

รายงาน

เรื่อง Easy Smile Fridge

จัดทำโดย

นางสาว ชนากัทร แซ่ตซิ่ง	EGCO	6013112
นางสาว พรพิชชา ทองมูล	EGCO	6013121
นางสาว วีไลลักษณ์ ขอประเสริฐ	EGCO	6013125
นาย ศุภเชษฐ์ อรุณสุขสิริ	EGCO	6013127
นาย สุภศักดิ์ เพرمกุศลชัย	EGCO	6013129
นางสาว สุกี้สรา พนาถิกุล	EGCO	6013300
นางสาว สุภากร จ้วนประเสริฐ	EGCO	6013301

เสนอ

อาจารย์ ดร.นริศ หนูหอม

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา EGCO486 Image Processing

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

ภาคเรียนที่ 1 ปีการศึกษา 2563

## คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของวิชา EGCO486 Image Processing ซึ่งได้อธิบายเกี่ยวกับทฤษฎีเบื้องต้นของการทำ Ingredients Recognition หรือการจำแนกส่วนประกอบในการทำอาหาร โดยอาศัยการทำ Deep Learning โดยใช้ model Yolo 3 รวมถึงอธิบายขั้นตอนการทำงานและผลที่ได้จากการทำ Deep learning

ผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานเล่มนี้จะมีประโยชน์สำหรับผู้ที่สนใจ หากรายงานเล่มนี้มีข้อผิดพลาดประการใดทางผู้จัดทำขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

# สารบัญ

คำนำ	๗
สารบัญ	๙
Dataset	๑
Data Preparation	๔
Method	๗
- Model YOLO V3	
Results	๙
Problem	๑๓
GUI	๑๔
Apendix	๒๕

## Dataset

ส่วนประกอบในการทำอาหาร หมายถึง ส่วนผสมที่ใช้เป็นวัตถุดิบเพื่อประกอบเป็นผลิตภัณฑ์อาหาร ซึ่งการตรวจจับวัตถุดิบที่คนผู้จัดทำได้ทำนี้ ได้กำหนดวัตถุดิบที่จะทำการตรวจจับเป็นจำนวนทั้งหมด 20 ชนิด (20 Class) และมีจำนวนรูปทั้งหมด 1,559 รูป โดยในแต่ละรูปจะมีวัตถุดิบหลากหลายชนิดอยู่ในรูปเดียวกัน ซึ่งวัตถุดิบทั้งหมดจำนวน 20 ชนิด มีดังต่อไปนี้

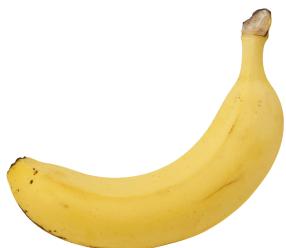
1. แอปเปิล (apple)



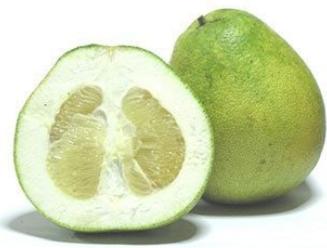
11. ข้าวโพด (corn)



2. กล้วย (banana)



12. ส้มโอ (grapefruit)



3. เนย (butter)



13. ดอกกะหลา (cauliflower)



4. นมกล่อง (milk)

14. สับปะรด (pineapple)



5. น้ำเปล่า (water)



6. มะนาว (lime)



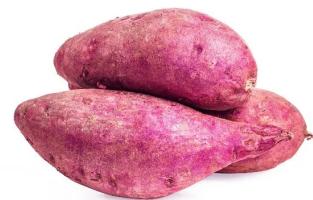
7. แตงกวา (cucumber)



8. แครอท (carrot)



15. มันฝรั่ง (potato)



16. แก้วมังกร (dragonfruit)



17. ขวดซอสมะเขือเทศ (ketchup)



18. ขนมปัง (bread)



9. เนื้อไก่ (chicken)



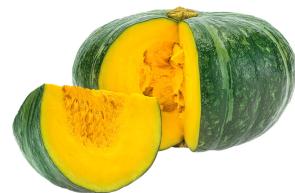
19. ผักกาด (cabbage)



10. ไข่ไก่ (egg)

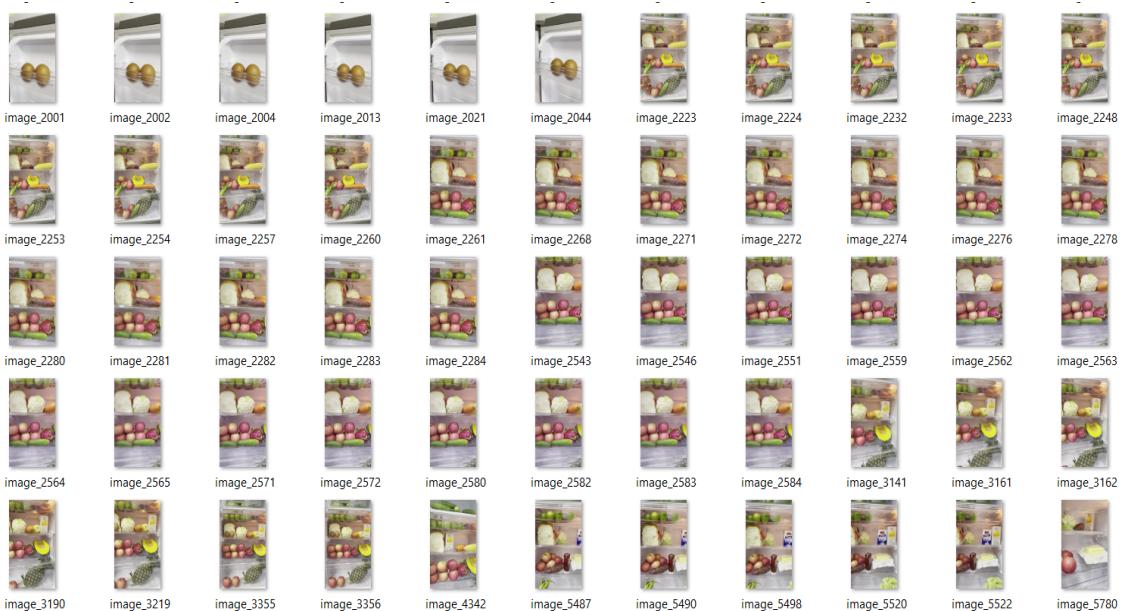


20. ฟักทอง (pumpkin)



## Data Preparation

1. นำส่วนประกอบในการทำอาหารวางไปจัดวางไว้ในตู้เย็น และทำการบันทึกวิดีโอ
2. นำวิดีโอที่ได้นั้นมาจับอุปกรณ์เป็นรูปภาพ frame ต่อ frame โดยใช้ฟังก์ชันของ opencv คือฟังก์ชัน cv2.VideoCapture()
3. หลังจากนั้นคัดเลือกรูปภาพเพื่อนำมา label โดยมีรูปภาพทั้งหมด 1,559 รูป โดยแบ่งรูปภาพไปเป็นชุดเรียนรู้ข้อมูลทั้งหมด 1,116 รูป และชุดของข้อมูลเพื่อนำไปทดสอบทั้งหมด 443 รูป



รูปที่ 1 แสดงภาพตัวอย่าง Dataset ที่ใช้

4. ทำการเตรียม Dataset สำหรับ YOLOv3 โดยทำการกำหนดวัตถุที่สนใจนั้นมีอะไรบ้างให้เขียนลงไว้ใน predefined\_classes.txt ในโฟลเดอร์ data ซึ่งในที่นี้เขียนลงไปทั้งหมด 20 class

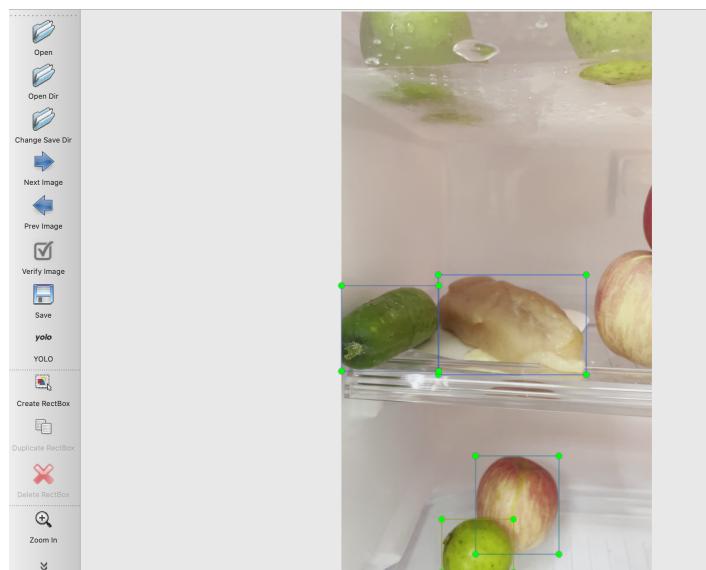
	egg
	corn
apple	grapefruit
banana	cauliflower
butter	pineapple
milk	potato
water	dragonfruit
lime	ketchup
cucumber	bread
carrot	cabbage
chicken	pumpkin

รูปที่ 2 แสดงชื่อของคลาสทั้งหมด

- นำรูปภาพทั้งหมดไปทำการ label เป็นการตีกรอบวัตถุดิบชนิดนั้นๆ พร้อมระบุชนิดโดยใช้โปรแกรม labelImg



รูปที่ 3 โปรแกรม LabelImg ที่ใช้นำไปตีกรอบวัตถุ



รูปที่ 4 แสดงภาพตัวอย่างในการตีกรอบวัตถุ

6. หลังจากนั้นจะได้ไฟล์ .txt ของแต่ละรูปโดยมีลักษณะดังรูปภาพด้านล่าง ซึ่งสิ่งที่ได้จากไฟล์คือ เลขคlassenของวัตถุนั้น, xmin, ymin, xmax, ymax



image\_12462 - Notepad  
File Edit Format View Help

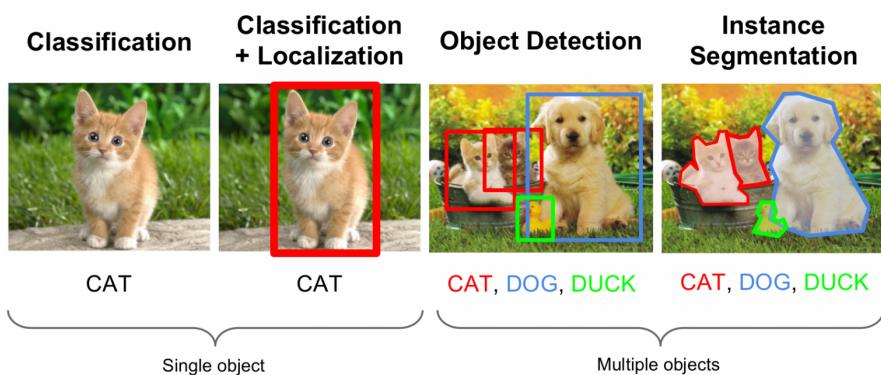
```
12 0.213889 0.550000 0.311111 0.158333
0 0.538889 0.053385 0.220370 0.105729
13 0.434259 0.281771 0.351852 0.163542
18 0.621296 0.548438 0.455556 0.296875
14 0.264815 0.723958 0.524074 0.093750
19 0.911111 0.625000 0.164815 0.277083
7 0.662037 0.320833 0.159259 0.089583
0 0.742593 0.057552 0.190741 0.114062
```

รูปที่ 5 แสดงภาพตัวอย่างผลลัพธ์ในการตีกรอบวัตถุ

## Method

### การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุ(Object Detection) คือ เทคโนโลยีในทางคอมพิวเตอร์ หลักการที่เกี่ยวกับ Computer Vision และ Image Processing ที่ใช้ในงาน AI ตรวจจับวัตถุชนิดที่กำหนด เช่น มนุษย์ รถยนต์ อาคาร ที่อยู่ในรูปภาพ หรือวิดีโอ ซึ่งการตรวจจับรูปภาพนั้นมีหลายแขนง เช่น การทำ Face Detection ตรวจจับหน้าคน Pedestrian Detection ตรวจจับคนเดินถนน สามารถประยุกต์ได้หลากหลาย เช่นการรักษาความปลอดภัย และอื่นๆ



รูปที่ 6 ภาพการตรวจจับวัตถุ

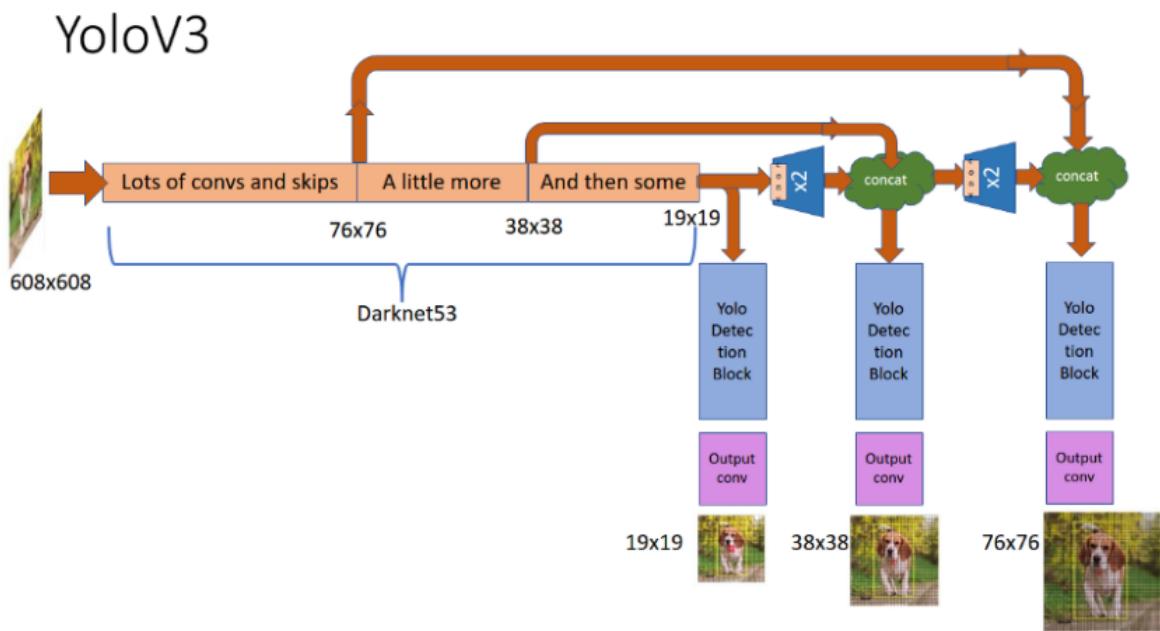
### Model YOLO V3

YOLO เป็น Framework สำหรับงานด้าน Object Detection และได้รับการพัฒนาโดยในโปรเจคนี้ได้นำโมเดล YOLOV3 มาใช้ในการทำ Object Detection ซึ่งเป็นโมเดลที่ที่ถูกพัฒนาด้วย TensorFlow โดย TensorFlow คือ deep learning library ของ Google โดยทาง Google ก็ได้ใช้ Machine Learning เพิ่มประสิทธิภาพกับผลิตภัณฑ์มากมาย ไม่ว่าจะเป็น เครื่องมือค้นหา (Search engine), การแปลภาษา (Translation), คำบรรยายภาพ (Image captioning) และ เครื่องมือช่วยการเสนอแนะ (Recommendations) เพื่อให้นักวิจัยและนักพัฒนาทำงานกับโมเดล AI ได้ ซึ่ง ในด้านสถาปัตยกรรมของ TensorFlow ได้แบ่งออกเป็น 3 ส่วน

- การเตรียมประมวลข้อมูล

- b. การสร้างแบบจำลอง
- c. ฝึกและประเมินแบบจำลอง

ข้อของ Tensorflow มาจาก การที่ Tensorflow รับข้อมูลเป็นอาร์เรย์หลายมิติ หรือที่เรียกว่า Tensors และเรามีหน้าที่จัดเรียงลำดับการประมวลผลเป็น Flowchart (หรือที่เรียกว่ากราฟ) ข้อมูลที่ถูกป้อนไป ก็จะผ่าน (Flow) กระบวนการจนออกมาเป็นผลลัพธ์ หรือ เอาท์พุท Tensorflow สามารถทำงานได้หลายแพลตฟอร์ม เช่น คอมพิวเตอร์ตั้งโต๊ะ ที่ใช้ระบบปฏิบัติการ Windows, macOS or Linux และ คลาวด์ หรือเว็บเซอร์วิซ และมือถือทั้งระบบ iOS and Android ทั้งนี้สามารถฝึก Tensorflow ได้จากหลากหลายอุปกรณ์ เมื่อเรามีโมเดลจำลองการฝึกแล้ว (Train Model)

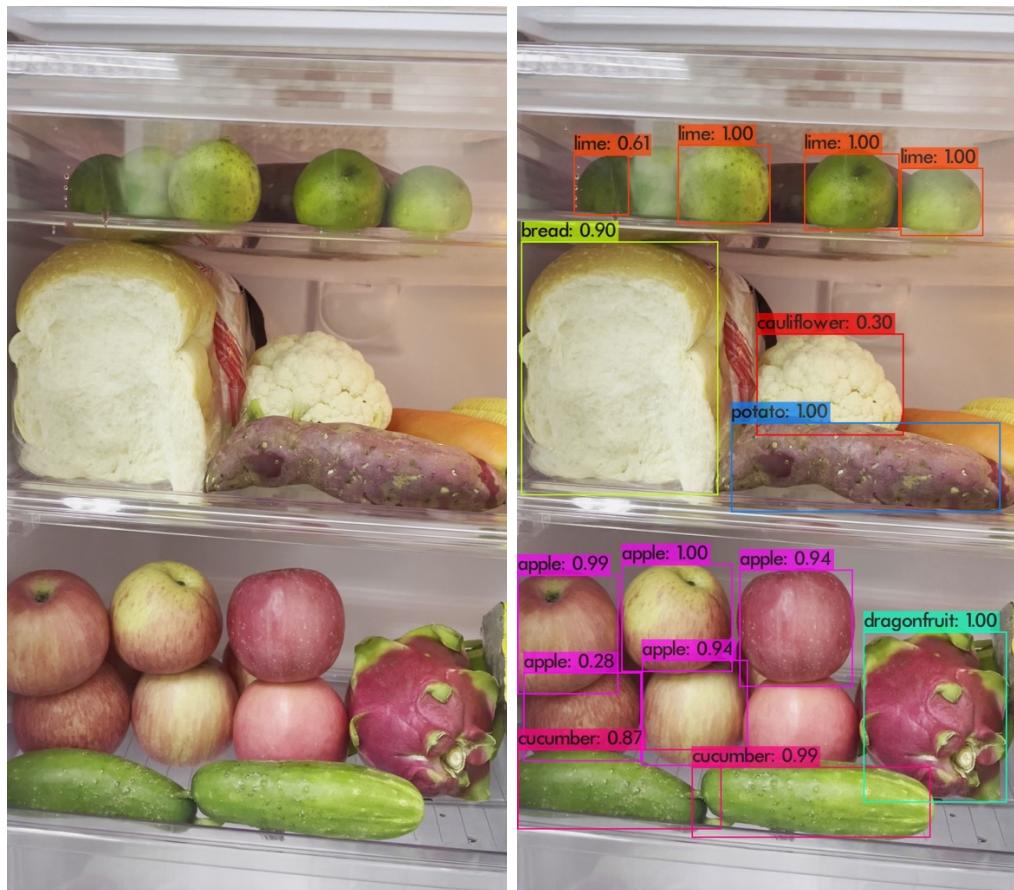


รูปที่ 7 ภาพแสดงสถาปัตยกรรมของระบบ YOLOv3

ได้รับแรงบันดาลใจจากสถาปัตยกรรม ResNet และ FPN (Feature-Pyramid Network) ตัวแยกคุณลักษณะ YOLO-V3 เรียกว่า Darknet-53 (มี 52 convolutions) โดยข้อดีของ YOLOv3 นั้นคือสามารถตรวจจับวัตถุที่มีขนาดเล็กได้ดี

## Results

ในการทดสอบโมเดลการตรวจจับและทำนายวัตถุดิบจากรูปภาพที่ถ่ายจากโทรศัพท์ โดยผู้จัดทำได้ทำการถ่ายภาพวัตถุดิบในตู้เย็นจากโทรศัพท์ และทำการทำนายจากโมเดลซึ่งสิ่งที่ได้จากการทำนายคือ รูปภาพที่มีการตีกรอบวัตถุดิบที่สามารถตรวจจับได้ และบอกถึงคลาสนั้นว่าสิ่งที่ตรวจจับได้คืออะไร รวมทั้งบอกค่าความมั่นใจในการทำนายเป็นคลาสนั้น ซึ่งได้ผลลัพธ์ดังรูปที่ 8



รูปที่ 8 ภาพด้านซ้ายเป็นภาพที่ถ่ายวัตถุดิบในตู้เย็นจากกล้องโทรศัพท์  
และภาพด้านขวาคือภาพที่ทำนายการตรวจจับจากโมเดล

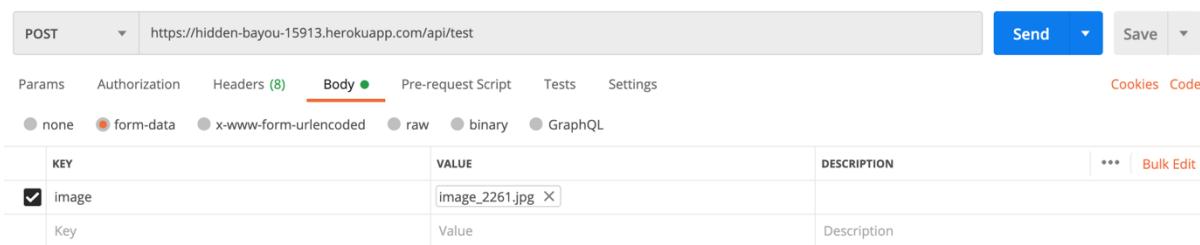
จากรูปที่ 8 ด้านซ้ายเป็นภาพที่ถ่ายวัตถุดิบในตู้เย็นจากกล้องโทรศัพท์ และภาพด้านขวาคือภาพที่ได้จากการทำนายและตรวจจับวัตถุดิบได้ โดยผลลัพธ์ที่ได้คือรูปภาพที่แสดงทางด้านขวาจะปรากฏภาพที่มีการตรวจจับวัตถุดิบและถูกตีกรอบเมื่อตรวจจับได้ พร้อมทั้งบอกว่าสิ่งที่จับได้คือคลาสใด จากรูปที่ 8 ด้านขวาสามารถตรวจจับแอปเปิล, ขมเปง, มะนาว, ดอกกะหลា, แตงกวา, มันฝรั่ง และแก้วมังกร โดยค่าความมั่นใจในการทำนายของแต่ละคลาสจะปรากฏอยู่ในภาพ ซึ่งจะ

เป็นค่าที่ไม่เกิน 1 หากคิดเป็นเปอร์เซ็นต์ค่าความมั่นใจเท่ากับ 1.00 ก็คือมีค่าความมั่นใจ 100 เปอร์เซ็นต์

```
cucumber: 87%
apple: 99%
bread: 90%
apple: 28%
lime: 61%
apple: 100%
apple: 94%
lime: 100%
cucumber: 99%
potato: 100%
apple: 94%
cauliflower: 30%
lime: 100%
dragonfruit: 100%
lime: 100%
```

รูปที่ 9 ภาพแสดงผลลัพธ์ที่ได้จากการทำนายและการตรวจจับของโมเดลของภาพที่ 8

ในการส่งรูปภาพจากแอปพลิเคชันสู่โมเดล โดยผ่าน API ของโมเดลที่ทำการ Deploy ของ Heroku ซึ่งจะทำการส่งรูปภาพที่ถ่ายจากโทรศัพท์มือถือ และสิ่งที่ได้จากการทำนายคือ คลาส ทั้งหมดที่สามารถตรวจจับและทำนายได้ รวมไปถึงค่าความถูกต้องของการทำนายเป็นคลาสนั้น และรูปภาพที่ทำการตีกรอบเหมือนรูปที่ 8 ด้านขวา โดย Response รูปภาพมาในรูปของ base 64 และเมื่อรับค่ามาถึง Convert รูปจาก base 64 เป็น bitmap เพื่อแสดงผลบนแอปพลิเคชัน



รูปที่ 10 ภาพแสดงการส่งผลลัพธ์เข้าไปบนโมเดลผ่าน Api ใน postman

```
1 "class": [
2     "potato: 0.9997",
3     "lime: 0.9993",
4     "dragonfruit: 0.9991",
5     "lime: 0.9990",
6     "lime: 0.9976",
7     "apple: 0.9969",
8     "apple: 0.9907",
9     "cucumber: 0.9846",
10    "apple: 0.9798",
11    "apple: 0.9434",
12    "bread: 0.9120",
13    "cucumber: 0.7924"
14 ],
15 "image": "iVBORw0KGgoAAAANSUhEUgAAABdGAAIAAACKa3BdAAEAELEQVR4n0z93a4t0Y8giJF5rHPyq6+g00MBug7Db//C/cNzZg+Gowhm/6DRoNzJ/
16 L1VXVmnb3CtEXFCn+KVasfU5+Vd1TQy+S1KiqIoiPoCv+v/7f/J51EAADABARUKJEIKLE/BUHIWJRjXB8SX3g/GXBUjcw5GeupJD5WbjspzTf1u292
+BFTrbVJ5BFMuAckm#4IuJY1C1iprFXIYtEmxZ3QGNXR04w0FEHYqcJgp-Wf68psbuaa6adXw1Xr
+WuHRriwalWdk4YBuPLUolHSB3Dkd4JbrBEu6VA1WaIUknrrk31fstZKQj7oVX3DX7iuRH9X6U4bh/wM7Nh8NgV0qyYuzJ0wQv
+zDxjDM4TGoox3MALjqg9PnuAYyBRMkjZM0ajF3Cbz4Ywfv/SIJz2CgM204wAIVieVpZwd5MtS14FTetmoj08+T8vfcmydINERnD8zzP8
-C_077zvz-7Y9lTY9lTY9lTF-04UjDrdiVlaDaCaoyY8Pd-DVUKCaY9g==Dw0G6771>0>G>A9DDB=4;C_D4=ANIMTEQH4;HdDUTz-/
```

รูปที่ 11 ภาพแสดง Response หลังจากการส่งรูปไปทำนายบนโมเดล

ในการวัดประสิทธิภาพของโมเดลจะพิจารณาจากค่าต่าง ๆ โดยเมื่อค่าเหล่านี้มีค่ามากและเข้าใกล้ 1 ก็จะสามารถบ่งบอกได้ว่าโมเดลนั้นมีประสิทธิภาพที่ดี โดยค่าที่จะนำมาพิจารณาได้แก่

1. Precision คือ ค่าความแม่นยำของโมเดล ในการทำนายคลาสนั้น ๆ ที่สนใจ เป็นการพิจารณา จำนวนผลลัพธ์ที่ทำนายถูกจากข้อมูลของผลลัพธ์ที่ทำนายเป็น Class ที่พิจารณา
  2. Recall คือ ค่าที่บ่งบอกว่าโมเดล ได้ตอบคำถามของ Class นั้น ๆ ที่สนใจดีหรือไม่ เป็นการพิจารณา จำนวนผลลัพธ์ที่ทำนายถูกจากจำนวนทั้งหมดใน Class ที่พิจารณา
  3. F1-measure คือ ค่าประสิทธิภาพโดยรวมโดยจะคำนวณด้วยค่า Precision และ Recall

โดยผู้จัดทำได้ทำการนำ Test set เข้าไปทดสอบในโมเดล และผลลัพธ์ที่ได้มีผลดังนี้

```
detections_count = 4845, unique_truth_count = 2722
class_id = 0, name = apple, ap = 93.04%           (TP = 436, FP = 144)
class_id = 1, name = banana, ap = 81.48%          (TP = 52, FP = 0)
class_id = 2, name = butter, ap = 97.69%          (TP = 53, FP = 3)
class_id = 3, name = milk, ap = 97.60%            (TP = 130, FP = 55)
class_id = 4, name = water, ap = 87.63%            (TP = 311, FP = 53)
class_id = 5, name = lime, ap = 90.80%            (TP = 323, FP = 60)
class_id = 6, name = cucumber, ap = 90.32%        (TP = 197, FP = 46)
class_id = 7, name = carrot, ap = 89.97%          (TP = 116, FP = 16)
class_id = 8, name = chicken, ap = 98.73%          (TP = 42, FP = 5)
class_id = 9, name = egg, ap = 97.07%              (TP = 89, FP = 1)
class_id = 10, name = corn, ap = 77.26%            (TP = 42, FP = 33)
class_id = 11, name = grapefruit, ap = 96.97%       (TP = 33, FP = 4)
class_id = 12, name = cauliflower, ap = 89.37%       (TP = 103, FP = 16)
class_id = 13, name = pineapple, ap = 83.02%         (TP = 80, FP = 16)
class_id = 14, name = potato, ap = 95.52%            (TP = 175, FP = 21)
class_id = 15, name = dragonfruit, ap = 78.58%       (TP = 50, FP = 11)
class_id = 16, name = ketchup, ap = 98.36%            (TP = 60, FP = 0)
class_id = 17, name = bread, ap = 94.45%              (TP = 87, FP = 36)
class_id = 18, name = cabbage, ap = 95.17%             (TP = 62, FP = 1)
class_id = 19, name = pumpkin, ap = 90.43%            (TP = 62, FP = 13)
```

รบที่ 12 ภาพแสดงผลลัพธ์ที่ได้จากการทำงานและการตรวจจับของโนมเดล

จากรูปที่ 12 จะแสดงให้เห็นถึงค่าแม่นยำการในการทำนายของคลาสนั้น ๆ ทั้งหมด 20 คลาส และแสดงค่า True positive และ False positive ของแต่ละคลาสนั้น ๆ จะเห็นได้ว่าค่าความแม่นยำในการทำนายของแต่ละคลาส มีค่ามากกว่า 70 เปอร์เซ็นต์

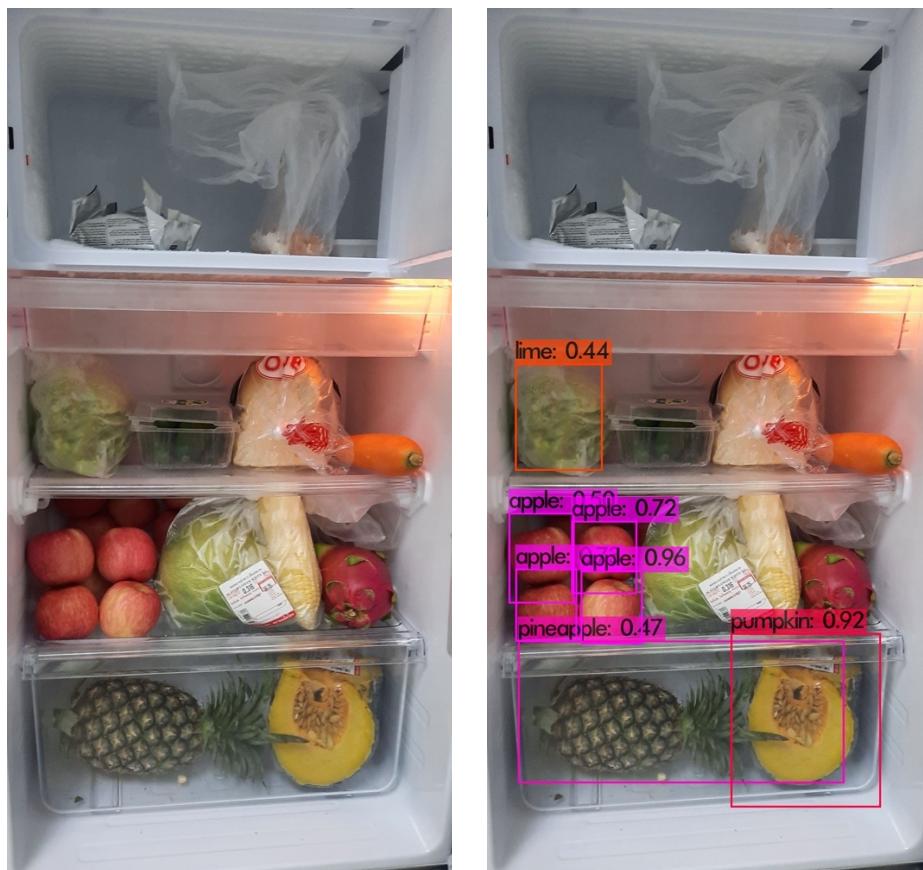
```
for conf_thresh = 0.25, precision = 0.82, recall = 0.92, F1-score = 0.87
for conf_thresh = 0.25, TP = 2503, FP = 534, FN = 219, average IoU = 69.99 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.911732, or 91.17 %
Total Detection Time: 4193 Seconds
```

รูปที่ 13 ภาพแสดงผลลัพธ์ที่ได้จากการทำนายและการตรวจจับของโมเดล

จากรูปที่ 13 แสดงค่าความแม่นยำของโมเดล ซึ่งในการทำนาย Test set ได้ค่าความแม่นยำในการทำนายอยู่ที่ 0.82 หรือ 82 เปอร์เซ็นต์, ค่า Recall อยู่ที่ 0.92 หรือ 92 เปอร์เซ็นต์ และค่า F1-score อยู่ที่ 0.87 หรือ 87 เปอร์เซ็นต์

## Problem



รูปที่ 12 ภาพด้านซ้ายเป็นภาพที่ถ่ายวัตถุในตู้เย็นจากกล้องโทรศัพท์  
และภาพด้านขวาคือภาพที่ทำนายการตรวจจับจากโมเดล

จากรูปที่ 12 ผู้จัดทำได้ทำการทดสอบจากภาพถ่ายอีกภาพ ซึ่งจะเห็นได้ว่าผลลัพธ์ที่ในการตรวจจับได้จะตรวจจับวัตถุได้น้อยกว่ารูปที่ 8 โดยวัตถุที่สามารถจับได้คือ แอปเปิล, สับปะรด และพิกทอง ซึ่งจะเห็นได้ว่ามีการตรวจจับและทำนายผิดจากผักกาด ทำนายเป็นมะนาว

ซึ่งปัญหาที่เจอก็อาจจะเป็นจากการที่ Dataset ที่นำไปเรียนรู้มีจำนวนไม่มากพอ ทำให้มีอัตราผิดพลาดสูง ไม่สามารถจัดแบ่งข้อมูลได้ดี ทำให้โมเดลไม่สามารถตรวจจับและทำนายคลาสให้ผลลัพธ์ที่ถูกต้องได้

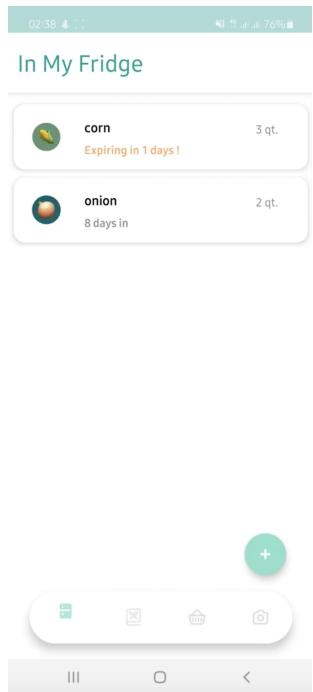
## GUI

ในหน้าแรกของแอปพลิเคชัน Easy Smile Fridge เป็นดังนี้



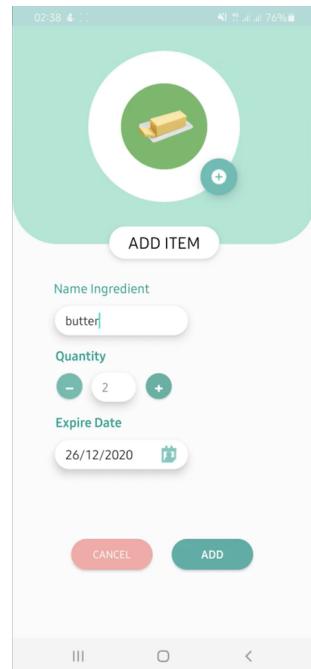
รูปที่ 13 แสดงหน้าแรกของแอปพลิเคชัน Easy Smile Fridge

เมื่อกดที่รูปตู้เย็นจะเข้าสู่หน้าหลักของแอปพลิเคชัน โดยในแอปจะมีให้เลือกอยู่สี่เมนูหลัก คือ หน้าหลักของตู้เย็น, ค้นหาเมนู, แผนที่ และค้นหาเมนูด้วยกล้องถ่ายรูป ซึ่งหน้าแรกของตู้เย็น จะประกอบไปด้วย รายละเอียดของวัตถุติดบนตู้เย็นที่อยู่ภายในตู้เย็น โดยสามารถเพิ่มหรือลดสิ่งที่อยู่ในตู้เย็นได้ดังรูปที่ 13



รูปที่ 14 แสดงหน้าหลักของตู้เย็นของแอปพลิเคชัน

- พัฟ์ชันที่สามารถบอกได้ว่าภายในตู้เย็นนั้นมีวัตถุดิบอะไรบางซึ่งเป็นหน้าแรกเมื่อกดเข้ามา
- พัฟ์ชันการเพิ่มวัตถุดิบภายในตู้เย็น
- พัฟ์ชันที่สามารถค้นหาเมนูอาหารจากวัตถุดิบที่มีภายในตู้เย็น โดยสามารถพิมพ์ค้นหาชื่อเมนูอาหารที่ต้องการ หรือ ให้แอปพลิเคชันทำการสกุลเมนูอาหารให้
- พัฟ์ชันค้นหาร้านขายของชำที่อยู่ใกล้ตัว
- พัฟ์ชันค้นหาเมนูอาหารโดยถ่ายรูปจากวัตถุดิบที่อยู่ภายในตู้เย็น

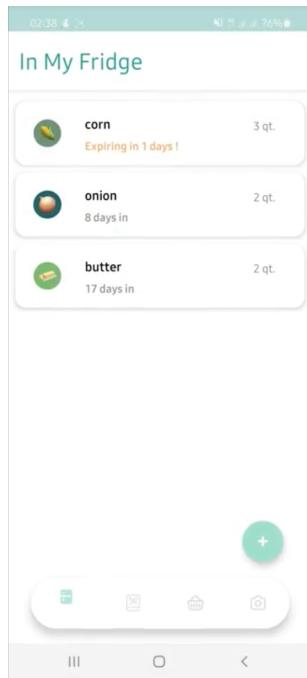


รูปที่ 15 แสดงหน้าเพิ่มวัตถุดิบของแอพลิเคชัน

จากรูปเป็นหน้าที่หลังจากกดปุ่มฟังก์ชันของการเพิ่มวัตถุดิบ จะประกอบไปด้วย

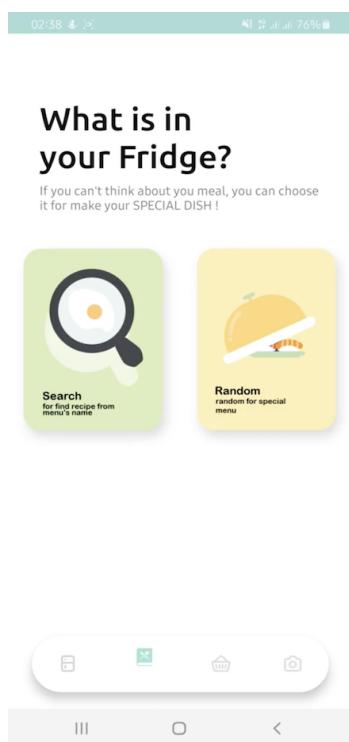
- ปุ่ม icon ของวัตถุดิบเพิ่มเข้ามาในตู้เย็น
- กล่องใส่ชื่อวัตถุดิบ
- กล่องใส่จำนวนวัตถุดิบ
- กล่องใส่วันหมดอายุของวัตถุดิบ

และเมื่อกรอกครบแล้วทำการกดปุ่ม ADD เพื่อเพิ่มวัตถุดิบเข้ารายการของตู้เย็น หากต้องการยกเลิกในการเพิ่มให้กดที่ปุ่ม Cancel เพื่อยกเลิก



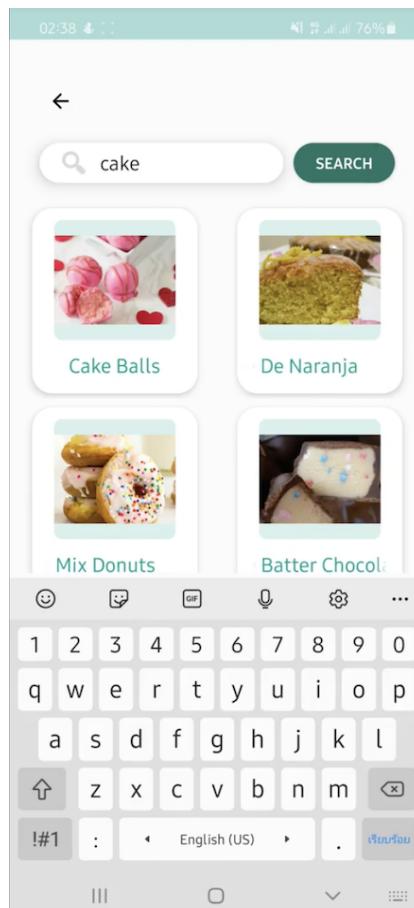
รูปที่ 16 แสดงหน้าหลักของตู้เย็นของแอพพลิเคชัน

จากรูปที่ 16 เป็นหน้าวัตถุดิบทั้งหมดที่มีอยู่ในตู้เย็นหลังจากที่ทำการ เพิ่มวัตถุดิบจะเห็นได้ว่ามีวัตถุดิบเพิ่มขึ้นมาอีกหนึ่งรายการ ซึ่งในหนึ่งรายการจะแสดงชื่อของวัตถุดิบ จำนวนวันที่จะหมดอายุ และปริมาณของวัตถุดิบนั้น ๆ



รูปที่ 17 แสดงหน้าค้นหาเมนูของแอพพลิเคชัน

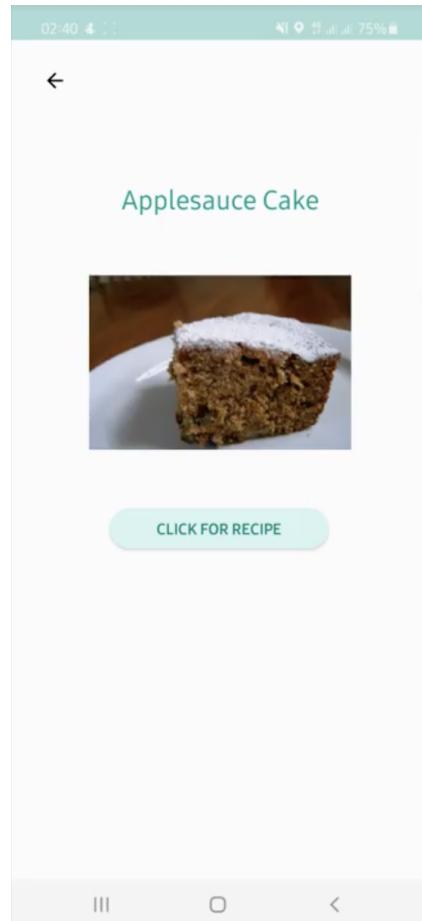
จากรูปที่ 17 เป็นหน้าของฟังก์ชันการค้นหาเมนูอาหารจากวัตถุดิบที่มีอยู่ในตู้เย็น โดยสามารถเลือกได้ว่าต้องการค้นหาซีอเมนูอาหารที่เราต้องการ หรือให้แอปพลิเคชันทำการสุ่มเมนูอาหารให้ โดยเมนูอาหารที่เลือกเอง หรือสุ่มนั้น จะอ้างอิงจากวัตถุดิบที่มีอยู่ในตู้เย็น



รูปที่ 18 แสดงหน้าค้นหาเมนูของแอปพลิเคชัน

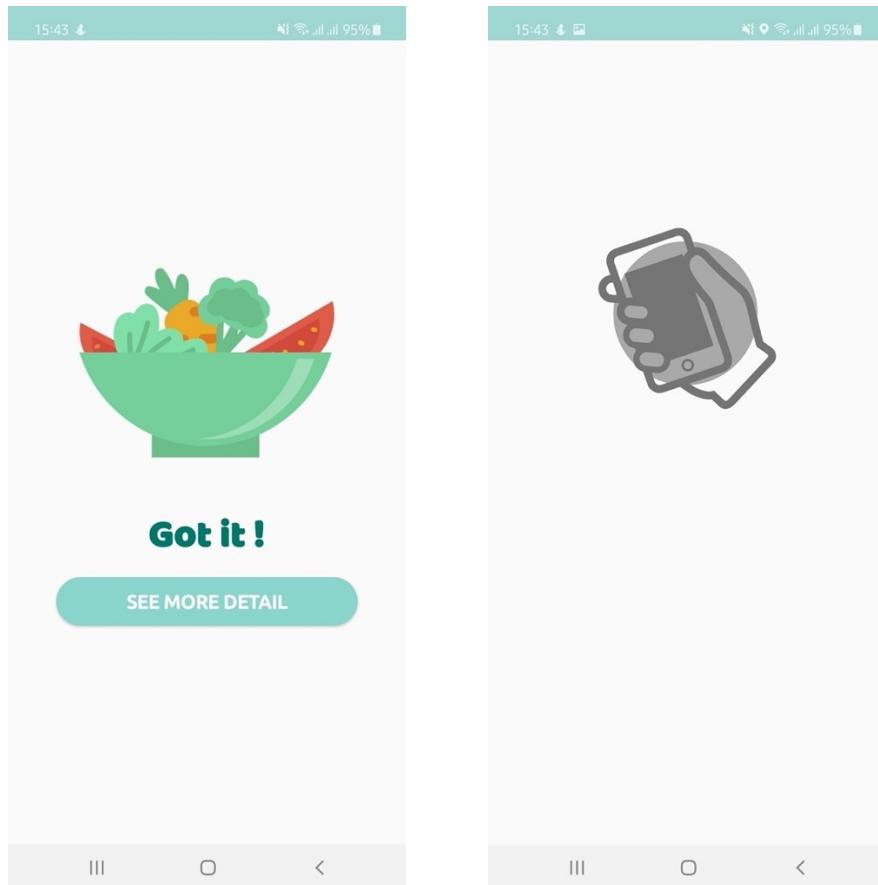
จากรูปที่ 18 เป็นหน้าหลังจากที่ทำการเลือกกดค้นหาเมนูอาหารที่ต้องการ เมื่อเลือกค้นหาเมนูแล้ว สามารถพิมพ์ชื่อเมนูที่ต้องการจะหาในกล่อง search จากนั้นคลิกที่ปุ่ม search เพื่อทำการค้นหา สิ่งที่ปรากฏจะเป็นชื่อของเมนูที่เราต้องการจะหาทั้งหมด เช่นค้นหาคำว่า Cake เมนูที่แสดงจะขึ้นเมนูที่มีคำว่า Cake ทั้งหมดมา โดยเมนูที่ได้นี้ได้จากการยิง API ของ spoonacular

และเมื่อเลือกเมนูใดเมนูหนึ่งจากการค้นหาแล้ว เมื่อคลิกเข้าไปที่เมนูนั้นจะเข้าสู่หน้าของรายละเอียดของเมนูนั้น ๆ โดยรายละเอียดที่จะแสดงมีชื่อของเมนู ภาพของเมนูนั้น และปุ่มเพื่อดูรายละเอียดวิธีการทำของเมนูนั้นดังรูปที่ 19



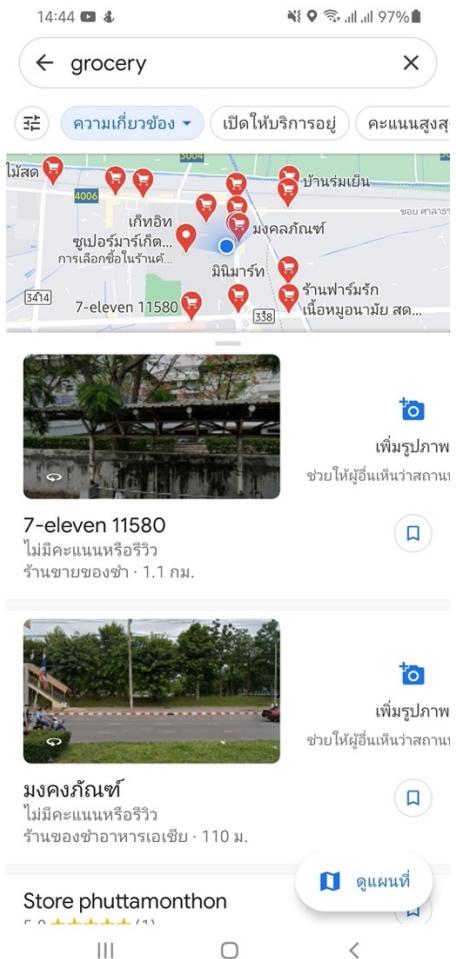
รูปที่ 19 แสดงหน้าแสดงรายละเอียดของเมนูอาหารที่เลือก

อีกฟังก์ชันในการค้นหาเมนูอาหารคือฟังก์ชันการสุ่มหรือ Random โดยเมื่อเลือกฟังก์ชันนี้แล้วจะทำการสุ่มเมนูอาหารที่มีวัตถุดิบจากของในตู้เย็นมา 1 รายการและเมื่อต้องการที่จะเปลี่ยนเมนูอาหารให้เบyx่าหน้าจอของโทรศัพท์เพื่อทำการเปลี่ยนเมนูอาหาร หากต้องการที่จะดูรายละเอียดของเมนูที่ได้รับให้คลิกที่ปุ่ม see more detail เพื่อดูรายละเอียดของเมนูที่ได้จากการสุ่มครั้งนั้น



รูปที่ 20 แสดงส่วนเมนูการจากวัตถุดิบในรายการของตู้เย็น

พังก์ชันต่อไปเป็นพังก์ชันการค้นหาร้านขายของชำหรือตลาดที่อยู่ใกล้เคียงมากที่สุด โดย เมื่อเลือกพังก์ชันนี้แล้วจะย้ายไปที่ Google Map ของโทรศัพท์ซึ่งจะปรากฏร้านขายของชำหรือ ตลาดที่ใกล้เคียงมากที่สุด และสามารถค้นหาร้านค้าได้



รูปที่ 21 แสดงหน้าแสดงค้นหาเว็บไซต์ค้าปลีกเชียงจากกุเกลเมป

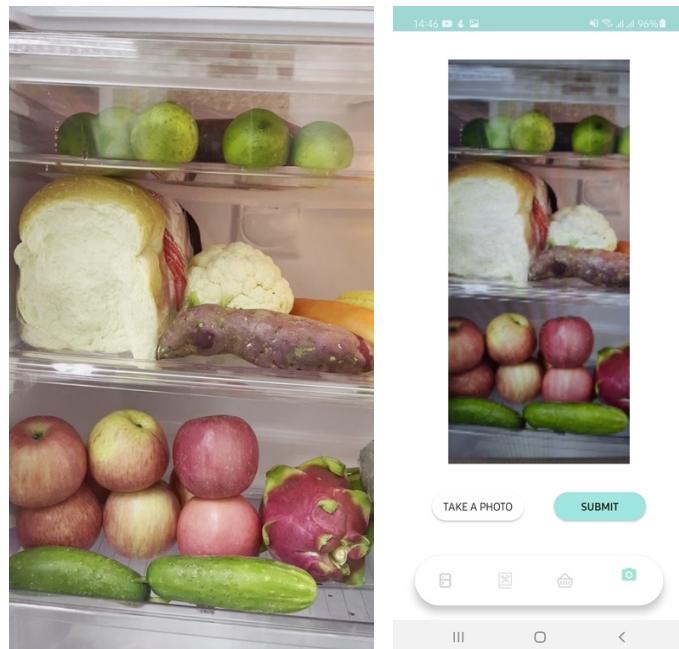
พังก์ชันสุดท้ายคือพังก์ชันการค้นหาเมนูอาหารจากการถ่ายรูปวัตถุดิบที่มีจากตู้เย็น โดยในพังก์ชันนี้จะให้ผู้ใช้ถ่ายภาพสิ่งที่มีอยู่ในตู้เย็น โดยเมื่อถ่ายแล้วสามารถเลือกที่จะถ่ายใหม่ได้อีกรอบ หากเลือกรูปภาพที่จะต้องการแล้วให้กด Submit ใน การยืนยันรูปภาพ



รูปที่ 22 แสดงฟังก์ชันค้นหาเมนูอาหารโดยการถ่ายรูปวัตถุดิบ

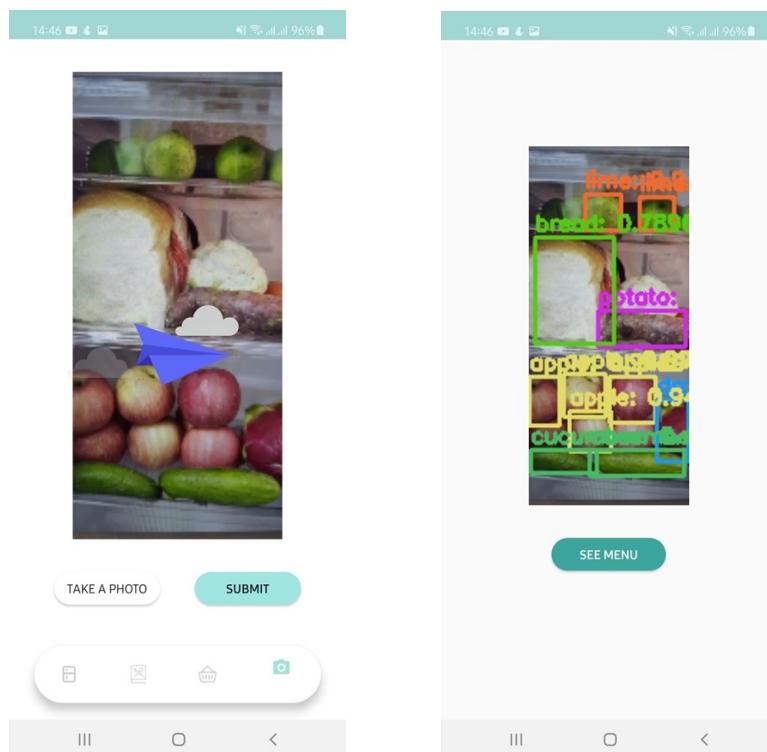
หลังจากนั้นระบบจะทำการยิง API ไปที่ Model เพื่อทำการคำนวณและตรวจจับสิ่งที่มีอยู่ในรูปภาพอุปกรณ์ และส่งค่ากลับมาเป็นรูปภาพที่มีการตีกรอบว่าในรูปภาพนั้นมีวัตถุดิบใดบ้าง และเมื่อกด see menu จะเป็นการเลือกดูเมนูอาหารที่ได้ทำการคิดค้นจากสิ่งที่สามารถตรวจจับได้ ซึ่งขึ้นตอนต่าง ๆ แสดงดังภาพข้างล่างดังนี้

- ขั้นแรกกดปุ่ม Take a photo และถ่ายรูปภาพวัตถุดิบที่มีอยู่ในตู้เย็นจากโทรศัพท์



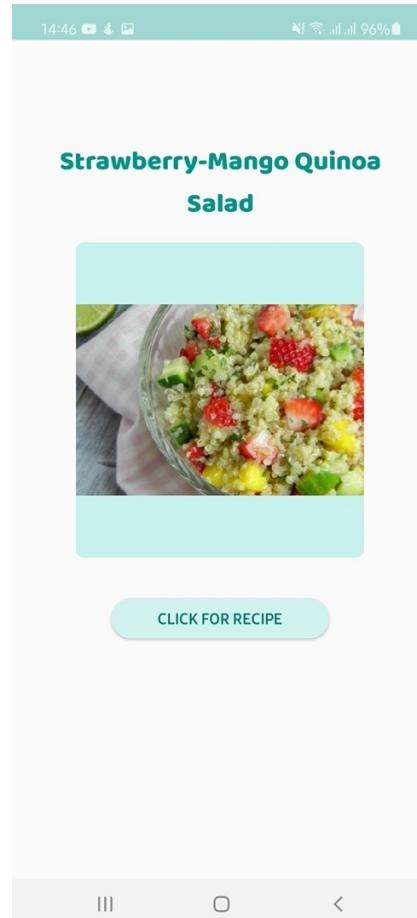
รูปที่ 23 แสดงภาพขั้นแรกของฟังก์ชันค้นหาราเมนูอาหารโดยการถ่ายรูปวัตถุดิบ

- หลังจากนั้นทำการกด submit เพื่อเลือกรูปภาพที่ต้องการ ซึ่งรูปที่จะเป็นแปลงเป็น Bitmap ก่อนที่จะส่งรูปไปยัง API ของ Model ที่ทำการ Deploy และจะได้รูป Response ที่ได้ทำการตรวจจับวัตถุกลับมาดังรูป



รูปที่ 24 แสดงภาพขั้นตอนการส่องรูปภาพครัวรับภาพที่ทำการตรวจจับแล้ว

- หลังจากนั้นทำการกด see menu เพื่อดูว่าเมนูอาหารที่สามารถทำได้คืออะไร และเมื่อเข้าหน้ารายละเอียดของเมนูนั้น จะแสดงชื่อของเมนูอาหาร ภาพของเมนูอาหาร และปุ่มคลิกเพื่อดูสูตรอาหารของเมนูนั้นดังรูป



รูปที่ 25 แสดงภาพขั้นเมนูอาหารที่ได้จากการตรวจจับวัตถุในตู้เย็น

## Apendix

Code ของ Mobile application : <https://bitbucket.org/N1flheimre/wtf>