามารถแชร์ผลลัพธ์ที่ได้ในรูปแบบของ url ให้คนทั่วไปได้รับชมได้อีกด้วยจึงเรียกว่าผลลัพธ์จากอุปกรณ์ตัวนี้ก็เป็น serverless ในรูปแบบหนึ่ง โดยในโปรเจกต์นี้ได้ใช้ Power BI เป็น frontend ในการนำเสนอผลลัพธ์ โดยการให้อุปกรณ์เชื่อมต่อกับ Google Sheet และสร้าง dashboard นำเสนอผลลัพธ์จำนวนมะม่วงในช่วงเวลาต่างๆ

2.4 ทฤษฎีและงานวิจัยอื่นๆที่เกี่ยวข้อง

2.4.1 มะม่วงน้ำดอกไม้สีทอง



FIGURE 12: Sample of mango in designated farm

วงศ์ : Anacardiaceae

ชื่อภาษาไทย : มะม่วงน้ำดอกไม้สีทอง

ชื่อวิทยาศาสตร์ : *Mangifera indica* L.

พันธุ์ : น้ำดอกไม้สีทอง

ลักษณะ : มีรูปร่างทางพฤกษศาสตร์เหมือนกับมะม่วงทั่วไปทุกอย่าง ผลเป็นรูปกลมรีและยาว มีเปลือกหนา เมื่อโตเต็มที่แต่ละผลจะมีน้ำหนักเฉลี่ยระหว่าง 300-400 กรัม ออกผลตกเป็นพวงติดกัน 5-7 ผล และสามารถออกผลได้ตลอดทั้งปี เกษตรกรมักจะเก็บเกี่ยวผลผลิตหลังจากดอกบาน 110-115 วัน โดยผลที่เพิ่งออกจะมีสีเขียว และจะเปลี่ยนเป็นสีเหลืองสดเมื่อสุก ผลที่สุกได้ที่จะมีเนื้อเยื่อสีเหลือง รสชาติหวานหอม เมล็ดบางและเลี่ยนน้อย^[5]

วิธีสังเกตระดับความสุก : ตัวแปรหลักในการพิจารณาความสุกคือสี โดยเริ่มนับเวลาตั้งแต่ดอกมะม่วงบาน คือ 1-2 เดือนแรกจะอยู่ในช่วงที่ผลอ่อน สีผลจะเป็นสีเขียว โดยหลังจาก 1-2 เดือนไปแล้วจะเปลี่ยนเป็นสีเหลืองอ่อนตลอดทั้งผล และจะเป็นเช่นนี้ไปจนกระทั่งผลแก่หรือสุกสีเหลืองจะเข้มขึ้น นอกจากนี้ยังสามารถใช้ระดับความสุกจากน้ำหนักผลได้ด้วย โดยเกษตรกรมักนำผลมะม่วงที่เก็บไปลอยน้ำ ซึ่งถ้าผลมะม่วงสามารถลอยน้ำได้ แปลว่า ผลยังไม่สุกเต็มที่^[4]

ช่วงเวลาการเก็บเกี่ยว : เดือนพฤศจิกายนและเดือนเมษายน โดยในระหว่างเดือนส.ค.-เม.ย. อาจมีการเก็บผลผลิตนอกฤดูกาลด้วย^[3]

2.4.2 Coco test-dev เปรียบเทียบประสิทธิภาพ machine learning model

Instance Segmentation on COCO test-dev

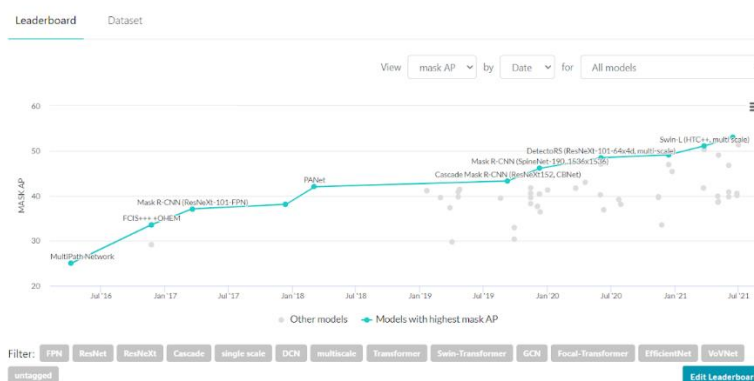


FIGURE 13: Graph Compare model with highest mark AP

กราฟเปรียบเทียบประสิทธิภาพ machine learning model สำหรับทำ instance segmentation โดยใช้ dataset COCO ปี 2021 โดย mask R-CNN(built on ResNet-101, FPN) ได้คะแนน 37.1 mask AP (mask average precision) [17]

2.4.3 Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning

รายละเอียดของงานวิจัยนี้คือการนำโมเดล MaskRCNN มาทำการ detect มะเขือเทศ โดยใช้ อุปกรณ์ Jetson ติดตั้งกล้อง RealSense Camera เพื่อถ่ายภาพขอต้นมะเขือเทศ และนับจำนวนมะเขือเทศที่สุกแล้วบนต้นนั้น

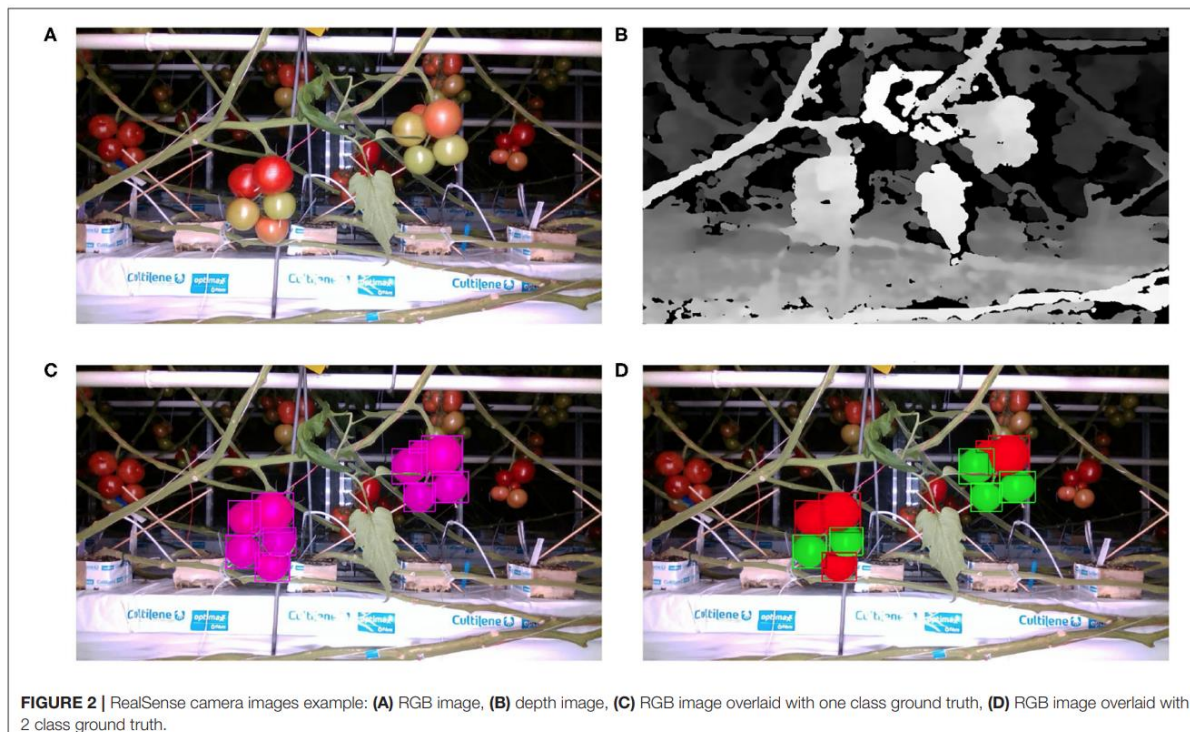


FIGURE 14: Example of Tomato Detection system using mask R-CNN model

กล้อง RealSense Camera ทำการถ่ายแล้วจำแนก Semantic Segmentation แล้วทำการ overlay แยก classification ออกมา ในที่นี้คณะผู้จัดทำ ได้ทำการแยกออกเป็น 2 class เพื่อแยกตัว มะเขือเทศที่สุกมากแล้ว กับตัวมะเขือเทศที่ยังไม่สุกคือออกมา (สีแดงสุกมาก สีเขียวยังไม่สุก) เป็นต้น^[18]

2.4.4 Image Segmentation Using Deep Learning: A Survey

รายละเอียดของงานวิจัยนี้เป็นการเปรียบเทียบ technique ต่างๆ ของ image Segmentation ที่ใช้ใน Deep Learning, อธิบาย เนื้อหาขั้นพื้นฐานของแต่ละตัว และ แนะนำให้ผู้อ่านรู้จักถึง model ที่ยังอยู่ในขั้นวิจัย หรือยังไม่ได้ใช้แพร่หลาย ใน โลกปัจจุบัน เช่น Encoders-Decoders based models , CNNs with graphical models , Fully convolutional models เป็นต้น โดยมีผลสรุปได้เป็น performance การเปรียบเทียบของแต่ละตัว ตามนี้

TABLE 5
Instance segmentation model performance on COCO test-dev 2017

Method	Backbone	FPS	AP
YOLACT-550 [74]	R-101-FPN	33.5	29.8
YOLACT-700 [74]	R-101-FPN	23.8	31.2
RetinaMask [172]	R-101-FPN	10.2	34.7
TensorMask [67]	R-101-FPN	2.6	37.1
SharpMask [173]	R-101-FPN	8.0	37.4
Mask-RCNN [62]	R-101-FPN	10.6	37.9
CenterMask [72]	R-101-FPN	13.2	38.3

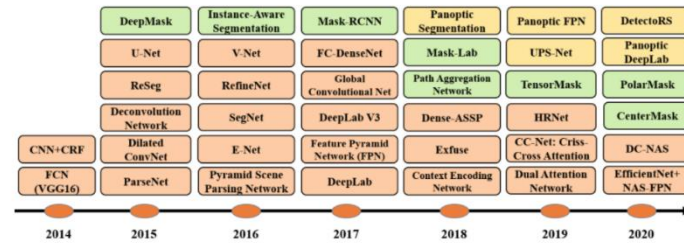


Fig. 33. Timeline of representative DL-based image segmentation algorithms. Orange, green, and yellow blocks indicate semantic, instance, and panoptic segmentation algorithms, respectively.

FIGURE 15: Comparing Efficiency of each instance segmentation model

จะสังเกตว่าใน Model ระหว่าง YOLACT, RetinaMask - SharpMask, Mask-RCNN และ CenterMask ตัว Mask-RCNN ได้ผลลัพธ์การทำงาน AP (Aptitude Performance) ที่เกือบเยอะสุด รองลงมาจาก CenterMask ที่ 37.9 และ FPS อยู่ในระดับที่ได้มาตรฐาน ประกอบกับ Mask-RCNN ได้ทำการเปิด Public แพร่หลายมากกว่า มี opensource ให้เลือกใช้จำนวนมาก จำเป็นตัวเลือกที่ดีในการจัดทำ instance segmentation.^[20]

บทที่ 3: ระบบติดตามผลผลิตมะม่วง

ระบบติดตามผลผลิตมะม่วงประกอบด้วยส่วนประกอบหลัก 3 ส่วน ได้แก่

3.1 Machine learning model

การทำ project ครั้งนี้ ได้นำ Mask-RCNN มาใช้ สำหรับการทำ deep learning เพื่อนับจำนวนมะม่วงบนต้นของรูปภาพ โดยที่การทำ deep learning ได้มีการใช้ เว็บ Makesense.ai เพื่อทำการ annotate รูปภาพ โดยที่ทำการ annotate แบบ polygons และได้เป็น JSON หลังจากนั้น นำ dataset ที่ annotate แล้ว มา load เพื่อทำ deep learning โดยที่ทำการ training บน Google Colab โดยที่ dataset สุดท้าย มีขนาด 245 รูป แบ่งเป็น รูปสำหรับ training 218 รูป และ รูปสำหรับ validation 27 รูป โดยที่ dataset นี้ มีทั้งหมด 1648 instances แบ่งเป็น 1413 สำหรับ training และ 235 สำหรับ validation นอกจากนี้ ยังมี test dataset 8 รูป 86 instances

Dataset ได้มีการทำ augmentation สำหรับ image ของ training set และ validation set โดยที่การทำ image augmentation เป็นการเพิ่มความหลากหลายของ dataset และเพิ่มขนาดของ dataset ด้วยเช่นกัน โดยที่ การทำ image augmentation มีหลากหลายวิธี โดย ทำการ augment แบบ manual ทั้งด้วยภาษา python บน Google Colab และ Microsoft Word โดยใช้วิธีการ crop รูป ปรับแสง rotate รูป ยีดยุรูป เป็นต้น

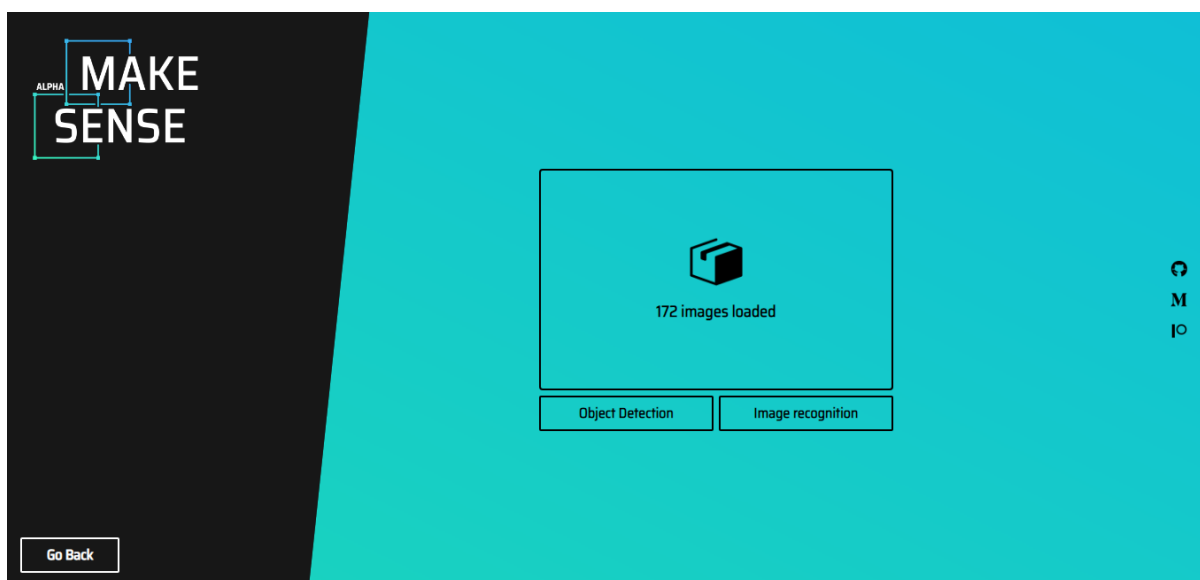


FIGURE 16: Picture of MakeSense.ai application

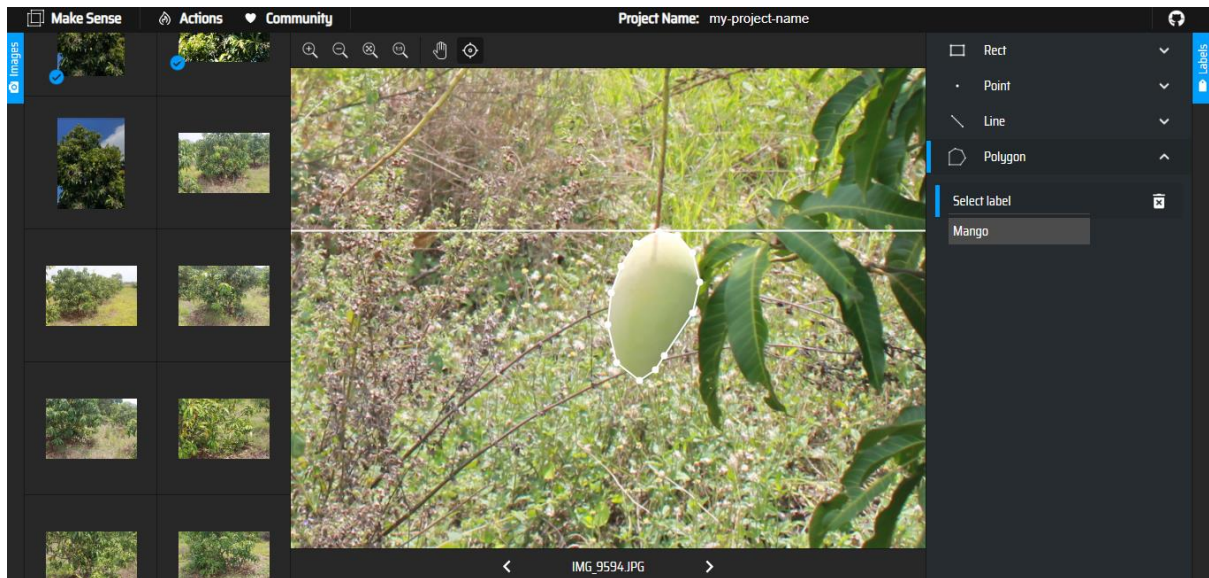


FIGURE 17: example of annotation using polygon in makesense.ai



FIGURE 18: example of annotated mango in makesense.ai



FIGURE 19: Another example of annotated mango in makesense.ai

3.2 ระบบ IOT

ในส่วนของตัวระบบ IOT ในเชิง Design เรามีหลายแบบมาก โดยในที่นี้เราจะพูดถึง format สุดท้ายที่ใช้เท่านั้น ขอเริ่มจาก Diagram ภาพรวมของตัวอุปกรณ์ที่เราจัดทำขึ้น

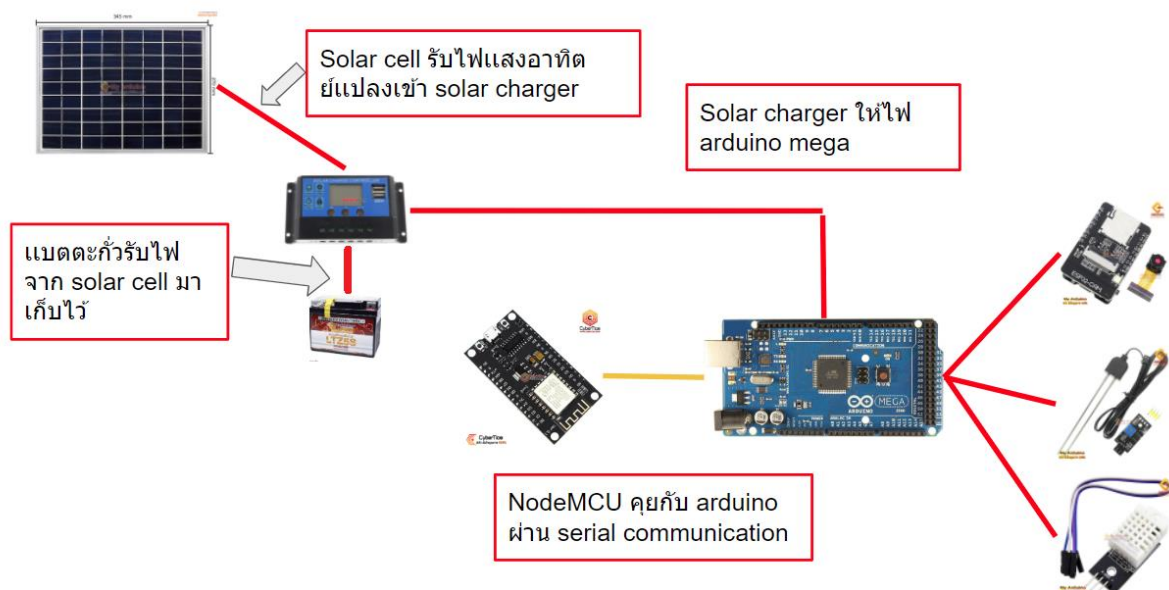


FIGURE 20: Full diagram of lot system

โดยรวมแล้วตัวอุปกรณ์ของเรา หน้าที่หลักคือถ่ายรูปในช่วงเวลาที่เราต้องการ และส่ง Data ขึ้น AWS Cloud ที่เรา assign ไว้ได้ ผ่าน wifi ที่อยู่ที่สวน และต้องมีความสามารถ ที่จะทำงานได้ โดยไม่ต้องต่อสายไฟ เพราะ ณ จุดฟาร์ม มันไม่มีสายให้เชื่อม

เราจึงทำระบบ ขึ้นมา 2 ส่วน คือ ส่วนที่เรา รับค่า Sensor ต่างๆ จาก อุปกรณ์ที่เราติดตั้ง ได้แก่ รูปภาพ ความชื้นในดิน และ อุณหภูมิ ณ จุดที่เราติดตั้ง ส่งค่าขึ้น cloud ผ่าน wifi โดยมีการประสานงานผ่าน NodeMCU Arduino V5 (ตัวควบคุมโค้ดหลัก และส่ง / รับ ค่าต่างๆ ขึ้นwifi โดยใช้ esp8266)

3.2.1 ส่วน เก็บ/รับ/ส่ง ข้อมูลเข้าสู่ AWS Cloud โดยประกอบด้วย รูปภาพ และค่าอุณหภูมิ-ความชื้นที่เราเก็บได้

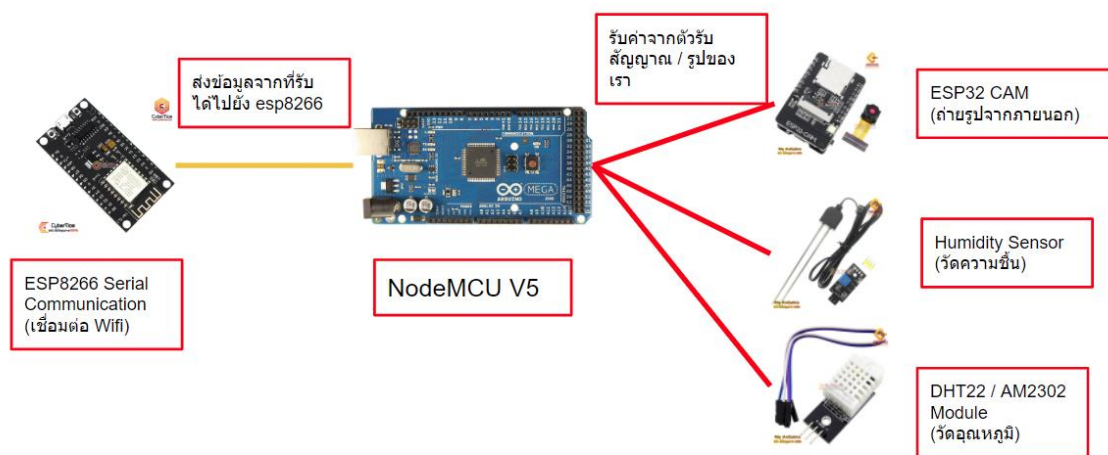


FIGURE 21: Diagram Part of sensor and communication system

โดยหน้าต่างข้อมูลที่เราได้ ณ ตอนส่งค่าขึ้น AWS แล้วจะเป็น รูปภาพ และ String ค่าความชื้น + อุณหภูมิที่เราวัดได้เป็นดังรูป

imageFolder/ Copy S3 URL

Objects Properties

Objects (848)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 Inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URL Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix Show versions

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	1644557860740	-	February 11, 2022, 12:37:41 (UTC+07:00)	54.5 KB	Standard
<input type="checkbox"/>	1644557890547	-	February 11, 2022, 12:38:11 (UTC+07:00)	54.5 KB	Standard
<input type="checkbox"/>	1644557920957	-	February 11, 2022, 12:38:42 (UTC+07:00)	54.6 KB	Standard
<input type="checkbox"/>	1644557951776	-	February 11, 2022, 12:39:12 (UTC+07:00)	54.7 KB	Standard
<input type="checkbox"/>	1644557980592	-	February 11, 2022, 12:39:41 (UTC+07:00)	54.6 KB	Standard
<input type="checkbox"/>	1644558010566	-	February 11, 2022, 12:40:11 (UTC+07:00)	54.7 KB	Standard
<input type="checkbox"/>	1644558040254	-	February 11, 2022, 12:40:41 (UTC+07:00)	54.5 KB	Standard
<input type="checkbox"/>	1644558070673	-	February 11, 2022, 12:41:11 (UTC+07:00)	54.5 KB	Standard
<input type="checkbox"/>	1644558100471	-	February 11, 2022, 12:41:41 (UTC+07:00)	51.0 KB	Standard
<input type="checkbox"/>	1644558130271	-	February 11, 2022, 12:42:11 (UTC+07:00)	52.3 KB	Standard
<input type="checkbox"/>	1644558160071	-	February 11, 2022, 12:42:41 (UTC+07:00)	51.8 KB	Standard
<input type="checkbox"/>	1644635636642	-	February 12, 2022, 10:13:58 (UTC+07:00)	82.4 KB	Standard
<input type="checkbox"/>	16446356666141	-	February 12, 2022, 10:14:27 (UTC+07:00)	59.5 KB	Standard
<input type="checkbox"/>	1644635696121	-	February 12, 2022, 10:14:57 (UTC+07:00)	67.5 KB	Standard
<input type="checkbox"/>	1644635726221	-	February 12, 2022, 10:15:27 (UTC+07:00)	59.5 KB	Standard
<input type="checkbox"/>	1644635773241	-	February 12, 2022, 10:16:14 (UTC+07:00)	84.7 KB	Standard
<input type="checkbox"/>	1644635798637	-	February 12, 2022, 10:16:27 (UTC+07:00)	84.5 KB	Standard
<input type="checkbox"/>	1644635815361	-	February 12, 2022, 10:16:56 (UTC+07:00)	55.9 KB	Standard
<input type="checkbox"/>	1644635846742	-	February 12, 2022, 10:17:27 (UTC+07:00)	104.3 KB	Standard
<input type="checkbox"/>	1644635880043	-	February 12, 2022, 10:18:01 (UTC+07:00)	69.4 KB	Standard
<input type="checkbox"/>	1644635906362	-	February 12, 2022, 10:18:27 (UTC+07:00)	92.0 KB	Standard

FIGURE 22: Example json file that uploaded to AWS via system



FIGURE 23: Sample picture from esp32 camera that get sent to AWS service

ถ้าเราเปิด file ขึ้นมา จะพบว่า มี data อยู่ใน form .txt file และ รูปภาพ ณ เวลาส่งตอนนั้น (ตัวอย่างรูปภาพ + file text ที่เราจะได้เมื่อเปิด json ออกมา โดยตัวรูปภาพผลลัพธ์ที่เราได้มาจะมีความละเอียดค่อนข้างสูงเมื่อเทียบกับ size ของตัวกล้อง Esp32CAM (ความละเอียด 5 ล้าน pixel)

3.2.2 ส่วนจ่ายพลังงานเข้ากับตัวระบบ (Solar Cell + Battery รถยนต์)

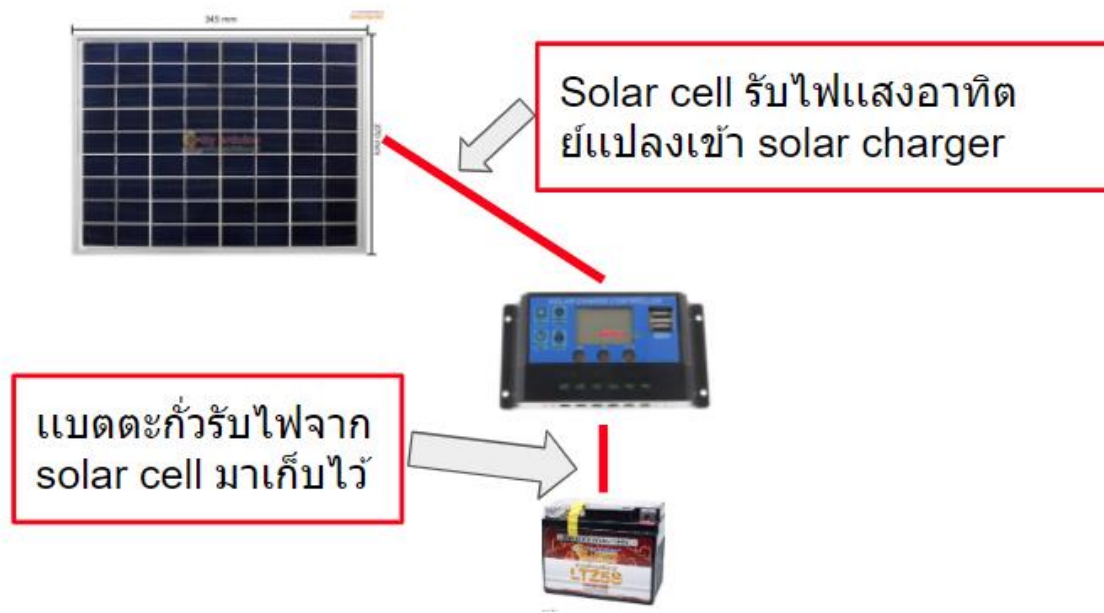


FIGURE 24: Sample picture from esp32 camera that get sent to AWS service

ในแง่ของแนวคิดแล้ว เราต้องการที่จะ ได้แหล่งเก็บไฟที่มีก้อนขนาดใหญ่หลายๆ (มากจริงๆ เมื่อเทียบกับตัวระบบ ณ ปัจจุบัน เพื่อรองรับการขยาย scale ของระบบในอนาคต รวมถึงแผง Solar Cell ด้วย

เราใช้ แบตเตอรี่รุ่น LTZ55 ซึ่งเป็นแบตเตอรี่ รถจักรยานยนต์ย่น ต่อเข้ากับ Solar cell แผงขนาด 18V 0.55A 10W 270*345mm ซึ่งจะจ่ายไฟให้กับอุปกรณ์ได้นาน ประมาณ 10 ชม รวมถึงตัว Solar controller ซึ่งใช้ควบคุม และ Tune กระแสไฟให้ Drop จาก 18 V เหลือ 5 V ให้เหมาะกับการใช้ใน วงจรของเรา

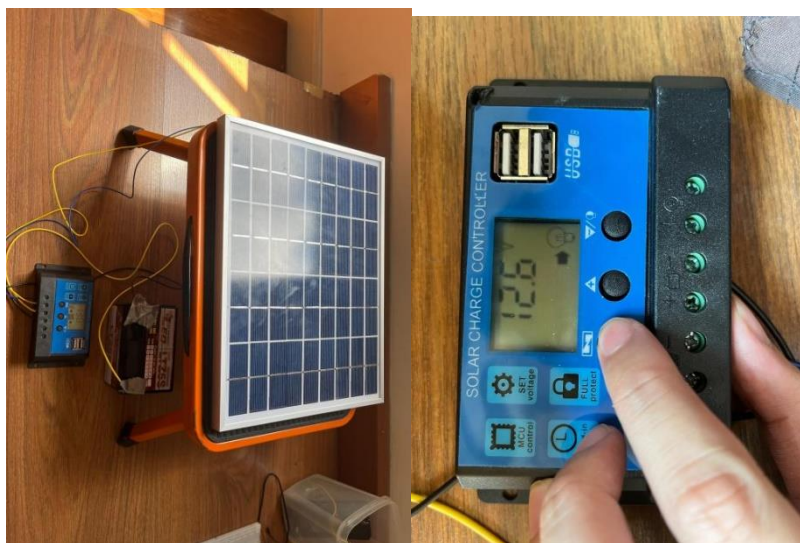


FIGURE 25: Sample usage of solar cell system

3.2.3 การต่อรวมวงจรเข้ากับ ตัวกล่องที่เราออกแบบไว้

ภาพตัวอย่างการ Test ตามรูป เป็นการทดสอบว่า ตัว Solar cell และตัว Controller สามารถจ่ายไฟได้ตามปรกติ โดยตัวไฟจะวิ่งที่ 12.6V เข้า battery แล้วออกไปยังวงจรที่ 5 V

หลังจากที่เราได้ตัวระบบ คร่าวๆ แล้วก็จะทำตัว Infrastructure เพื่อ ใส่วงจรของเราลงกล่อง จะให้ดูตอนสุดท้ายที่ ประกอบเสดก่อนว่าเป็นอย่างไร

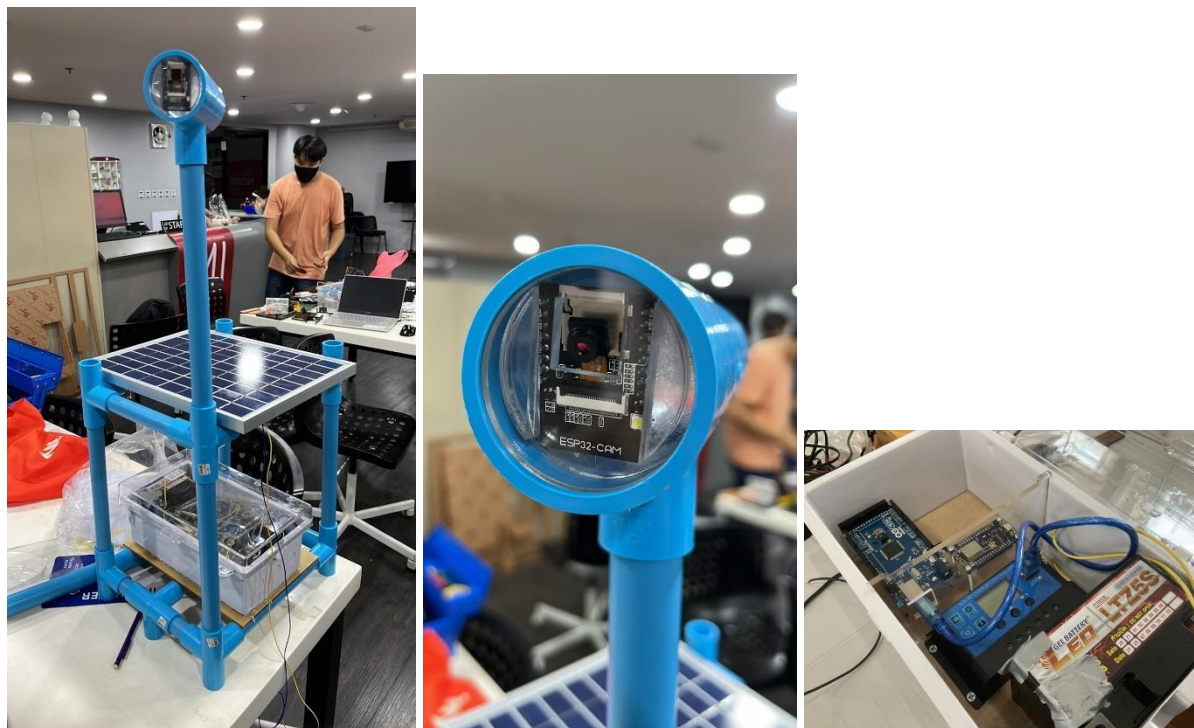


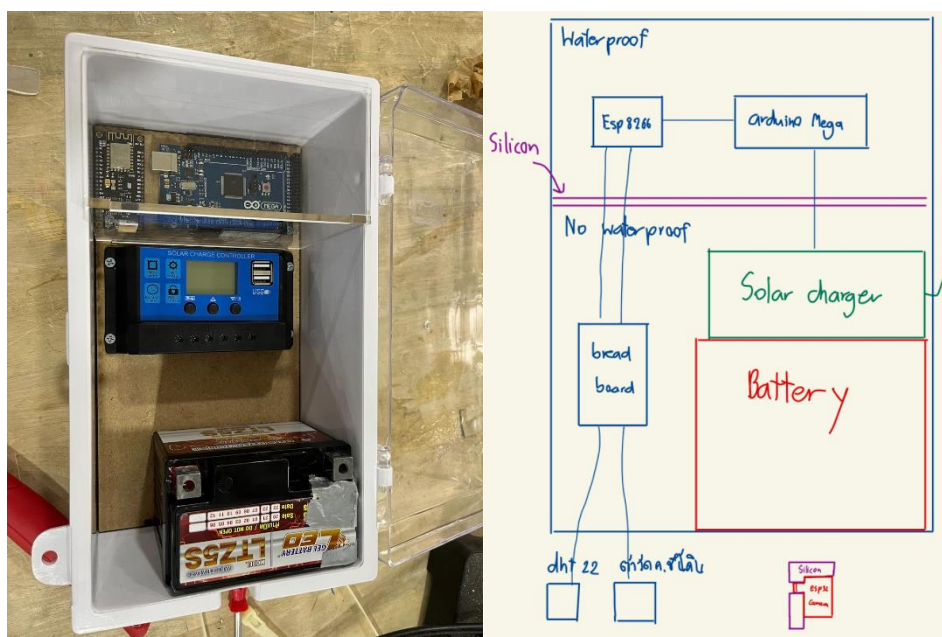
FIGURE 26: Our system after incorporated all require item

เนื่องจากว่าเป็น Prototype แรก เราใช้ ท่อ PVC ยึดตัวอุปกรณ์เข้าด้วยกัน แล้วเชื่อมกันด้วยน็อต ตามร่องต่างๆ, ยึด Solar Cell บนตัว Packaging และลือคกคล้องด้วย อะคริลิก ที่ตัดโดยใช้ laser cutting



FIGURE 27: Sample design for some piece that we need to handcraft

ภายในตัวกล่องของเรา อุปกรณ์ ประเภหห้ามอุปกรณ์ประเภหห้ามโดน น้ำจะถูกครอบด้วยอะคริลิก แล้วเราจะ ปูพื้นกล่องด้วยไม้ mdf เพื่อยึดบอร์ด esp8266, Arduino Mega และ solar charger ด้วยสกรู ส่วน อุปกรณ์อื่นๆ เช่น แบตเตอรี่ ตะกั่ว หรือ breadboard ใช้วิธีติดกาว และพันด้วยสก็อตเทป



ในส่วนของแผงวงจร เราจะต้องจระตาม schematic Diagram คร่าวๆตามนี้

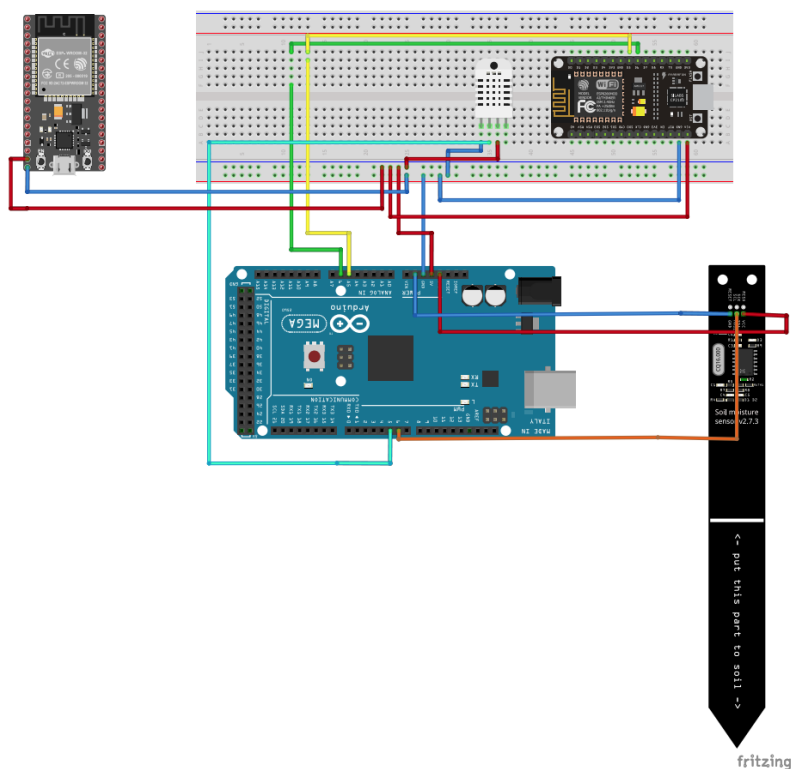


FIGURE 28: Wiring diagram of our system

โดย pin ที่เชื่อมต่อไว้ เป็น pin จริงที่เราใช้อยู่บนบอร์ด เพื่อจะทำการแก้ไขเพิ่มเติมในอนาคต สามารถกลับมาดูอันนี้ได้



FIGURE 29: Picture of system when collecting data from mango farm

ผลลัพธ์ที่ได้จากการเก็บข้อมูลเราจะดึงออกมาจัดเก็บใน Google Drive ต่อไป:

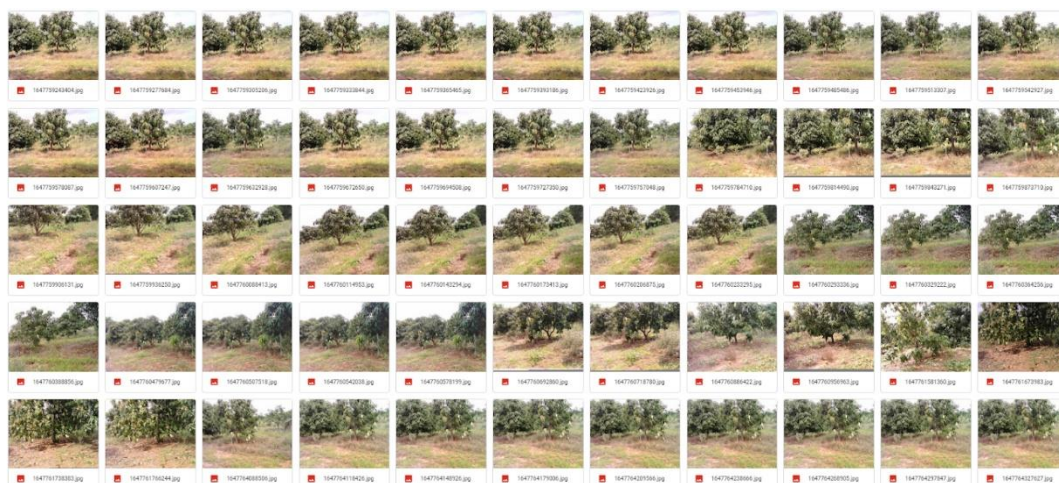


FIGURE 30: Sample picture that are collected from the field

3.3 Cloud application

สามารถแบ่ง flow ของ cloud application ออกเป็น 2 ส่วนหลัก ได้แก่ “flow การรับรูปภาพจากอุปกรณ์ IOT ไปเก็บที่ S3 storage” และ “flow การรัน ML model และอัปเดตข้อมูล dashboard บน Power BI”

3.3.1 Flow การรับรูปภาพจากอุปกรณ์ IOT ไปเก็บที่ S3 storage

1. สร้าง API gateway ชื่อ ImageApi สำหรับการส่ง put request และ publish URL และทำการ route API ไปยัง Lambda function สำหรับ decode รูปภาพและ format เป็น JSON เพื่อส่งไปเก็บยัง S3 storage

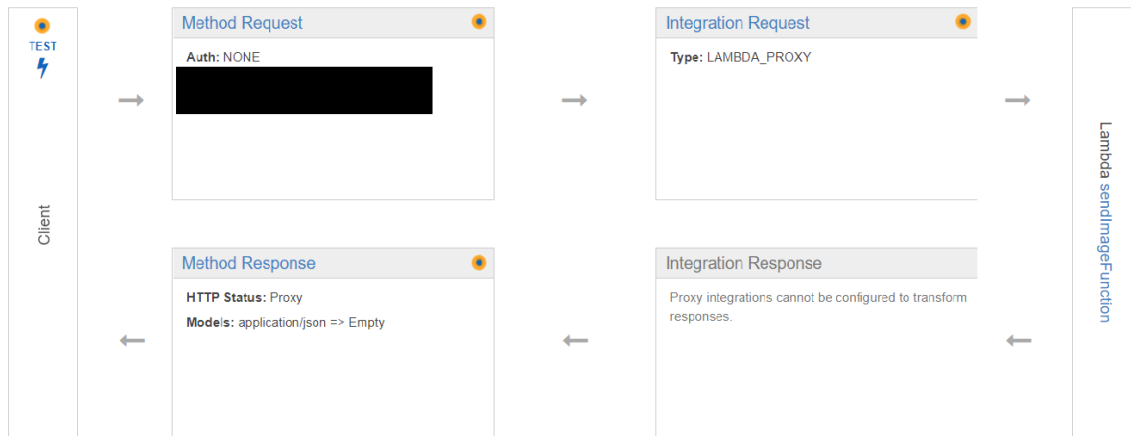


FIGURE 31: API gateway building process

2. เมื่อมีการถ่ายรูป ระบบ IOT ที่รันบน Arduino จะทำการส่งข้อมูลรูปภาพมายัง URL ของ API gateway ผ่าน library “esp_http_client.h” โดยข้อมูลจะถูกส่งไปในรูปแบบ image/jpg จากนั้นจึงถูกส่งผ่าน API gateway ไปยัง Lambda function เพื่อทำการ decode รูปภาพเป็น ascii และ format เป็น JSON แล้วจึงส่งต่อไปเก็บยัง S3 storage
3. ในการใช้งานจริงอุปกรณ์ IOT จะถูกติดตั้งที่สวนมะม่วง และถูกตั้งค่า schedule ในการถ่ายรูปและส่งข้อมูลไปยัง S3 storage วันละ 1 รอบ(ตอนเที่ยงวัน เพราะมีแสงมากที่สุด) รอบละ 5 รูป
4. หากต้องการเข้าถึงรูปภาพบน S3 สามารถทำได้ 2 วิธี ได้แก่ การดาวน์โหลดโดยตรงจาก S3 และเข้าถึงด้วย command ใน AWS cli ผ่านทาง URL ของ S3 object(EC2 จะใช้วิธีเข้าถึงรูปภาพผ่าน AWS cli) โดยคนที่เข้าถึงรูปภาพเหล่านี้ต้องมี access key ของ IAM role เท่านั้น

imageFolder/							Copy S3 URI
Objects Properties							
Objects (848) Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 Inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more							
<input type="button" value="Copy S3 URI"/> <input type="button" value="Copy URL"/> <input type="button" value="Download"/> <input type="button" value="Open"/> <input type="button" value="Delete"/> <input type="button" value="Actions"/> <input type="button" value="Create folder"/> <input type="button" value="Upload"/>							
<input type="text" value="Find objects by prefix"/> <input type="checkbox"/> Show versions							<input type="button" value="1"/> <input type="button" value="2"/> <input type="button" value="3"/> <input type="button" value="4"/>
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class		
<input type="checkbox"/>	1644557860740	-	February 11, 2022, 12:37:41 (UTC+07:00)	54.5 KB	Standard		
<input type="checkbox"/>	1644557890547	-	February 11, 2022, 12:38:11 (UTC+07:00)	54.5 KB	Standard		
<input type="checkbox"/>	1644557920957	-	February 11, 2022, 12:38:42 (UTC+07:00)	54.6 KB	Standard		
<input type="checkbox"/>	1644557951776	-	February 11, 2022, 12:39:12 (UTC+07:00)	54.7 KB	Standard		
<input type="checkbox"/>	1644557980692	-	February 11, 2022, 12:39:41 (UTC+07:00)	54.6 KB	Standard		
<input type="checkbox"/>	1644558010566	-	February 11, 2022, 12:40:11 (UTC+07:00)	54.7 KB	Standard		
<input type="checkbox"/>	1644558040254	-	February 11, 2022, 12:40:41 (UTC+07:00)	54.5 KB	Standard		
<input type="checkbox"/>	1644558070673	-	February 11, 2022, 12:41:11 (UTC+07:00)	54.5 KB	Standard		
<input type="checkbox"/>	1644558100471	-	February 11, 2022, 12:41:41 (UTC+07:00)	51.0 KB	Standard		
<input type="checkbox"/>	1644558130271	-	February 11, 2022, 12:42:11 (UTC+07:00)	52.3 KB	Standard		
<input type="checkbox"/>	1644558160071	-	February 11, 2022, 12:42:41 (UTC+07:00)	51.8 KB	Standard		
<input type="checkbox"/>	1644635636642	-	February 12, 2022, 10:13:58 (UTC+07:00)	82.4 KB	Standard		
<input type="checkbox"/>	1644635666141	-	February 12, 2022, 10:14:27 (UTC+07:00)	59.5 KB	Standard		
<input type="checkbox"/>	1644635696121	-	February 12, 2022, 10:14:57 (UTC+07:00)	67.5 KB	Standard		
<input type="checkbox"/>	1644635726221	-	February 12, 2022, 10:15:27 (UTC+07:00)	59.5 KB	Standard		
<input type="checkbox"/>	1644635773241	-	February 12, 2022, 10:16:14 (UTC+07:00)	84.7 KB	Standard		
<input type="checkbox"/>	1644635786837	-	February 12, 2022, 10:16:27 (UTC+07:00)	84.5 KB	Standard		
<input type="checkbox"/>	1644635815361	-	February 12, 2022, 10:16:56 (UTC+07:00)	55.9 KB	Standard		
<input type="checkbox"/>	1644635846742	-	February 12, 2022, 10:17:27 (UTC+07:00)	104.3 KB	Standard		
<input type="checkbox"/>	1644635880043	-	February 12, 2022, 10:18:01 (UTC+07:00)	69.4 KB	Standard		
<input type="checkbox"/>	1644635906362	-	February 12, 2022, 10:18:27 (UTC+07:00)	92.0 KB	Standard		

FIGURE 32: Sample of image that has been sent from system to AWS server

3.3.2 Flow การรัน ML model และอัปเดตข้อมูล dashboard บน Power BI

1. จากหัวข้อ 3.2 Machine learning model ผลลัพธ์ของ model ที่ train แล้วจะถูก load มาในรูปของ .h5 ไฟล์ และเก็บไว้ใน S3 storage พร้อมกับ dependencies อื่นๆที่จำเป็น เช่น mask RCNN library, coco pretrained weight เป็นต้น
2. สร้าง EC2 instance โดยโหลด .h5 model จาก S3 ผ่านทาง AWS cli และ install dependency ต่างๆที่จำเป็น ทั้งจากใน S3 storage และ pip3
3. สร้าง python script ที่มี environment เป็น python 3.7 สำหรับรัน machine learning model เพื่อนับผลมะม่วง โดยจะทำการดึงรูปภาพ 5 รูปล่าสุดใน S3 storage(คือรูปที่ถ่ายด้วย IOT ณ วันนั้น) มาเป็น input ในการประมวลผล
4. สร้าง schedule สำหรับการรัน script python บน EC2 instance ด้วย Lambda function (ใน 1 วันจะรัน EC2 instance แค่ประมาณ 5 นาทีเพื่อประหยัดค่าใช้จ่าย) โดยจะส่งผลลัพธ์จำนวนมะม่วงที่นับได้ไปเก็บยัง Google Sheet ที่เชื่อมต่อกับ Power BI
5. Power BI จะทำการดึงข้อมูลจาก Google Sheet ที่เชื่อมต่อกับ เพื่อนำข้อมูลที่ได้ไปวิเคราะห์ และนำเสนอผลลัพธ์เป็น dashboard จากนั้นจึง share หน้าแสดงผล ผ่านทาง URL เพื่อให้เกษตรกรได้นำไปใช้ต่อไป โดย Power BI จะสามารถตั้งค่า schedule การอัปเดตข้อมูลจาก Google sheet ได้ ทำให้ข้อมูลที่นำเสนอเป็นข้อมูลล่าสุดจาก Google sheet (หลังจาก update ข้อมูลจาก Google sheet จะมี delay ประมาณ 1 ชม. ก่อนที่หน้าเว็บของ Power BI จะอัปเดตให้ทันข้อมูล)

Scheduled refresh

Keep your data up to date

Configure a data refresh schedule to import data from the data source into the dataset. [Learn more](#)

☒ On

Refresh frequency

Daily

Time zone

(UTC+07:00) Bangkok, Hanoi, Jakarta

Time

1 00 PM

[Add another time](#)

Send refresh failure notifications to

☒ Dataset owner

☐ These contacts:

FIGURE 33: Scheduled refresh interface in powerBI

ดังนั้นการจัด schedule ของ application ใน 1 วัน จะสามารถเรียงลำดับ ได้ดังนี้

1. ให้ IOT ถ่ายรูป และส่งข้อมูลรูปมะม่วงขึ้น S3 storage
2. Restart EC2 instance เพื่อรัน python script เพื่อนับจำนวนมะม่วงจากรูปในข้อ 1 และส่งผลลัพธ์ไปเก็บใน Google sheet
3. ให้ Power BI refresh ข้อมูลจาก Google sheet เพื่ออัปเดตข้อมูลล่าสุดจากข้อ 2

3.3.3 AWS pricing

- Free tier: services ที่ใช้ free tier ได้แก่ S3, Lambda function และ API gateway

<p>Free Tier</p> <p>Amazon EC2</p> <p>750 ชั่วโมง</p> <p>ต่อเดือน</p> <p>ความจุในการประมวลผลที่ปรับขนาดได้ในระบบคลาวด์</p> <p>การใช้งานอินสแตนซ์ Linux, RHEL หรือ SLES t2.micro หรือ t3.micro 750 ชั่วโมงต่อเดือน ขึ้นอยู่กับเรจัน</p> <p>การใช้งานอินสแตนซ์ Windows t2.micro หรือ t3.micro 750 ชั่วโมงต่อเดือน ขึ้นอยู่กับเรจัน</p>	<p>Free Tier</p> <p>Amazon S3</p> <p>5 GB</p> <p>สำหรับพื้นที่จัดเก็บแบบมาตรฐาน</p> <p>โครงสร้างพื้นฐานการวัดเก็บข้อมูลเรจันที่ปลอดภัย คงทน และปรับขนาดได้</p> <p>พื้นที่จัดเก็บแบบมาตรฐาน 5 GB</p> <p>20,000 ก่าบ Get</p> <p>2,000 ก่าบ Put</p>	<p>Free Tier</p> <p>AWS Lambda</p> <p>1 ล้าน</p> <p>ก่าบต่อเดือนฟรี</p> <p>บริการประมวลผลที่รันโค้ดของคุณเพื่อตอบสนองต่อเหตุการณ์และการกรรพพยากรณ์ประมวลผลโดยอัตโนมัติ</p> <p>ฟรี 1,000,000 ก่าบต่อเดือน</p> <p>เวลาประมวลผลสูงสุด 3.2 ล้านวินาทีต่อเดือน</p>	<p>Free Tier</p> <p>Amazon API Gateway</p> <p>1 ล้าน</p> <p>การเรียกใช้ API ที่ได้รับต่อเดือน</p> <p>เผยแพร่ ป่ารุงรักษา ดูแล และรักษาความปลอดภัยของ API ได้ทุกขนาด</p> <p>การเรียกใช้ API ที่ได้รับ 1 ล้านครั้งต่อเดือน</p>
--	--	---	--

FIGURE 34: Amazon tier and feature that are available

- S3 storage pricing (กรณี S3 ใช้ทรัพยากรเกินโควตาของ free tier ได้แก่ การใช้พื้นที่มากกว่า 5Gb หรือรับ put request สำหรับเก็บรูปภาพมากกว่า 2,000 ครั้งใน 1 เดือน)

	PUT, COPY, POST, LIST requests (per 1,000 requests)	GET, SELECT, and all other requests (per 1,000 requests)	Lifecycle Transition requests into (per 1,000 requests)	Data Retrieval requests (per 1,000 requests)	Data retrievals (per GB)
S3 Standard	\$0.005	\$0.0004	n/a	n/a	n/a
S3 Intelligent - Tiering *	\$0.005	\$0.0004	\$0.01	n/a	n/a
Frequent Access	n/a	n/a	n/a	n/a	n/a
Infrequent Access	n/a	n/a	n/a	n/a	n/a
Archive Instant	n/a	n/a	n/a	n/a	n/a
Archive Access, Standard	n/a	n/a	n/a	n/a	n/a
Archive Access, Bulk	n/a	n/a	n/a	n/a	n/a
Archive Access, Expedited	n/a	n/a	n/a	\$10.00	\$0.03
Deep Archive Access, Standard	n/a	n/a	n/a	n/a	n/a
Deep Archive Access, Bulk	n/a	n/a	n/a	n/a	n/a
S3 Standard - Infrequent Access **	\$0.01	\$0.001	\$0.01	n/a	\$0.01
S3 One Zone - Infrequent Access **	\$0.01	\$0.001	\$0.01	n/a	\$0.01
S3 Glacier Instant Retrieval ***	\$0.02	\$0.01	\$0.02	n/a	\$0.03
S3 Glacier Flexible Retrieval ***	\$0.03	\$0.0004	\$0.03	See below	See below
Expedited	n/a	n/a	n/a	\$10.00	\$0.03
Standard	n/a	n/a	n/a	\$0.05	\$0.01
Bulk ***	n/a	n/a	n/a	n/a	n/a
Provisioned Capacity Unit ****	n/a	n/a	n/a	n/a	\$100.00 per unit
S3 Glacier Deep Archive ***	\$0.05	\$0.0004	\$0.05	See below	See below
Standard	n/a	n/a	n/a	\$0.10	\$0.02
Bulk	n/a	n/a	n/a	\$0.025	\$0.0025

FIGURE 35: Amazon pricing for S3 standard

- EC2 Pricing: เนื่องจากการรัน Mask RCNN model จำเป็นต้องใช้ dependencies ขนาดใหญ่หลายตัว เช่น tensorflow นอกจากนี้ยังต้อง load pretrained weight จาก coco model ส่งผลให้ทรัพยากรไม่เพียงพอในการใช้ instance t2.micro หรือ t3.micro ที่ free tier รองรับได้ โดยในโปรเจกต์นี้ได้ใช้ EC2 instance ขนาด t2.xlarge(มีการ schedule ช่วงเวลาที่จะรันใน 1 วัน ทำให้ไม่ต้องรันตลอดเวลา)

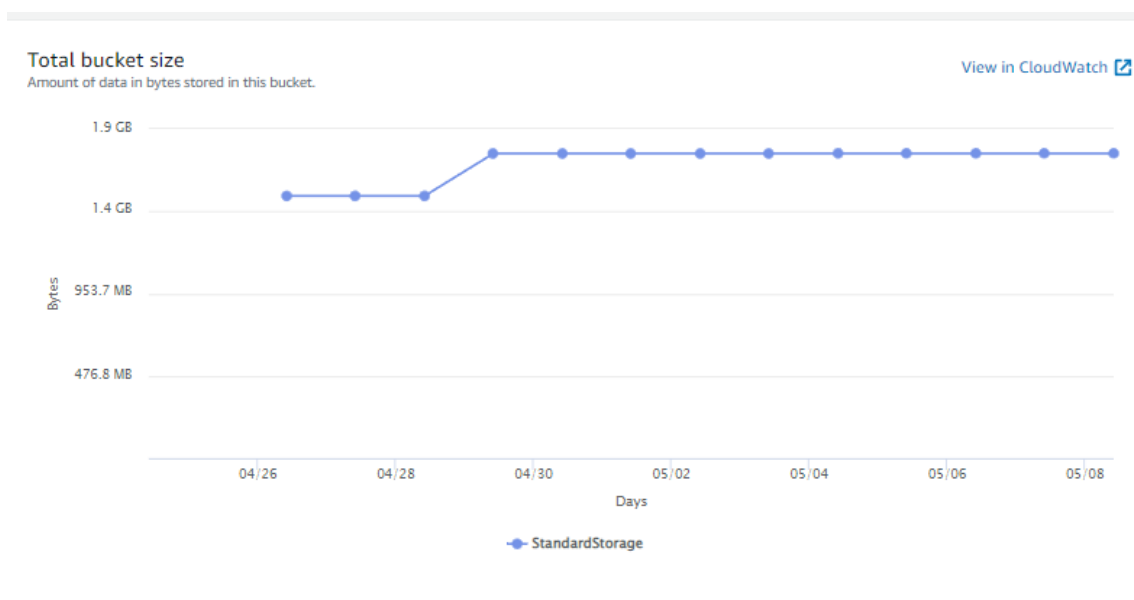
ชื่อ	vCPU	RAM (GiB)	CPU Credit/ ชม.	ราคาแบบตามต้องการ/ ชม.*	การใช้งานอินสแตนซ์แบบเหมาจ่ายรายชั่วโมงแบบคุ่มค่า เป็นระยะเวลา 1 ปี*
t2.nano	1	0.5	3	0.0058 USD	0.003 USD
t2.micro	1	1.0	6	0.0116 USD	0.007 USD
t2.small	1	2.0	12	0.023 USD	0.014 USD
t2.medium	2	4.0	24	0.0464 USD	0.031 USD
t2.large	2	8.0	36	0.0928 USD	0.055 USD
t2.xlarge	4	16.0	54	0.1856 USD	0.110 USD
t2.2xlarge	8	32.0	81	0.3712 USD	0.219 USD

FIGURE 36: t2 pricing for each CPU/RAM per hour in USD

3.3.4 AWS monitoring tools

3.3.4.1 Cloudwatch

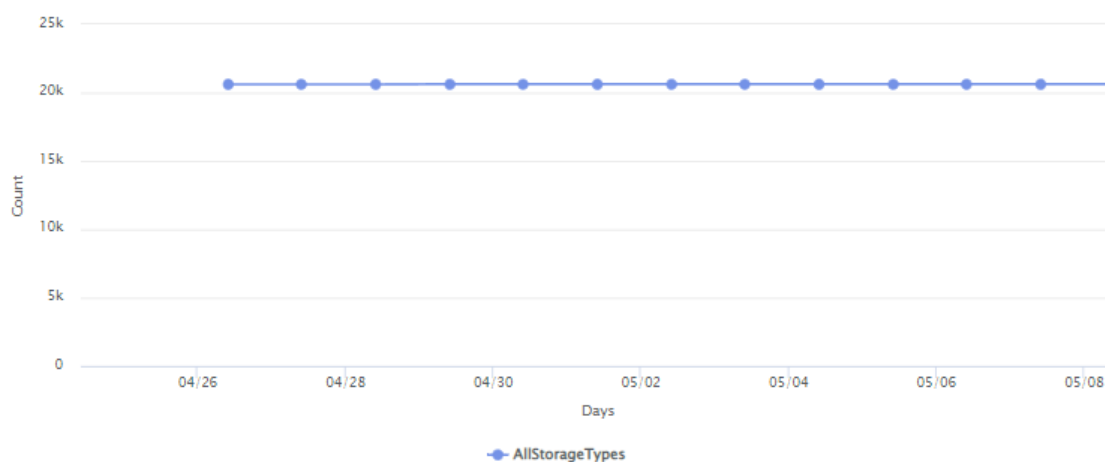
เป็นอุปกรณ์เสริมพื้นฐานของ AWS สำหรับการ monitor ปริมาณการใช้งาน services ต่างๆ โดยหลักๆแล้วโปรเจกต์นี้ได้ใช้อุปกรณ์ตัวนี้ในการ monitor S3 storage โดยทำการ monitor ปริมาณการใช้ storage, จำนวน object ใน storage และปริมาณการส่ง put request สำหรับส่งรูปภาพจากอุปกรณ์ IOT มาเก็บ



Total number of objects

Total number of objects stored in this bucket for all storage classes.

[View in CloudWatch](#)



All requests

All HTTP requests made to this bucket.

[View in CloudWatch](#)

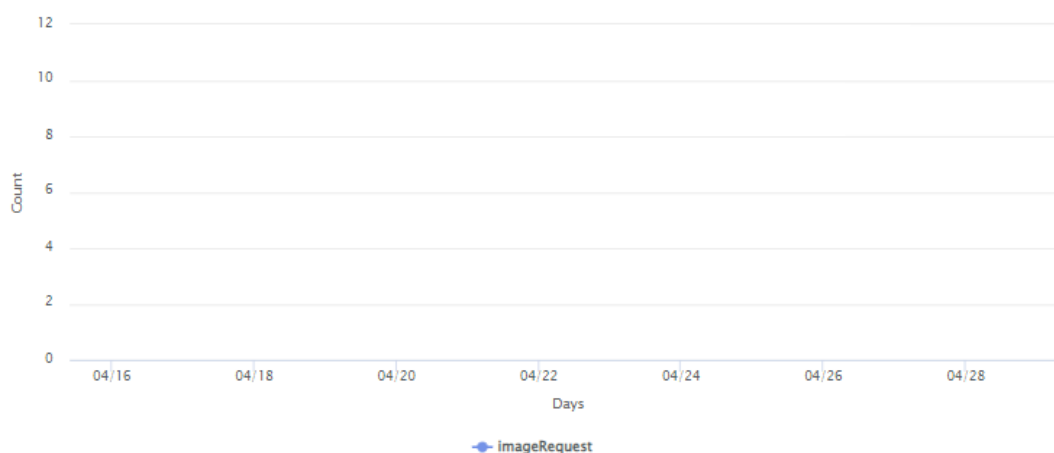


FIGURE 37: AWS Billing Dashboard

3.3.4.2 AWS billing dashboard

เป็นหนึ่งในอุปกรณ์เสริมพื้นฐานของ AWS สำหรับการ monitor ค่าใช้จ่ายจาก services ต่างๆ รวมถึง monitor ปริมาณการใช้ทรัพยากรจาก service ต่างๆว่าเกินขอบเขตของ free tier หรือไม่ โดยใน project นี้จะโฟกัสไปที่การ monitor S3 storage และ EC2 ซึ่งมีแนวโน้มในการสร้างค่าใช้จ่ายมากที่สุด

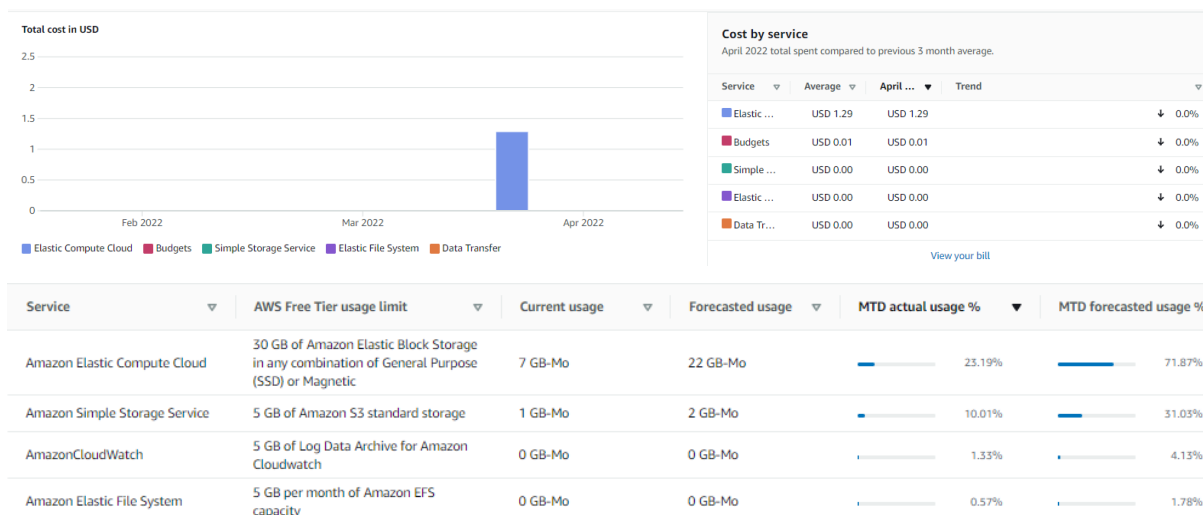


FIGURE 38: AWS can monitor each service and control price accordingly

นอกจากนี้ อุปกรณ์ตัวนี้ยังมีฟีเจอร์ Budget สำหรับวางแผนการใช้จ่ายในกรณีที่มีการใช้งานทรัพยากรเกินขอบเขตของ free tier โดยในโปรเจกต์นี้ได้ตั้ง budget ให้ S3 storage และ EC2 ในการเก็บค่าใช้จ่ายเพิ่มเติมเดือนละไม่เกิน 10 ดอลลาร์ โดยจะมีการแจ้งเตือนบิลสรุปผลค่าใช้จ่ายในทุกๆเดือนด้วย

Date: April 2022 [Download CSV](#) [Print](#)

Total	\$1.30
Amazon Web Services, Inc. - Service Charges	\$1.30
Payment Summary	
Payment Summary	
+ Expand All	
AWS Service Charges	\$1.30
Budgets	\$0.01
CloudWatch	\$0.00
Data Transfer	\$0.00
Elastic Compute Cloud	\$1.21
Elastic File System	\$0.00
Key Management Service	\$0.00
Lambda	\$0.00
Simple Storage Service	\$0.00
Taxes	
VAT to be collected	\$0.08

FIGURE 39: AWS also has a feature budget to plan resource usage

บทที่ 4: ผลลัพธ์การทำงาน

4.1 Machine Learning Model

การทดลอง มีการใช้วิธีวัด accuracy หลายรูปแบบ โดยเริ่มจาก mean average precision หรือ map หลังจากนั้น มีการเพิ่ม precision และ recall

ขั้นตอนแรก ได้แก่ การหาค่า Intersection over Union (IoU) หรือ Jaccard Index คำนี้อาจได้จาก บริเวณ overlap ระหว่าง prediction และ ground truth / บริเวณ union ของ prediction และ ground truth หากมีค่า IoU สูงกว่าค่าๆหนึ่ง จะจัดว่าเป็น True Positive ไม่เช่นนั้นจัดว่าเป็น False Positive หาก ground truth หนึ่ง มีหลาย detection แล้ว detection ที่มี IoU สูงที่สุดจะถูกเลือกเป็น True Positive

หลังจากนั้น ทำการหา Precision และ Recall ของทุก detection โดยที่ Precision = True positive / (True positive + False positive)

และ Recall = True positive / (True positive + False negative)

ค่า Precision และ Recall เหล่านี้ ถูกนำมา plot เป็น PR หรือ Precision Recall curve มีพื้นที่คือค่า AP โดย mAP หาได้จากค่าเฉลี่ยของ AP ระหว่างทุก object category^[2]

	Validation (mAP)	Testing (mAP)
การทดลองครั้งที่ 1	0.5740	0.4935
การทดลองครั้งที่ 2	0.7089	0.4936
การทดลองครั้งที่ 3	0.6014	0.5248
การทดลองครั้งที่ 4	0.7055	0.5235
การทดลองครั้งที่ 5	0.7055	0.5089
การทดลองครั้งที่ 6	0.6323	0.5084
การทดลองครั้งที่ 7	0.6201	0.5044

การทดลองครั้งที่ 1

Training 96 รูป 605 instance Validate 11 รูป 121 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.5740

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.4935

การทดลองครั้งที่ 2

Training 103 รูป 723 instance Validate 13 รูป 130 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.7089

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.4936

การทดลองครั้งที่ 3

Training 150 รูป 925 instance Validate 18 รูป 171 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.6014

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.5248

การทดลองครั้งที่ 4

Training 218 รูป 1413 instance Validate 27 รูป 235 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.7055

ผลของการทดลอง precision ของ validation set อยู่ที่ 0.9403

ผลของการทดลอง recall ของ validation set อยู่ที่ 0.7186

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.5235

ผลของการทดลอง precision ของ test set อยู่ที่ 0.9750

ผลของการทดลอง recall ของ test set อยู่ที่ 0.5280

การทดลองครั้งที่ 5 เพิ่มขนาด Test Set

Training 218 รูป 1413 instance Validate 27 รูป 235 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.7055

ผลของการทดลอง precision ของ validation set อยู่ที่ 0.9403

ผลของการทดลอง recall ของ validation set อยู่ที่ 0.7186

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.5089

ผลของการทดลอง precision ของ test set อยู่ที่ 0.8722

ผลของการทดลอง recall ของ test set อยู่ที่ 0.5191

การทดลองครั้งที่ 6 เปลี่ยนการแบ่ง Train/Valid เป็น 70%/30% จาก 90%/10%

Training 210 รูป 1378 instance Validate 35 รูป 270 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.6323

ผลของการทดลอง precision ของ validation set อยู่ที่ 0.8663

ผลของการทดลอง recall ของ validation set อยู่ที่ 0.6350

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.5084

ผลของการทดลอง precision ของ test set อยู่ที่ 0.8538

ผลของการทดลอง recall ของ test set อยู่ที่ 0.5138

การทดลองครั้งที่ 7 เพิ่มการ train เป็น 10 epoch

Training 210 รูป 1378 instance Validate 35 รูป 270 instance

ผลของการทดลอง mAP ของ validation set อยู่ที่ 0.6201

ผลของการทดลอง precision ของ validation set อยู่ที่ 0.8780

ผลของการทดลอง recall ของ validation set อยู่ที่ 0.6282

ผลของการทดลอง mAP ของ test set อยู่ที่ 0.5044

ผลของการทดลอง precision ของ test set อยู่ที่ 0.8153

ผลของการทดลอง recall ของ test set อยู่ที่ 0.5127

```
Counted number of mangos: 8
Correct amount of mangos: 8
Correction percentage: 100.0
precision: [1. 1. 1. 1. 1. 1. 1. 1. 0.]
recall: [0. 0.125 0.25 0.375 0.5 0.625 0.75 0.875 1. ]
precision: 1.0
recall: 1.0
mAP: 1.0
```

FIGURE 40: Experiment 7th result

4.2 ระบบ IOT

ได้ผลลัพธ์เป็น ระบบ IOT ที่สามารถ ถ่ายรูปต้นมะม่วงที่ ฟาร์มมะม่วง แล้วส่งรูปขึ้น AWS ได้ โดยไม่ต้องต่อสายไฟ
ในระหว่างการทำงาน (Self sustainable) สามารถทำงานได้เป็นระยะเวลานาน (5-8 ชม.)



FIGURE 41: Picture of system when collecting data from mango farm

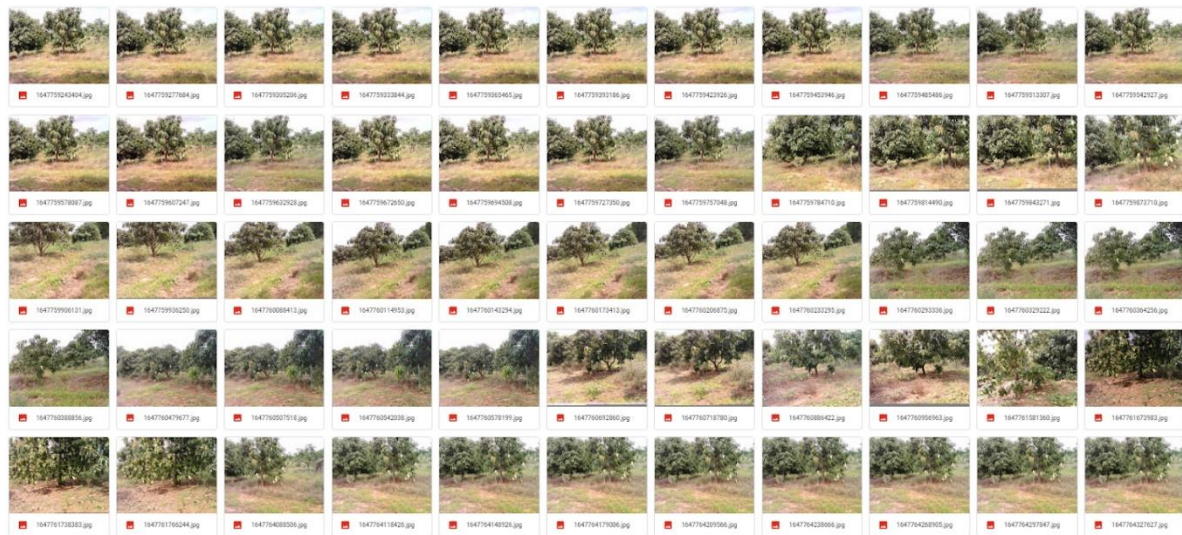


FIGURE 42: Sample picture that are collected from the field

4.3 Cloud Application

ผลลัพธ์สุดท้ายที่ได้เป็นหน้าเว็บของ Power BI ที่นำเสนอข้อมูล ได้แก่

1. วันเวลาสุดท้าย ที่ทำการถ่ายรูปและนับผลมะม่วง รวมถึงจำนวนผลมะม่วงของวันนั้น
2. รูปต้นมะม่วงหลังทำการ mask ผลมะม่วงด้วย Mask R-CNN
3. Dashboard แสดงจำนวนมะม่วงบนต้นในช่วงเวลาต่างๆ



FIGURE 43: Sample picture that are collected from the field

โดยหน้าเว็บดังกล่าวจะมีการอัปเดตข้อมูลมะม่วงในทุกๆวัน เช่น หากตั้งค่าให้ระบบ IOT มี schedule การถ่ายรูปมะม่วงตอนเที่ยงวัน ข้อมูลจะถูกส่งไปเก็บยัง S3 storage ผ่านขั้นตอนการใช้ EC2 instance นับมะม่วงและส่งผลลัพธ์ไปยัง

Google sheet เพื่อให้ Power Bi ดึงมาแสดงผลแล้ว โดยรวมจะมี delay ประมาณ 2 ชั่วโมง แปลว่าหน้าเว็บนี้จะอัปเดตข้อมูลทุกวัน โดย delay หลังจากการถ่ายรูปของระบบ IOT ประมาณ 2 ชั่วโมง

4.3.1 ขีดจำกัดของ application

1. สมมติรูปภาพมีขนาด 120Kb S3 storage จะสามารถรองรับรูปภาพได้ 41,666 รูป ซึ่งหากกล้อง 1 ตัวถ่ายรูปวันละ 5 รูป หมายความว่า storage free tier นี้เพียงพอต่อการใช้งาน 8,333วัน/จำนวนกล้องที่ติดตั้งในสวน โดยสามารถกำหนดเวลาหมดอายุของรูปภาพ storage เพื่อลดจำนวนการใช้พื้นที่ในระยะยาวได้
2. ใน 1 เดือนจำเป็นต้องส่ง put request ไปยัง S3 storage เพื่อเก็บรูปภาพอย่างน้อยจำนวน 150 ครั้ง(วันละ 5 ครั้ง x 30 วัน)xจำนวนกล้องที่ติดตั้งในสวน
3. มี cost การรัน application ใน 1 วันขั้นต่ำ = 0.0155 USD หรือ 0.465 USD ใน 1 เดือน โดยมี cost หลักมาจากการรัน EC2 instance + ค่าใช้จ่ายในกรณีใช้ storage หรือจำนวน request มากกว่าที่ free tier มีให้

บทที่5: timeline การทำงาน

ภาคเรียนที่ 1

รายละเอียด	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ปรึกษากับสมาชิกในกลุ่ม และอาจารย์ที่ปรึกษาเพื่อกำหนดหัวข้อของงานที่จะทำ																
2. รวบรวมเทคนิคการนับมะม่วงด้วย machine learning model ต่างๆ																
3. รวบรวมข้อมูลงานวิจัย เพื่อเปรียบเทียบประสิทธิภาพของ machine learning model ที่หาข้อมูลมา																
4. ศึกษาพื้นฐานของ deep learning และ image processing ผ่านสื่อการสอนต่างๆ เช่น คลิปวิดีโอบนยูทูป																
5. ศึกษาข้อมูลของอุปกรณ์ที่ใช้ในการทำระบบ IOT สำหรับเก็บรูปภาพจากสวนผลไม้ต่างๆ																
6. ศึกษาหลักการการทำงานของ model CNN และ mask RCNN โดยละเอียด + ลองรับ model จริงโดยทำตาม tutorial บนอินเทอร์เน็ต																
7. แบ่งหน้าที่ของสมาชิกในกลุ่มอย่างชัดเจนขึ้น โดยแบ่งกันไปศึกษาหัวข้อที่ตนรับผิดชอบ ได้แก่ mask RCNN, IOT และ cloud application																
8. ลองใช้ model mask RCNN เพื่อ detect ผลมะม่วงโดยใช้รูปภาพเว็บไชท์ของสวนมะม่วง																
9. ศึกษาข้อมูลเกี่ยวกับอุปกรณ์ในการทำ cloud application ในการประมวลผลและนำเสนอข้อมูล รวมถึงระบบสำหรับเชื่อมต่อข้อมูลกับอุปกรณ์ IOT																
10. ศึกษาข้อมูลการใช้ Jetson Nano และส่งข้อมูลอุปกรณ์เพื่อลองประกอบ																
11. จัดทำ proposal report																
12. เตรียมตัวนำเสนอ project แก่กรรมการ																
13. ปรับปรุงและแก้ไข proposal report ตามที่กรรมการเสนอ																

FIGURE 44: Timeline of the 1st semester

ภาคเรียนที่ 2

Timeline การทำงาน	ธันวาคม				มกราคม				กุมภาพันธ์				มีนาคม				เมษายน				พฤษภาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาการใช้เทคนิค tranfer learning และ fine tuning เพิ่มเติม+ลองใช้ library mask RCNN อื่น																								
2. หาข้อมูลเกี่ยวกับการติดตั้งระบบ IOT เพื่อเก็บรูปภาพในสวนผลไม้เพิ่มเติม																								
3. ทำ data augmentation + ใช้ evaluation tool + ลองใช้เทคนิคต่างๆเพื่อเพิ่มความแม่นยำในการนับมะม่วงของ model																								
4. ปรับปรุง design ของระบบ IOT โดยลองลดอุปกรณ์ต่างๆที่ไม่จำเป็น																								
5. สร้าง cloud storage และลองรับรูปและข้อความที่ส่งจากระบบ IOT ไปเก็บบน S3 storage																								
6. ทดสอบส่งรูปขึ้น cloud storage + ปรับ resolution ของรูปให้เหมาะสมแก่การนำไปใช้ train model																								
7. ออกแบบ Outdoor packaging สำหรับนำไปติดตั้งที่สวน																								
8. ใช้ monitor tools เพื่อตรวจสอบพื้นที่เก็บข้อมูลและค่าใช้จ่ายของ cloud storage																								
9. ประกอบรวม packaging ของระบบ IOT + ทดสอบ																								
10. ไปที่สวนเพื่อเก็บรูปมะม่วง + ทดลองนำอุปกรณ์ IOT ไปติดตั้ง																								
11. ใช้ dataset ที่เก็บจากสวนมะม่วงในการ train machine learning model และปรับปรุงความแม่นยำของ model ในการ detect จำนวนมะม่วงจากรูปเหล่านั้น																								
12. Deploy Mask RCNN model บน AWS EC2 และลองดาวน์โหลดรูปจาก S3 storage เพื่อใช้ model detect ผลลัพธ์จำนวนมะม่วง																								
13. ส่งข้อมูลการ detect ผลลัพธ์จำนวนมะม่วงจาก EC2 instance ไปยัง Google sheet และตั้งค่า schedule สำหรับช่วงเวลาที่จะรัน EC2 instance ใน 1 วันเพื่อลดค่าใช้จ่าย																								
14. เชื่อมต่อ Google sheet กับ Power BI และออกแบบ interface ของเพื่อสรุปผลและนำเสนอผลลัพธ์จำนวนมะม่วงในช่วงเวลาต่างๆ																								
15. จัดทำรายงานเพื่อสรุปผล																								

FIGURE 45: Timeline of the 2st semester

ประเภทของงาน: แดง=IOT, เขียว=Machine Learning, ฟ้า=Cloud application, เทา=รายงาน, ม่วง=ลงภาคสนาม

บทที่ 6: สรุปผลการทำงาน

6.1 สรุปผลลัพธ์และสิ่งที่ต้องกังวลเกี่ยวกับ Machine Learning Model

6.1.1 สรุปผลลัพธ์การทำงานของ Machine Learning Model

ได้ผลลัพธ์เป็น Mask RCNN model สำหรับนับจำนวนมะม่วงบนต้นจากรูปภาพ โดยมีวิธีวัดความแม่นยำ 3 รูปแบบ ได้แก่ Mean Average Precision (mAP), Precision, และ Recall โดยที่ผลสุดท้าย ที่ดีที่สุด มี mAP อยู่ที่ 0.7055 สำหรับ Validation และ 0.5235

ผลลัพธ์	Training	Testing
mAP	0.7055	0.5235
precision	0.9403	0.9750
recall	0.7186	0.5280

6.1.2 สิ่งที่ต้องกังวลเกี่ยวกับ Machine Learning Model

การใช้โมเดล detect รูปมะม่วง อาจไม่สามารถตรวจจับมะม่วงที่มีขนาดเล็ก หรือถูกบัง ทำให้การวัด ได้มะม่วงไม่ครบ

6.2 สรุปผลลัพธ์และสิ่งที่ต้องกังวลเกี่ยวกับระบบ IOT

6.2.1 สรุปผลลัพธ์การทำงานของระบบ IOT

1. ระบบสามารถทำงานได้ตาม วัตถุประสงค์ที่ตั้งไว้ สามารถเก็บข้อมูล อุณหภูมิ ความชื้นและรูปภาพมะม่วง ส่งขึ้น AWS Cloud storage ผ่าน wifi ได้ รวมถึงมีการใช้พลังงานหมุนเวียนจากแสงอาทิตย์ ทำให้ระบบสามารถใช้งานต่อเนื่องได้เป็นเวลานานโดยไม่ต้องมีการลากต่อสายไฟไปยังแหล่งจ่ายไฟ
2. ตัวอุปกรณ์ จะทำการถ่ายรูป และส่งข้อมูลขึ้น Cloud ทุกๆ 1 นาที และสามารถทำงานได้นานถึง 7-8 ชม (ในวันทดสอบ ระบบสามารถทำงานต่อเนื่องได้ตั้งแต่ 10 โมงเช้าถึงบ่าย 4 เย็น)
3. ตัวรูปภาพมีความละเอียดมากพอที่จะ ใช้ประมวลผลผ่าน Model ของเราได้ (5 Megapixel)

6.2.2 สิ่งที่ต้องกังวลเกี่ยวกับระบบ IOT

1. หากมีแสงสว่างใน environment ที่ทำการถ่ายรูปมากเกินไป จะส่งผลให้ภาพที่กล้องถ่ายจะมีรายละเอียดสูงขึ้นตามไปด้วย ซึ่งอาจทำให้เกิดปัญหาตอนเก็บเซฟข้อมูลภาพผ่าน arduino คือรูปภาพใหญ่เกินไปจนประมวลผลไม่ทัน เกิดเป็นแถบ noise บนรูปภาพ
2. ในกรณีที่ ต้องการ scale อุปกรณ์ให้มีขนาดใหญ่ขึ้น รองรับได้หลายๆ กล้อง อาจต้องใช้สายไฟคนละแบบกับที่ใช้ ไม่นับมีสิทธิสัญญาณขาดหายระหว่างทาง

6.3 สรุปผลลัพธ์และสิ่งที่ต้องกังวลเกี่ยวกับ Cloud application

6.3.1 สรุปผลลัพธ์การทำงานของระบบ Cloud application

สามารถสร้าง application platform สำหรับเป็น pipeline ในการเชื่อมทั้ง 3 parts ของโปรเจกต์เข้าด้วยกัน โดยมีขั้นตอน ได้แก่ การรับรูปภาพมะม่วงที่ส่งจากระบบ IOT ไปเก็บยัง AWS S3 storage จากนั้นจึง รัน Machine learning model บน Ec2 instance เพื่อนับจำนวนมะม่วงจากรูปที่เก็บมา จากนั้นจึงส่งจำนวนมะม่วงที่นับได้ไปเก็บยัง Google sheet ที่ผูกกับ Power BI เพื่อให้สามารถใช้อุปกรณ์เพื่อดูข้อมูลนำไป visualize เพื่อให้เกษตรกรนำไปใช้งานต่อไป

Application ที่สำหรับให้เกษตรกรใช้งานจะอยู่ในรูปของ link url ที่ถูก published ให้ใครก็ตามที่มี link สามารถเข้าไปชม application ได้ โดยข้อมูลมะม่วงจะถูกอัปเดตเป็นรายวัน

6.3.2 สิ่งที่ต้องกังวลเกี่ยวกับระบบ Cloud application

การใช้จำนวน storage หรือส่ง request สำหรับเก็บรูปมากเกินไปอาจทำให้มีค่าใช้จ่ายจาก AWS มากขึ้นได้ โดยเหตุการณ์ดังกล่าวอาจเกิดขึ้นได้หากมีจำนวนกล้องที่ติดตั้งที่สวนมะม่วงเพิ่มมากขึ้น

6.4 ผลกระทบต่อสังคม

ระบบติดตามผลผลิตทางการเกษตร จะช่วยให้เกษตรกรสามารถติดตามและวางแผนการเก็บเกี่ยวผลผลิตได้อย่างมีประสิทธิภาพ เพื่อลดความสูญเสียจากผลผลิตเน่าเสีย และเพิ่มโอกาสเก็บเกี่ยวผลไม้นอกฤดูกาลให้แก่เกษตรกร โดยจะมีการออกแบบระบบแสดงผลผ่าน web application ให้มีความเข้าใจง่าย เพื่อให้เกษตรกรสามารถนำข้อมูลไปใช้ได้ง่าย นอกจากนี้ตัวระบบยังออกแบบให้ชุดอุปกรณ์ที่ติดตั้งในภาคสนาม สามารถครอบคลุมพื้นที่การทำงานที่ต้องการได้ โดยใช้จำนวนชุดอุปกรณ์จำนวนน้อยที่สุด เพื่อช่วยลดต้นทุนการติดตั้งอุปกรณ์ของเกษตรกร

บทที่ 7: ปัญหาที่พบ

7.1 Machine Learning Model

เมื่อเพิ่มขนาด dataset พร้อมทั้งจำนวน Region of Interest ทำให้ RAM ของ Google Colab ไม่พอ ทำให้ machine learning ไม่สำเร็จ แก้ไขโดยการลด Images Per GPU จาก 4 รูป เหลือ 2 รูป แต่ เมื่อเพิ่มขนาด dataset อีก พบ ปัญหา แก้ไขโดยการ ใช้ Google Colab Pro

7.2 ระบบ IOT

1. เรามีปัญหาตอน Design และ implement วงจรหลายอย่างมาก ตั้งแต่ wiring ผิดทำให้อุปกรณ์เสียหาย ตัวจ่ายไฟ ไม่ตก spec ต่อ แผง solar cell ผิดรูปแบบ ทำให้เสียเวลาในการต่ออุปกรณ์เป็นเวลานาน
2. ปัญหาที่พบ ณ ตอนเอาอุปกรณ์ลงภาคสนามคือ ตัวกล้อง เนื่องจากว่า กำลังถ่ายม้วนมาก สู้ที่ 5 MP พอโดนแสงแดด ที่แรงมากๆ เหมือนมันจะเพิ่มปริมาณ KB ต่อรูปภาพ (จากเดิม 60-70 KB ไปเป็น 1-2 MB) ทำให้ไม่สามารถ

7.3 Cloud application

ปัญหาหลักที่พบคือ การที่ Mask RCNN model มีขนาดของ model และ dependencies ใหญ่กว่าที่คิดไว้มาก เนื่องจากต้องใช้ library เสริมต่างๆจำนวนมาก เช่น tensorflow, keras, numpy นอกจากนี้ ในการรันยังต้อง load pretrained model ขนาดใหญ่ ทำให้มีความต้องการใช้ storage และ RAM จำนวนมาก ซึ่งไม่เพียงพอทั้งกับ Lambda function และ instance t2.micro หรือ t3.micro ที่เป็น free tier ทำให้ต้องเสียค่าใช้จ่ายซื้อ t2.xlarge เพื่อเพิ่มปริมาณทรัพยากรใน instance ด้วยเหตุนี้จึงต้องมีการ monitor การใช้ instance ตัวนี้ตลอดเวลาเพื่อป้องกันค่าใช้จ่ายเพิ่มเติม

บทที่ 8: สิ่งที่จะทำต่อไป

8.1 Machine Learning Model

พัฒนา model ให้สามารถจำแนก class ของมะม่วงได้อย่างหลากหลายมากขึ้น เช่น แบ่งคลาสมะม่วงในความสุขระดับต่างๆ โดยจะต้องมีการเก็บ dataset ของมะม่วงเพิ่มในความสุขที่หลากหลายขึ้น เพื่อนำมา train model ให้สามารถทำได้ทั้งการนับจำนวนมะม่วง และ detect ระดับความสุข

8.2 ระบบ IOT

เราสามารถ Duplicate แค่นี้เฉพาะ ระบบกล้องแล้วเดิมเข้าไปยังระบบเดิมได้ เนื่องจาก เรามีกำลังไฟที่มากพอ จะได้ตั้งตัวกล้องไว้อย่างจัดต่างๆ ของ สวนมะม่วง และ ส่งข้อมูลเป็นหลายๆ ชุด + Scale ระบบได้

8.3 Cloud application

1. เปลี่ยนวิธีการ deploy machine learning model โดยใช้อุปกรณ์อื่นที่รองรับการใช้ทรัพยากรมากกว่านี้และมีราคาถูกกว่า EC2 โดยจะมีการสร้าง API Gateway และ Lambda function สำหรับรับ request เพื่อเข้าถึงรูปภาพใน S3 storage จากภายนอก AWS server
2. ปรับปรุง design ของ Power BI ให้สามารถ responsive ของขนาดหน้าจอโทรศัพท์มือถือและ tablet เพื่อให้เกษตรกรสามารถเข้าถึงได้ผ่านอุปกรณ์ที่หลากหลาย

References

- [1] University of toronto ,“Image Segmentation Using Deep Learning: A Survey”
<https://ieeexplore-ieee-org.chula.idm.oclc.org/document/9356353> (accessed : 8 Jan 2022)
- [2] Nick Zeng ,“An Introduction to Evaluation Metrics for Object Detection”
<https://blog.zenggyu.com/en/post/2018-12-16/an-introduction-to-evaluation-metrics-for-object-detection/> (accessed : 9 Jan 2022)
- [3] สวนมะม่วง ฟาร์มออ.เอช , “รูปสวนมะม่วง ฟาร์มออ.เอช”
<https://sites.google.com/view/sweetmango/farm-or-ace> (accessed : 6 Feb 2022)
- [4] สุทธิพร จิตต์มิตรภาพ, “เอกสารการปลูกมะม่วงน้ำดอกไม้สีทอง”
http://www.agi.nu.ac.th/postharvest/downloads/upload_file/281118140408_Mango.pdf (accessed : 14 Apr 2022)
- [5] คุณสุธินี โกกัลสิน, “ข้อมูลมะม่วงน้ำดอกไม้ เขียนโดยคุณสุธินี โกกัลสิน”
<http://www.khlongkhuean.com/%E0%B8%99%E0%B9%89%E0%B8%B3%E0%B8%94%E0%B8%AD%E0%B8%81%E0%B9%84%E0%B8%A1%E0%B9%89%E0%B8%AA%E0%B8%B5%E0%B8%97%E0%B8%AD%E0%B8%87/> (accessed : 20 Dec 2021)
- [6] pasupol bunsan, “React The Series : Intro To JSX”
<https://medium.com/niawjr/react-the-series-ep1-intro-to-jsx-a2e4b924864a> (accessed : 26 Dec 2021)
- [7] marcuscode, “การใช้งาน Express.js บน Node.js”
<http://marcuscode.com/tutorials/nodejs/using-expressjs> (accessed : 6 Feb 2022)
- [9] Soumya Yadav, “A simple guide to Mask R-CNN implementation on a custom dataset”
<https://medium.com/analytics-vidhya/a-simple-guide-to-maskrcnn-custom-dataset-implementation-27f7eab381f2> (accessed : 21 Feb 2022)
- [10] Dhruvil Shah, “Mask R-CNN implementation on custom dataset”
<https://towardsdatascience.com/mask-rcnn-implementation-on-a-custom-dataset-fd9a878123d4>
- [11] thebinarynotes , “How to train Mask R-CNN on custom dataset”
<https://thebinarynotes.com/how-to-train-mask-r-cnn-on-the-custom-dataset/> (accessed : 7 Jan 2022)

- [12] thebinarynotes, “Google Colab : Train deep learning models on the cloud”
<https://thebinarynotes.com/google-colab-train-deep-learning-models-without-gpu-machine/> (accessed : 3 Jan 2022)
- [13] Yasar Niyazoglu AL, “Mask R-CNN | Custom Dataset - Görüntü Segmentasyonu!”
https://www.youtube.com/watch?v=ymnxng_SB_c (accessed : 11 Jan 2022)
- [14] Vijendra Singh, “Custom Mask RCNN using Tensorflow Object Detection API” Custom Mask RCNN using Tensorflow Object Detection API | by Vijendra Singh | Medium (accessed : 3 Mar 2022)
- [15] RomRoc , “Mask R-CNN google collab”
https://colab.research.google.com/github/RomRoc/maskrcnn_train_tensorflow_colab/blob/master/maskrcnn_custom_tf_colab.ipynb (accessed : 4 Mar 2022)
- [16] Adrian Rosebrock, “Fine-tuning”
<https://www.pyimagesearch.com/2019/06/03/fine-tuning-with-keras-and-deep-learning/> (accessed : 19 Feb 2022)
- [17] hello@paperswithcode.com, “Coco test-dev for instance segmentation task”
<https://paperswithcode.com/sota/instance-segmentation-on-coco> (accessed : 19 Feb 2022)
- [18] hello@paperswithcode.com, “Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning”
<https://www.frontiersin.org/articles/10.3389/fpls.2020.571299/full> (accessed : 31 Mar 2022)
- [19] Elisha Odemakinde, “Convolutional Neural Network”
<https://viso.ai/deep-learning/mask-r-cnn/> (accessed : 4 Apr 2022)
- [20] Soumya Yadav, “A simple guide to Mask R-CNN implementation on a custom dataset”
<https://medium.com/analytics-vidhya/a-simple-guide-to-maskrcnn-custom-dataset-implementation-27f7eab381f2> (accessed : 13 Apr 2022)
- [21] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos, “Image Segmentation Using Deep Learning : A Survey”
<https://arxiv.org/pdf/2001.05566.pdf> (accessed : 26 Apr 2022)

[22] DINOQ Co.,Ltd, “ทำความรู้จัก Cloud platform เพื่อเลือกใช้งาน”

<https://dinoq.medium.com/%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81%E0%B8%84%E0%B8%A5%E0%B8%B2%E0%B8%A7%E0%B8%94%E0%B9%8C%E0%B9%80%E0%B8%9E%E0%B8%A5%E0%B8%95%E0%B8%9F%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%A1%E0%B9%80%E0%B8%9E%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B9%80%E0%B8%A5%E0%B8%B7%E0%B8%AD%E0%B8%81%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B8%AB%E0%B8%99%E0%B9%88%E0%B8%AD%E0%B8%A2%E0%B8%94%E0%B8%B5%E0%B8%81%E0%B8%A7%E0%B9%88%E0%B8%B2-dcfbae93ac11> (accessed : 6 Mar 2022)

[23]

Valeriy Ilchenko, “Serverless vs microservices architecture”

<https://www.byteant.com/blog/serverless-vs-microservices-architecture-what-does-the-future-of-business-computing-look/> (accessed : 17 Mar 2022)

[24] Amazon Web Services, Inc, “amazon aws service all free tier”

https://aws.amazon.com/th/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all (accessed : 19 Mar 2022)

[25] Sergio Canu, “Mask RCNN Colab tutorial and source code”

<https://pysource.com/2021/08/10/train-mask-r-cnn-for-image-segmentation-online-free-gpu/> (accessed : 21 Mar 2022)