

แอปพลิเคชันรวมสินค้าไลฟ์สดด้วยการประมวลผลทางภาพ  
application live shop with image processing

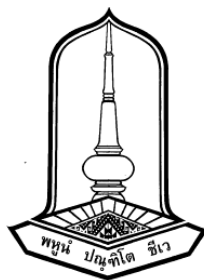
โครงการปริญญาานิพนธ์  
ของ  
นายกฤตเมธ บัวสิงห์  
นายกิตติศักดิ์ มนพรหมมา

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
ปีการศึกษา 2565  
ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม  
คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

แอปพลิเคชันรวมสินค้าไลฟ์สดด้วยการประมวลผลทางภาพ  
application live shop with image processing

โครงการปริญญานิพนธ์  
ของ  
นายกฤตเมธ บัวสิงห์  
นายกิตติศักดิ์ มนพรหมมา

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร  
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
ปีการศึกษา 2565  
ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม  
คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม



คณะกรรมการสอบโครงการปริญญานิพนธ์ ได้พิจารณาปริญญานิพนธ์ของ นายกฤตเมธ บัวสิงห์ และนายกิตติศักดิ์ มั่นพรมมา แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบโครงการปริญญานิพนธ์

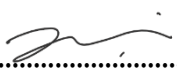
ประธานสอบ

.....  
(.....)

กรรมการ

.....  
(.....)

ที่ปรึกษาโครงการปริญญานิพนธ์หลัก

  
.....  
(ดร.พัฒนพงษ์ ชมพูวิเศษ)

หลักสูตรวิทยาการคอมพิวเตอร์อนุมัติให้รับโครงการปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

.....  
(อาจารย์พชร พฤกษ์ศรี)

.....  
(ผู้ช่วยศาสตราจารย์พิมลรัตน์ อ้วนศรีเมือง)

อาจารย์ผู้ประสานงานวิชาโครงการปริญญานิพนธ์

วันที่ เดือน พ.ศ.

## สารบัญ

### หน้า

สารบัญ.....	ก
สารบัญตาราง .....	1
สารบัญรูปภาพ .....	2
บทที่ 1 บทนำ.....	4
1.1 หลักการและเหตุผล.....	4
1.2 วัตถุประสงค์ของโครงการ.....	4
1.3 ขอบเขตของโครงการ .....	4
1.3.1 ผู้ใช้งาน (ใช้งานผ่าน Mobile Application).....	4
1.3.2 ผู้ดูแลระบบ (ใช้งานผ่าน Desktop Application ที่พัฒนาด้วย Python).....	6
1.3.3 ระบบ (ระบบพัฒนาเป็น Module ด้วย Python).....	6
1.4 ภาพรวมของระบบ.....	8
จากภาพรวมระบบจะการทำงานดังนี้ .....	8
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	9
1.5.1 ช่วยให้ผู้ใช้งานที่เป็นผู้ค้าขายมีช่องทางการขายสินค้ามากขึ้น.....	9
1.5.2 ช่วยให้ผู้ใช้งานที่เป็นลูกค้ามีความสะดวกมากขึ้น.....	9
1.6 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน .....	9
1.6.1 ฮาร์ดแวร์.....	9
1.6.2 ซอฟต์แวร์.....	9
1.7 แผนการดำเนินงาน.....	10
1.8 ตัวอย่างโปรแกรม.....	11
1.8.1 แอปพลิเคชันบนอุปกรณ์ไร้สายแบบเคลื่อนที่.....	11
1.8.2 แอปพลิเคชันบนคอมพิวเตอร์สำหรับผู้ดูแลระบบ.....	16
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	19
2.1 ทฤษฎีที่เกี่ยวข้อง .....	19
2.1.1 Flutter .....	19
2.1.2 Firebase .....	20
2.1.3 การประมวลผลภาพ (Image Processing) .....	21
2.1.4 OCR หรือ Optical Character Recognition .....	23
2.1.1 Object Detection.....	24
2.1.1 YOLO (You Only Look Once).....	24
2.2 ระบบงานที่เกี่ยวข้อง.....	25

## สารบัญ(ต่อ)

	หน้า
2.2.1 Safety Helmet Detection Based on YOLOv5.....	25
บทที่ 3.....	28
3.1 เก็บรวบรวมข้อมูล.....	29
3.2 การสร้างโมเดล.....	30
3.2.1 การเตรียมข้อมูลในการ training .....	30
3.2.2 การวาดภาพผลเฉลย.....	30
3.2.3 Data Augmentation .....	32
3.2.4 การสร้างโมเดล.....	33
3.3 ในส่วนของ desktop application.....	36
3.4 การตัดบรรทัดด้วยเทคนิค projection profile.....	42
3.5 อ่านข้อความ OCR.....	42
3.6 Mobile application.....	44
3.7 การจัดเก็บข้อมูล.....	44
3.8 วัดประสิทธิภาพ.....	46
อ้างอิง.....	48

## สารบัญตาราง

	หน้า
ตารางที่ 1 แผนการดำเนินงาน .....	10
ตารางที่ 2 ตารางจำนวนและเวลาเฉลี่ยของวิดีโอในแต่ละร้านค้า.....	29
ตารางที่ 3 จำนวนของข้อมูลในแต่ละส่วน .....	31
ตารางที่ 4 จำนวนของข้อมูลก่อนและหลังทำการ Generate .....	32
ตารางที่ 5 clone folder ของ yoloV5 และติดตั้งแพ็คเกจที่จำเป็น.....	33
ตารางที่ 6 การต่อ google colab เข้ากับ google drive .....	33
ตารางที่ 7 การกำหนด path ของไฟล์ข้อมูลที่ได้เตรียมมาและทำการประกาศชื่อของ class.....	34
ตารางที่ 8 การกำหนด จำนวนของคลาส.....	34
ตารางที่ 9 การเทรนข้อมูลและพารามิเตอร์ที่ใช้ในการเทรน .....	34
ตารางที่ 10 ผลลัพธ์ในการเทรนโมเดล .....	34
ตารางที่ 11 การ detect ทดสอบโมเดล.....	35
ตารางที่ 12 การ import ทรัพยากรที่จำเป็น .....	36
ตารางที่ 13 constructor.....	36
ตารางที่ 14 Function load_model.....	37
ตารางที่ 15 Function score_frame .....	37
ตารางที่ 16 Function plot_box.....	37
ตารางที่ 17 function __call__ (1).....	38
ตารางที่ 18 function __call__ (2).....	38
ตารางที่ 19 function __call__ (3).....	39
ตารางที่ 20 การเชื่อมต่อ firebase กับ python.....	41
ตารางที่ 22 การแปลงรูปเป็นตัวอักษรด้วย tesserract.....	43

## สารบัญรูปภาพ

	หน้า
ภาพประกอบที่ 1.1 ตัวอย่างข้อมูลที่นำมาแยก.....	7
ภาพประกอบที่ 1.2 ภาพรวมของระบบ.....	8
ภาพประกอบที่ 1.3 หน้าแรกของโปรแกรม .....	11
ภาพประกอบที่ 1.4 หน้าล็อกอินเข้าสู่ระบบ .....	12
ภาพประกอบที่ 1.5 หน้าหลักของแอปพลิเคชัน .....	13
ภาพประกอบที่ 1.6 หน้าค้นหาสินค้า.....	14
ภาพประกอบที่ 1.7 หน้ารายละเอียดสินค้า .....	15
ภาพประกอบที่ 1.8 หน้าแรกโปรแกรม .....	16
ภาพประกอบที่ 1.9 หน้าเลือกพื้นที่ภาพ.....	17
ภาพประกอบที่ 1.10 หน้าผลลัพธ์ในการถ่ายภาพหน้าจอ.....	17
ภาพประกอบที่ 1.11 หน้ารายละเอียดข้อมูล.....	18
ภาพประกอบที่ 2.1 ตัวอย่างโครงสร้าง Flutter.....	19
ภาพประกอบที่ 2.2 การตรวจจับและประมวลผลใบหน้าจากภาพ [4] .....	22
ภาพประกอบที่ 2.3 ขบวนการการทำงานของ Optical Character Recognition.....	24
ภาพประกอบที่ 2.4 ภาพการตรวจจับหวนกนิรภัยของคนงาน [8] .....	25
ภาพประกอบที่ 2.5 การแบ่งช่องของรูปภาพออกเป็น grid [9].....	26
ภาพประกอบที่ 2.6 ภาพของเทคนิค Anchor Box [9].....	26
ภาพประกอบที่ 2.7 การทำงานของขบวนการ IOU [9].....	27
ภาพประกอบที่ 3.1 ขั้นตอนการดำเนินงานของระบบ .....	28
ภาพประกอบที่ 3.2 Streamlabs Desktop.....	29
ภาพประกอบที่ 3.3 ตัวอย่างรูปภาพที่ทำการเก็บ.....	30
ภาพประกอบที่ 3.4 การวาดภาพผลเฉลยรูปภาพใน Roboflow .....	30
ภาพประกอบที่ 3.5 การวาดภาพผลเฉลยรูปภาพใน Roboflow (2).....	31
ภาพประกอบที่ 3.6 ภาพผลเฉลยที่ถูกตัดจากทั้ง 3 ร้านค้า .....	31
ภาพประกอบที่ 3.7 การเพิ่มจำนวนรูปภาพใน Traing Set .....	32
ภาพประกอบที่ 3.8 ตัวอย่างรูปภาพที่ Generate ออกมา.....	32
ภาพประกอบที่ 3.9 ผลลัพธ์ในการ detect ของรูปร้านค้าแต่ละร้าน .....	35
ภาพประกอบที่ 3.10 การทำงานในส่วนของการเลือกส่วนที่ต้องการ.....	40
ภาพประกอบที่ 3.11 ผลลัพธ์ในการ detect และการเลือกพื้นที่ที่ต้องการ.....	40
ภาพประกอบที่ 3.12 ผลลัพธ์ในการ detect ใน desktop appliction.....	41
ภาพประกอบที่ 3.13 เทคนิคการทำ projection profile .....	42
ภาพประกอบที่ 3.14 ผลรวมตรวจสอบของเมทริกซ์.....	42
ภาพประกอบที่ 3.15 การแปลงรูปเป็นตัวอักษร.....	43

## สารบัญรูปภาพ(ต่อ)

	หน้า
ภาพประกอบที่ 3.16 ตัวอย่างหน้าตา UI mobile application .....	44
ภาพประกอบที่ 3.17 การสร้างโปรเจ็ค firebase.....	45
ภาพประกอบที่ 3.18 การสร้าง Document ใน firebase.....	45
ภาพประกอบที่ 3.19 ตัวอย่างข้อมูลและประเภทของข้อมูล .....	46
ภาพประกอบที่ 3.20 กราฟการเปลี่ยนแปลงค่า mAP ของ model.....	46
ภาพประกอบที่ 3.21 สมการ CER .....	47
ภาพประกอบที่ 3.22 ตัวอย่างตัวแปรในสมการ CER.....	47
ภาพประกอบที่ 3.23 ตัวอย่างการหาค่า CER.....	47



## บทที่ 1

### บทนำ

#### 1.1 หลักการและเหตุผล

ปัจจุบันเทคโนโลยีอิเล็กทรอนิกส์ มีความสำคัญในการดำรงชีวิตของผู้คนในปัจจุบันมากขึ้น ผู้คนมากมายได้ทำการใช้เทคโนโลยีในการค้าขายของออนไลน์ผ่านการไลฟ์และได้เกิดการพัฒนาของปัญญาประดิษฐ์ (AI : Artificial Intelligence) คือเครื่องจักรหรือเทคโนโลยีที่มีฟังก์ชันที่มีความสามารถในการทำความเข้าใจ เรียนรู้องค์ความรู้ต่างๆ อาทิเช่น การรับรู้ การเรียนรู้ การให้เหตุผล และการแก้ปัญหาต่างๆ เครื่องมือที่มีความสามารถเหล่านี้ก็ถือว่าเป็น ปัญญาประดิษฐ์ เพราะฉะนั้นจึงสามารถกล่าวได้ว่า AI ถือกำเนิดขึ้นเมื่อเครื่องจักรมีความสามารถที่จะเรียนรู้นั่นเอง

ปัจจุบันการที่จะขายพัสดุนั้นในการไลฟ์สดนั้น ใช้เวลาที่ค่อนข้างมากที่จะต้องให้ลูกค้าเข้ามาเข้าชมเพื่อเลือกสินค้าที่ละชิ้นและทำให้ลูกค้าบางรายที่ไม่สะดวกมาเข้าชมไม่ได้เลือกซื้อสินค้าที่ต้องการจึงทำให้พวกเราได้ทำการพัฒนาแอปพลิเคชันรวมสินค้าไลฟ์สดขึ้นมาเพื่อแก้ไขปัญหาดังกล่าว

ดังนั้นเพื่อให้เกิดความสะดวกสบายพวกเราจึงได้พัฒนาแอปพลิเคชันเพื่อตอบโจทย์การใช้งานของแม่ค้าและลูกค้า ก็คือแอปพลิเคชันรวมสินค้าไลฟ์สดซึ่งจะทำการรวมเอาสินค้าต่างๆในไลฟ์สดของแม่ค้าแต่ละคนมารวมไว้ในแอปพลิเคชันเดียว โดยการใช้ image processing ในการตรวจจับข้อมูลสินค้าในไลฟ์สด และนำข้อมูลมาเก็บในฐานข้อมูลและนำไปแสดงผลในแอปพลิเคชัน และเมื่อเลือกสินค้าที่ต้องการจะสามารถดูข้อมูลของสินค้าและไปสู่นำไลฟ์สดของแม่ค้าได้ เพื่อให้ลูกค้าได้เลือกสินค้ากันอย่างสะดวกและแม่ค้าก็ไม่เสียลูกค้าที่ไม่สะดวกเข้าชม

#### 1.2 วัตถุประสงค์ของโครงการ

สร้างแอปพลิเคชันที่สามารถใช้ AI ในการตรวจสอบและดึงข้อมูลของสินค้าในไลฟ์สดขายของเข้ามาในแอปพลิเคชันได้

#### 1.3 ขอบเขตของโครงการ

โครงการแอปพลิเคชันรวมสินค้าไลฟ์สดทำงานบนระบบปฏิบัติการ android แบ่งการทำงานของระบบออกเป็น 3 ส่วน ประกอบไปด้วย

##### 1.3.1 ผู้ใช้งาน (ใช้งานผ่าน Mobile Application)

1. ผู้ใช้งานสามารถเข้าสู่ระบบผ่าน Facebook ได้
2. ผู้ใช้งานสามารถดูสถานะการไลฟ์สดของร้านแต่ละร้านค้าได้ ประกอบด้วย
  - สถานะ online / offline
  - รูปภาพในการไลฟ์สด ถ้าสถานะ online รูปจะเป็นสีปกติ และถ้าเป็นสถานะ offline รูปจะเป็นรูปภาพขาวดำ

3. ผู้ใช้งานสามารถดูรายการของสินค้าที่แสดงสินค้าของทุกร้านที่มีสถานะ online ทุกร้าน ใน Mobile Application ได้ โดยรายการที่ปรากฏในไลฟ์สด ประกอบไปด้วย

- ชื่อร้านค้า
- รหัสสินค้า
- ชื่อสินค้า
- ราคาสินค้า
- วันเวลาที่ capture รูปภาพ

ตัวอย่าง เช่น ปรีนย่า บิวตี้ ช็อป คือชื่อร้าน ,น39 คือรหัสสินค้า, ดีท็อกญี่ปุ่น คือชื่อสินค้า ,39 คือราคาสินค้า,16/07/2022 16:30 คือ วันเวลาที่ capture รูปภาพ

4. ผู้ใช้งานสามารถเลือกที่สินค้าเพื่อดูรายละเอียดเพิ่มเติมได้โดยรายการที่ปรากฏในไลฟ์สดประกอบด้วย

- ชื่อร้านค้า
- รหัสสินค้า
- ชื่อสินค้า
- ราคาสินค้า
- รูปภาพในการไลฟ์สด
- วันเวลาที่ capture รูปภาพ

ตัวอย่าง เช่น FIRST SHOP V2คือชื่อร้าน ,๐7 คือรหัสสินค้า, สครับ ขมิ้น คือชื่อสินค้า ,59 คือราคาสินค้า,14/06/2022 12:30 คือ วันเวลาที่ capture รูปภาพ

5. เมื่อผู้ใช้เลือกสินค้าผู้ใช้งานสามารถกดปุ่ม “chat”ในหน้ารายละเอียดสินค้าเพื่อไปยังแชท ของร้านค้าได้

6. เมื่อผู้ใช้เลือกสินค้าผู้ใช้งานสามารถกดปุ่ม “buy”ในหน้ารายละเอียดสินค้าเพื่อไปยังแชท ของร้านค้าและจะมีการนำรหัสสินค้า ชื่อสินค้า ราคาสินค้า ส่งไปให้ร้านค้าอัตโนมัติ

7. ผู้ใช้งานสามารถค้นหาสินค้าจากชื่อสินค้าหรือชื่อร้านค้าได้ ในช่วงที่เวลาร้านค้าไลฟ์สดตัวอย่างเช่น

- ชื่อสินค้า
- ชื่อร้านค้า

ตัวอย่าง เช่น ลิป KO(เบอร์ 7) คือชื่อสินค้า, FIRST SHOP V2 คือชื่อร้าน

8. เมื่อร้านค้ามีสถานะ offline ผู้ใช้จะไม่สามารถมองเห็นสินค้าของร้านค้าที่ offline ได้

### 1.3.2 ผู้ดูแลระบบ (ใช้งานผ่าน Desktop Application ที่พัฒนาด้วย Python)

1. ผู้ดูแลระบบสามารถเลือกร้านค้าที่ไลฟ์สดได้ 3 ร้านค้ามีร้านค้า
  - FIRST SHOP V2
  - KANYA SHOP ขายถูกทุกอย่าง
  - มหัศจรรย์"วันของAuuM
2. ผู้ดูแลระบบสามารถหยุดร้านค้าที่กำลังประมวลผลในระบบและแจ้งเตือนไปยัง Application ของผู้ใช้งานได้
3. ผู้ดูแลระบบสามารถดูข้อมูลรายการสินค้า ของแต่ละร้านค้าได้ ประกอบด้วย
  - รหัสสินค้า
  - ชื่อสินค้า
  - ราคาสินค้า

ตัวอย่าง เช่น ๐61 คือรหัสสินค้า, ลิป KO(เบอร์ 8) คือชื่อสินค้า, 59 คือราคาสินค้า

### 1.3.3 ระบบ (ระบบพัฒนาเป็น Module ด้วย Python)

1. ระบบสามารถจับภาพหน้าจอไลฟ์สดได้ 3 ร้านค้ามีร้านค้า
  - FIRST SHOP V2
  - KANYA SHOP ขายถูกทุกอย่าง
  - มหัศจรรย์"วันของAuuM
2. ระบบสามารถส่งสตรีมมิงภาพหน้าจอไลฟ์สดเพื่อนำไปประมวลผลและนำไปแสดงในแต่พื้นที่แสดงผลแต่ละพื้นที่ใน Desktop Application ของผู้ดูแลระบบ ได้ 3 ร้านค้ามีร้านค้า
  - FIRST SHOP V2
  - KANYA SHOP ขายถูกทุกอย่าง
  - มหัศจรรย์"วันของAuuM
3. ระบบสามารถตรวจจับกรอบของตัวอักษร รหัสสินค้า ชื่อสินค้า ราคาสินค้า ที่เป็นรายการรวมสินค้าจากไลฟ์สดได้โดยที่ขนาดไม่เล็กเกินไป
4. ระบบสามารถตัดบรรทัดของข้อมูลในกรอบที่ตรวจจับมาได้
5. ระบบลบข้อมูลสินค้าของร้านค้าที่มีสถานะ offline ได้
6. ระบบสามารถแยกข้อมูลที่แปลงจากภาพเป็นตัวอักษรได้โดยรายการที่ปรากฏในไลฟ์สด ประกอบด้วย
  - รหัสสินค้า
  - ชื่อสินค้า
  - ราคาสินค้า

ตัวอย่าง เช่น ป29 คือรหัสสินค้า, ดีที่อกญี่ปุ่นคือชื่อสินค้า, 39 คือราคาสินค้า

ป29  
 รหัสสินค้า
 

            
 ด็ก๊อญ่ปุ่  
 ชื่อสินค้า
 

            
 39  
 ราคา

**ภาพประกอบที่ 1.1 ตัวอย่างข้อมูลที่นำมาแยก**

7. ระบบสามารถนำข้อมูลไปจัดเก็บในฐานข้อมูลได้โดยรายการที่ปรากฏในไลฟ์สดประกอบด้วย

- ชื่อร้านค้า
- รหัสสินค้า
- ชื่อสินค้า
- ราคาสินค้า
- รูปภาพในการไลฟ์สด
- วันเวลาที่ capture รูปภาพ

ตัวอย่าง เช่น ของฝากใต้ขายส่งราคาถูก คือชื่อร้าน,มม3 คือรหัสสินค้า,เม็ดมะม่วงหัก คือชื่อสินค้า,89บ คือราคาสินค้า,14/06/2022 14:30 คือ วันเวลาที่ capture รูปภาพ

8. ระบบจะทำงานแบบ real-time เป็น Database ของ Firebase

9. การวัดประสิทธิภาพประเมินจากความถูกต้องของการตรวจจับกรอบของข้อความและความถูกต้องของการทำ OCR(Optical Character Recognition) โดยวัดด้วย CER (character error rate)

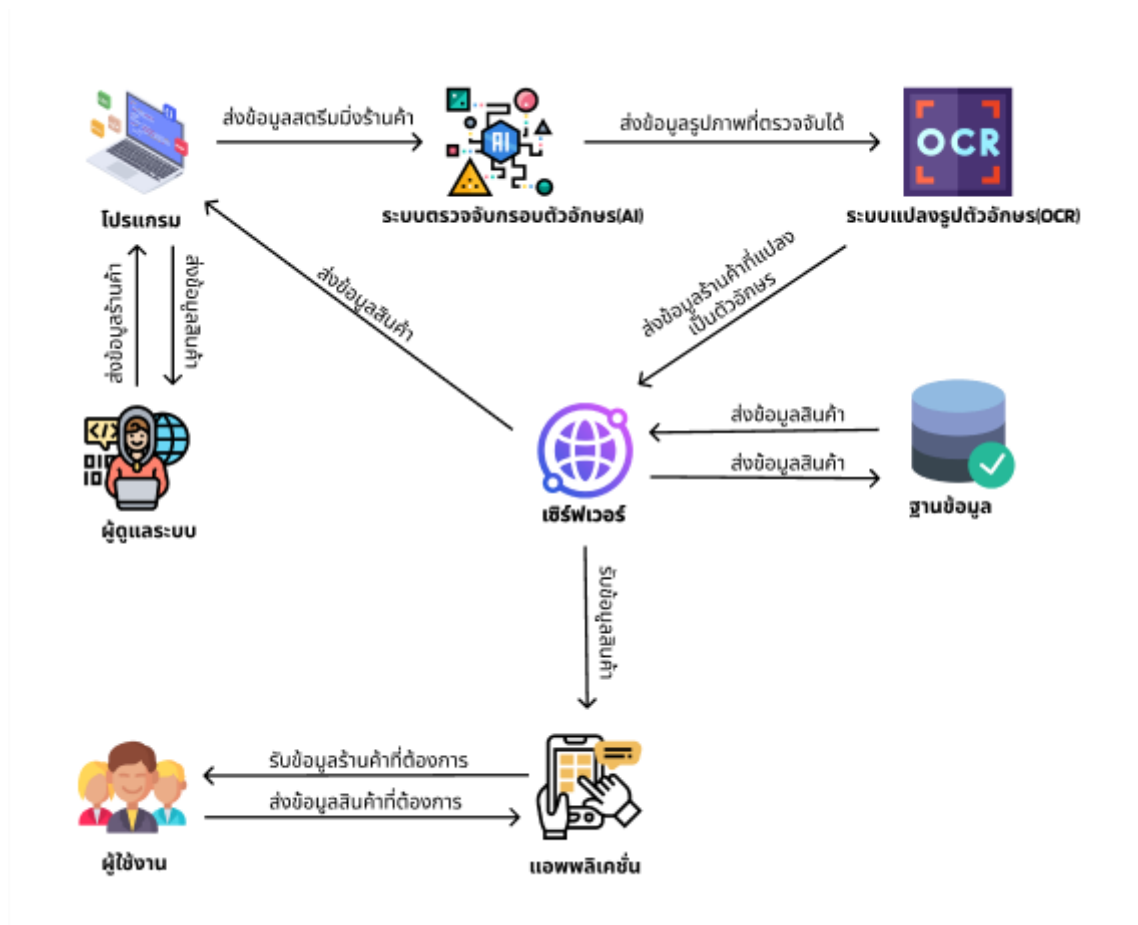
10. ระบบสามารถแจ้งเตือนผู้ใช้งานได้ เมื่อมีร้านค้าที่มีสถานะ online

11. ระบบสามารถตรวจจับข้อความของไลฟ์สดที่จบแล้ว และหยุดการประมวลผลของร้านค้าที่จบไลฟ์และแจ้งเตือนไปยัง Mobile Application ผู้ใช้งานได้

12. ระบบไม่สามารถตรวจจับไลฟ์สดที่ไม่มีกรอบข้อความได้

13. ระบบไม่สามารถตรวจจับกรอบตัวอักษรและตัวอักษรที่มีสีเหมือนกันได้

#### 1.4 ภาพรวมของระบบ



ภาพประกอบที่ 1.2 ภาพรวมของระบบ

จากภาพรวมระบบจะการทำงานดังนี้

1. ผู้ดูแลระบบต้องทำการเลือกร้านค้าที่ต้องการจำนวน 3 ร้านค้า ในโปรแกรมที่พัฒนาด้วย python ที่ปุ่ม “capture” ในตัวโปรแกรมเพื่อทำการครอบหน้าจอในส่วนที่จะนำไปสตรีม และสามารถกดปุ่ม “clear” เพื่อลบร้านค้าที่ต้องการได้
2. หลังจากได้หน้าจอสตรีมที่ต้องการระบบจะทำการตรวจจับกรอบตัวอักษรในสตรีมโดยใช้อัลกอริทึม YOLOv5 และนำข้อมูลที่เป็นรูปภาพกรอบตัวอักษรมาตัดเป็นแถวๆโดยจะใช้เทคนิคในการตรวจ pixel ในหน้าจอทีละแถว เมื่อตรวจเจอช่องว่างระหว่างบรรทัด จะทำการตัดรูปเพื่อแยกข้อมูลออกจากกันและนำชุดข้อมูลสินค้าที่เป็นรูปภาพที่ได้ไปส่งต่อไปยังขบวนการ แปลงภาพเป็นตัวอักษร
3. ขบวนการแปลงภาพเป็นตัวอักษรจะทำการนำชุดข้อมูลที่เป็นภาพมาแปลง เป็นชุดข้อมูลที่เป็นตัวอักษร โดยใช้ Tesseract ในการแปลง
4. หลังจากได้ชุดข้อมูลที่เป็นตัวอักษร จะนำข้อมูลไปเก็บไว้ใน Database ของ firebase ที่มีลักษณะการทำงานแบบ real-time ผ่านตัว server โดยข้อมูลที่ส่งไปจะมี ชื่อร้านค้า รหัสสินค้า ชื่อ

สินค้า ราคาสินค้า รูปภาพของไลฟ์สดที่ทำการบันทึกมา ตัวอย่างของข้อมูลชื่อร้านค้า คือ ปิรันย่า บิวตี้ ซุป รหัสสินค้าคือ ก22, ชื่อสินค้าคือ ดีท็อกซ์ญี่ปุ่น, ราคาสินค้าคือ 39 บาท เป็นต้น

5. server จะทำการส่งข้อมูลสินค้าประกอบไปด้วย ชื่อร้านค้า รหัสสินค้า ชื่อสินค้า ราคาสินค้า และรูปภาพของไลฟ์สด ใน Database ไปแสดงผลใน Desktop Application ของผู้ดูแลระบบ

6. server จะทำการส่งข้อมูลสินค้าประกอบไปด้วย ชื่อร้านค้า รหัสสินค้า ชื่อสินค้า ราคาสินค้า และรูปภาพของไลฟ์สด ใน Database ไปแสดงผลใน mobile application ของผู้ใช้งาน

7. ผู้ใช้ต้องเข้าสู่ระบบผ่านเฟซบุ๊กจึงจะสามารถเข้า application เพื่อเลือกสินค้าที่ต้องการและรอแจ้งเตือนร้านที่กำลังไลฟ์สดได้

8. เมื่อผู้ใช้งานกดปุ่ม “chat” ในหน้ารายละเอียดสินค้าเพื่อไปยังแชท ของร้านค้าและผู้ใช้งานสามารถกดปุ่ม “buy” ในหน้ารายละเอียดสินค้าเพื่อไปยังแชท ของร้านค้าและจะมีการนำรหัสสินค้า ชื่อสินค้า ราคาสินค้า ส่งไปให้ร้านค้าอัตโนมัติ

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ช่วยให้ผู้ใช้งานที่เป็นผู้ค้าขายมีช่องทางการขายสินค้ามากขึ้น

1.5.2 ช่วยให้ผู้ใช้งานที่เป็นลูกค้ามีความสะดวกมากขึ้น

## 1.6 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

### 1.6.1 ฮาร์ดแวร์

คอมพิวเตอร์ 2 เครื่อง มีคุณสมบัติดังนี้

1. Acer Nitro5
  - ระบบปฏิบัติการ Windows 11
  - AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
  - Installed RAM 16.0 GB (13.9 GB usable)
2. ROG STRIX
  - ระบบปฏิบัติการ Windows 11
  - Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
  - Installed RAM 16.0 GB (15.9 GB usable)

### 1.6.2 ซอฟต์แวร์

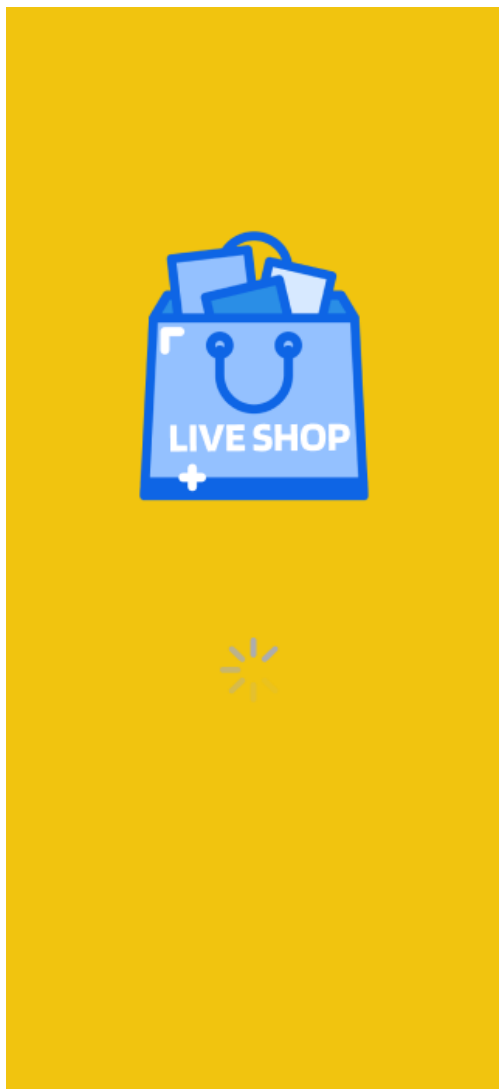
1. Python ใช้สำหรับการพัฒนาโปรแกรม
2. Tesseract ใช้สำหรับการประมวลผลภาพ
3. firebase ใช้เป็นฐานข้อมูลในการเก็บข้อมูลที่ใช้ระหว่างพัฒนาโปรแกรม
4. Flutter เป็น Framework หลักสำหรับการพัฒนาแอปพลิเคชัน
5. Android Studio ใช้สำหรับการสร้างแอปพลิเคชัน

## ตารางที่ 1 แผนการดำเนินงาน

[illegible]

## 1.8 ตัวอย่างโปรแกรม

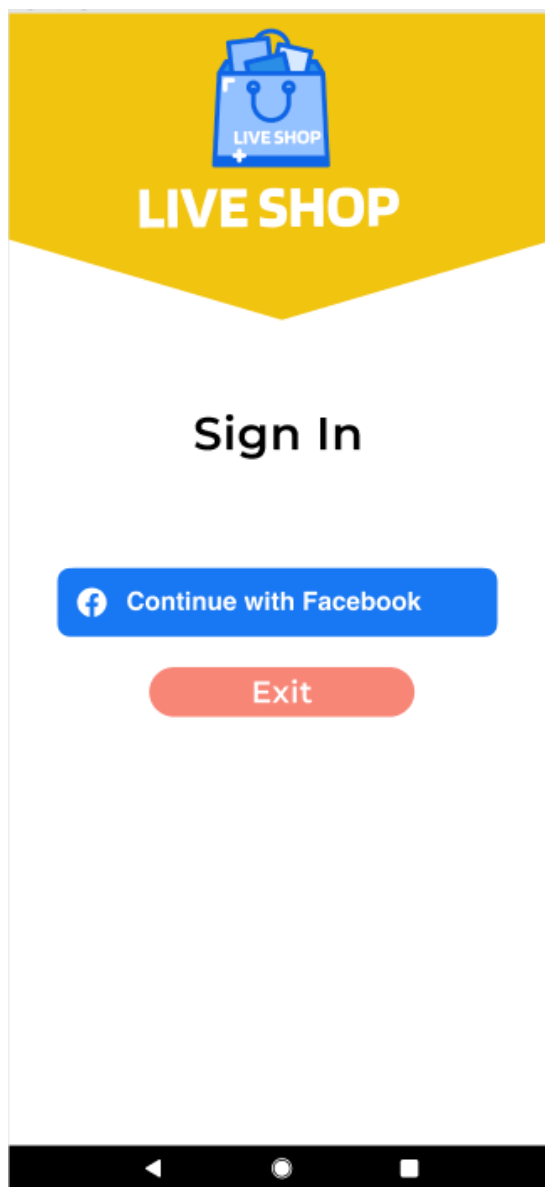
### 1.8.1 แอปพลิเคชันบนอุปกรณ์ไร้สายแบบเคลื่อนที่



ภาพประกอบที่ 1.3 หน้าแรกของโปรแกรม

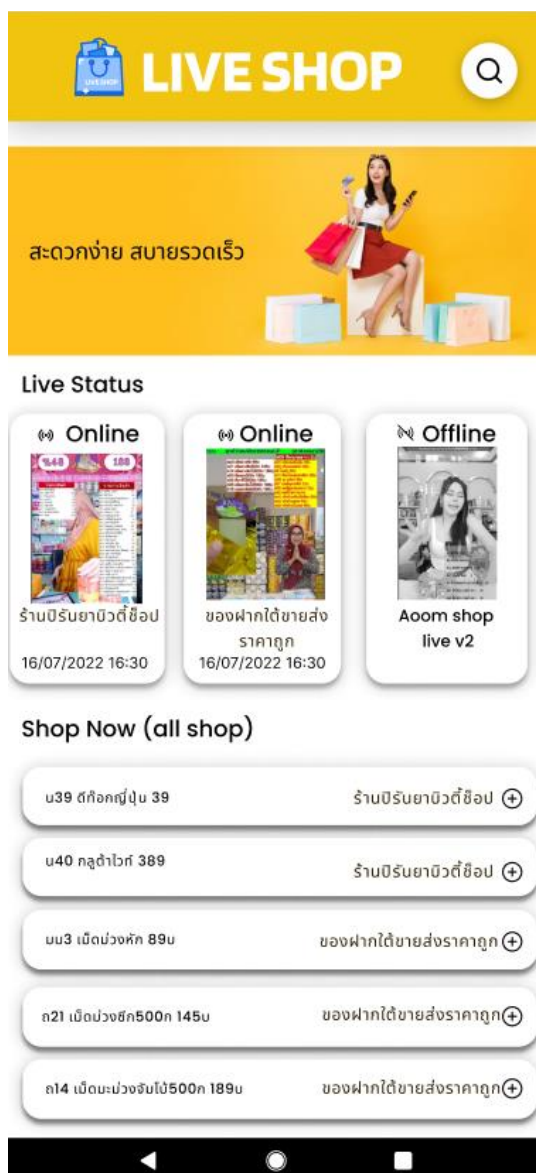
หน้าแรกของโปรแกรม จะมีการรอโหลดข้อมูลจาก Database เมื่อทำการโหลดข้อมูลเสร็จจะนำพาไปยังหน้าล็อกอินทันที





ภาพประกอบที่ 1.4 หน้าล็อกอินเข้าสู่ระบบ

หน้าล็อกอินเข้าสู่ระบบ จะมีปุ่ม 2 ปุ่ม “Continue with Facebook” เพื่อเข้าสู่ระบบผ่านแอปพลิเคชันเฟซบุ๊กเมื่อเข้าสู่ระบบสำเร็จจะนำไปยังหน้าหลักของแอปพลิเคชันและ ปุ่ม “Exit” เพื่อออกจากแอปพลิเคชัน



ภาพประกอบที่ 1.5 หน้าหลักของแอปพลิเคชัน

หน้าหลักของแอปพลิเคชัน จะมีปุ่ม ”ค้นหา” กดเพื่อไปยังหน้าค้นหาสินค้า และ จะมีแถบ “Live Status” แสดงสถานะการไลฟ์สดของร้านทั้งสามร้านว่าร้านไหนกำลังออนไลน์หรือออฟไลน์อยู่ และวันที่อัปเดตรูปภาพและจะมีแถบ ”Shop Now” แสดงรายการสินค้าของร้านที่กำลังไลฟ์สดอยู่ทั้งหมด



ภาพประกอบที่ 1.6 หน้าค้นหาสินค้า

หน้าค้นหาสินค้า จะมีช่องเพื่อใส่ข้อมูลชื่อของสินค้าหรือชื่อร้านค้าที่ต้องการค้นหา และเมื่อใส่ข้อมูลลงไปจะมีส่วนในการแสดงผลลัพธ์ ของการค้นหาเป็นรายการสินค้าขึ้นมาและเมื่อเลือกรายการสินค้าจะไปยังหน้ารายละเอียดสินค้าทันที



ภาพประกอบที่ 1.7 หน้ารายละเอียดสินค้า

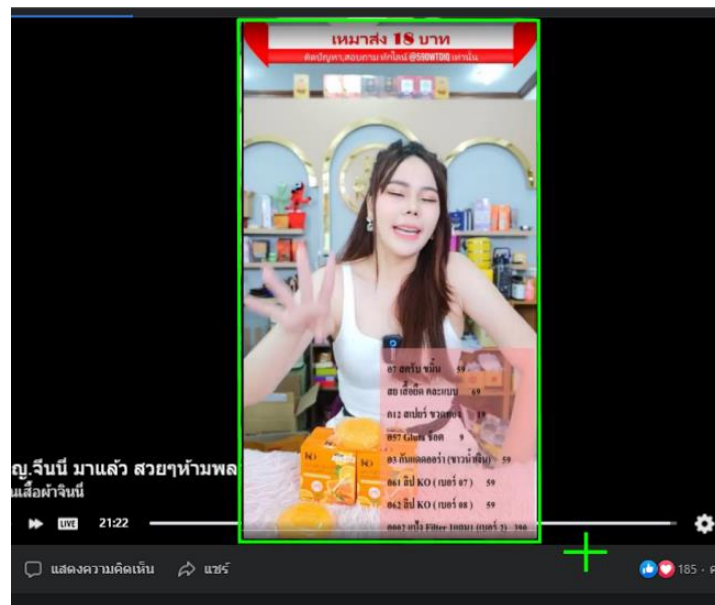
หน้ารายละเอียดสินค้า จะแสดงรูปภาพที่แคปมาจากไลฟ์สดและวันเวลาทวิตเดทรูปภาพและแสดงชื่อร้านค้า ข้อมูลของสินค้าประกอบไปด้วย รหัสสินค้า ชื่อสินค้า ราคา และจะมีปุ่ม 2 ปุ่ม “Chat” เพื่อลิงไปยังแชทของร้านค้าและ “Buy” เพื่อลิงไปยังแชทของร้านค้าและนำข้อมูลสินค้าส่งไปอัตโนมัติ

### 1.8.2 แอปพลิเคชันบนคอมพิวเตอร์สำหรับผู้ดูแลระบบ



ภาพประกอบที่ 1.8 หน้าแรกโปรแกรม

หน้าแรกโปรแกรม มีพื้นที่ในการแสดงผลสำหรับสตรีม 3 ส่วน และมีปุ่ม 3 ปุ่ม “capture” เพื่อเลือกกรอบจากหน้าไลฟ์สดที่ต้องการนำมาประมวลผล ปุ่ม “clear” เพื่อทำการลบสตรีมที่ต้องการยกเลิกประมวลผล และปุ่ม “detail” เพื่อทำการแสดงข้อมูลรหัสสินค้า ชื่อสินค้า และราคาสินค้า ที่ระบบทำการประมวลผลแปลงรูปภาพเป็นข้อความออกมาโชว์เป็น Popup ที่บริเวณหน้าจอ



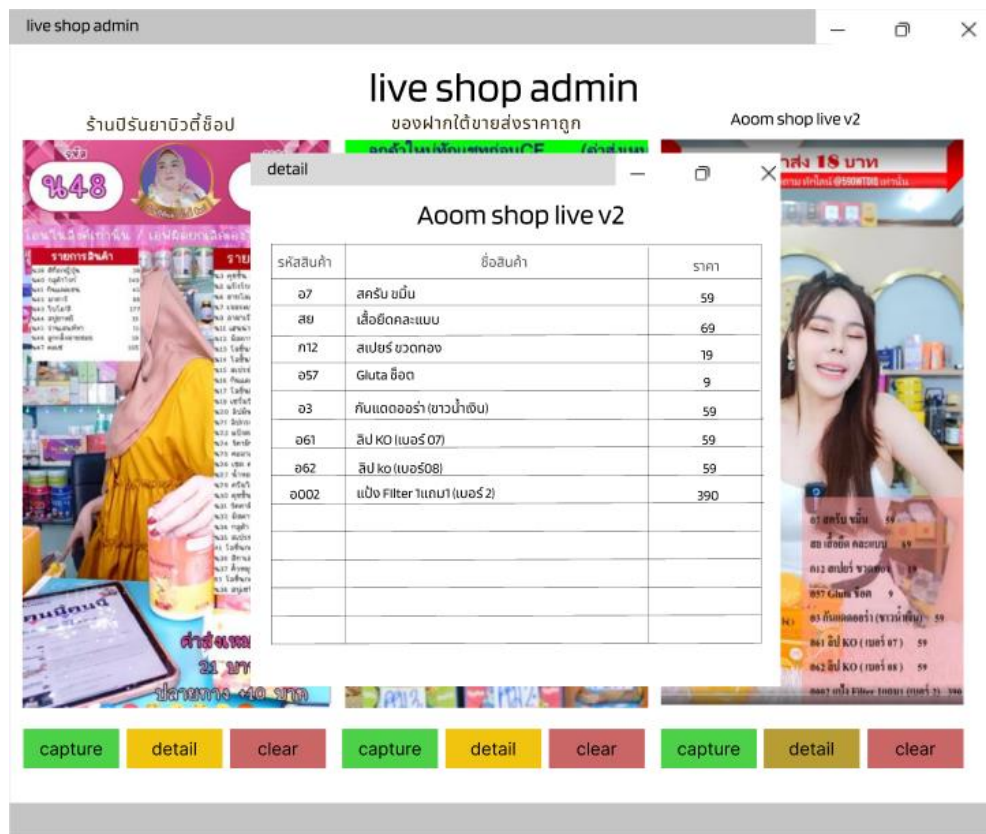
ภาพประกอบที่ 1.9 หน้าเลือกพื้นที่ภาพ

เมื่อกดปุ่ม “capture” แล้วต้องลาก cursor ครอบพื้นที่ที่ต้องการสตรีมจากไลฟ์สดเพื่อนำไปประมวลผล



ภาพประกอบที่ 1.10 หน้าผลลัพธ์ในการถ่ายภาพหน้าจอ

ผลลัพธ์ในการถ่ายภาพหน้าจอ ของปุ่ม Capture



ภาพประกอบที่ 1.11 หน้ารายละเอียดข้อมูล

หน้ารายละเอียดข้อมูล เมื่อกดที่ปุ่ม "detail" ในส่วนของสตรีมมิ่งนั้นๆจะทำการแสดงข้อมูลรหัสสินค้า ชื่อสินค้า และราคาสินค้า ที่ระบบทำการประมวลผลแปลงรูปภาพเป็นข้อความ ออกมาโชว์เป็น Popup ที่บริเวณหน้าจอ

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 Flutter

Flutter [1] คือ Framework ที่ใช้สร้าง UI สำหรับ mobile application ที่สามารถทำงานได้ทั้ง iOS และ Android ในเวลาเดียวกันโดยที่ใช้ source code ตัวเดียวกัน โดยภาษาที่ใช้ใน Flutter นั้นจะเป็นภาษา dart ซึ่งถูกพัฒนาโดย Google และ ยังเป็น open source ที่สามารถใช้งานได้ฟรี ตัวอย่าง syntax ของภาษา dart ที่ใช้ใน Flutter ซึ่งจะมีความคล้ายกับภาษา Java เนื่องจาก dart เป็นภาษาที่รองรับ OOP และมีแนวคิด เช่นเดียวกับภาษา Java

```
1  import 'package:flutter/material.dart';
2
3  void main() {
4    runApp(new MaterialApp(
5      home: new MyApp(),
6    ));
7  }
8
9  class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return new Scaffold(
13       appBar: new AppBar(
14         title: new Text("Example App"),
15         backgroundColor: Colors.blue,
16       ),
17       backgroundColor: Colors.blue,
18       body: new Center(
19         child: new Column(
20           mainAxisAlignment: MainAxisAlignment.center,
21           children: <Widget>[
22             new Icon(Icons.favorite, color: Colors.redAccent, size: 200.0,
23             ),
24           ],
25         ),
26       ),
27     );
28   }
29 }
```

ภาพประกอบที่ 2.1 ตัวอย่างโครงสร้าง Flutter

ซึ่งหากสังเกตจาก จะเห็นว่า Flutter นั้นจะมี Widget พื้นฐานมาให้ เพื่อให้การออกแบบ UI มีความง่าย และสะดวกยิ่งขึ้น โดย Widget พื้นฐานของ Flutter หลัก ๆ จะมีอยู่ 2 ชนิดคือ StatelessWidget และ StatefulWidget โดยที่ StatelessWidget จะใช้สร้าง Widget ที่ไม่มีการจัดการสถานะการทำงานใดหรือหน้านั้นๆจะไม่มีเปลี่ยนแปลง เช่น การแสดงข้อความ,Iconหรือ



รูปภาพที่ไม่มี animation เข้ามาเกี่ยวข้องเป็นต้น ส่วน StatefulWidget จะใช้สร้าง Widget ที่มีการจัดการสถานะการทำงานต่างๆหรือมีปุ่มที่มี action เช่น การสร้าง Icon ที่มีการใส่ animation ให้สามารถขยับไปมาได้, ปุ่มกดต่างๆ บนหน้า UI เป็นต้น

จุดเด่นหลัก ๆ ของ Flutter คือ ระบบ Hot Reload โดยเมื่อมีการทดสอบ, การสร้าง, การ add features หรือการกระทำต่าง ๆ กับ UI จะต้องมีการ reload เพื่อให้หน้า UI update ซึ่งระบบ Hot Reload จะเข้ามาช่วยในส่วนของการ reload โดยจุดเด่นของระบบนี้คือการย่นระยะเวลาที่ใช้ในการ reload ให้เหลือเพียงเสี้ยววินาทีเท่านั้น ทำให้การพัฒนา UI ของ application มีความรวดเร็วขึ้นอย่างมาก และยังมีจุดเด่นอื่น ๆ ที่ช่วยให้การพัฒนาเป็นไปได้ง่ายขึ้นไม่ว่าจะเป็น Build-In ที่ช่วยในการออกแบบ UI ให้มีความสวยงามยิ่งขึ้นอย่าง Material Design และ Cupertino (iOS-flavor), มี Framework ที่ช่วยให้การทำ animation ต่าง ๆ หรือ gesture ของ UI เป็นเรื่องง่ายยิ่งขึ้น และยังสามารถใช้งานร่วมกับ IDE ที่กำลังเป็นที่นิยมอยู่ในปัจจุบันอย่าง VS Code และ Android Studio ได้

ข้อเสียหลัก ๆ คือ การใช้ภาษา dart ในการเขียน ซึ่งคนส่วนใหญ่อาจจะยังไม่คุ้นเคยกับ syntax ของภาษา dart ประกอบกับ community ยังเล็กเนื่องจาก Flutter ยังเปิดตัวมาได้ไม่นานนัก เมื่อเทียบกับ Framework ตัวอื่น ๆ อย่าง React Native ที่มี community ค่อนข้างใหญ่จึงทำให้ document ต่าง ๆ ยังไม่เยอะเท่าที่ควร ทำให้เวลามีปัญหาเกี่ยวกับการใช้งานอาจจะต้องใช้เวลาในการหาวิธีแก้

### 2.1.2 Firebase

เป็นหนึ่งในผลิตภัณฑ์ของ Google โดย Firebase [2] คือ Platform ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วนของการเก็บข้อมูล ซึ่งทำให้สามารถ สร้าง Mobile Application ได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรี และเครื่องมือที่มีค่าใช้จ่าย Firebase มีบริการให้ใช้หลายอย่าง สามารถแบ่งเป็นหมวดหมู่ดังนี้

#### 1.) Better Apps

- Cloud Firestore จัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้และอุปกรณ์ในระดับโลกโดยใช้ฐานข้อมูล NoSQL ที่โฮสต์บนคลาวด์ Cloud Firestore ให้การซิงโครไนซ์แบบสดและการสนับสนุนออฟไลน์พร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ การผสมผสานกับผลิตภัณฑ์ Firebase อื่นๆ ช่วยให้คุณสามารถสร้าง แอปแบบไร้เซิร์ฟเวอร์ได้อย่างแท้จริง

- Authentication จัดการผู้ใช้ของคุณด้วยวิธีที่ง่ายและปลอดภัย Firebase Auth มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึง Email และ password ผู้ให้บริการบุคคลที่สามเช่น Google หรือ Facebook และใช้ระบบบัญชีที่คุณมีอยู่โดยตรง สร้างอินเทอร์เฟซของคุณเองหรือใช้ประโยชน์จากโอเพ่นซอร์ส UI ที่ปรับแต่งได้อย่างเต็มที่

- Hosting ลดความซับซ้อนของเว็บโฮสติ้งของคุณด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปสมัยใหม่ เมื่อคุณอัปโหลดเนื้อหาเว็บของคุณเราจะส่งเนื้อหาเหล่านั้นไปยัง CDN ทั่วโลกของเราโดยอัตโนมัติและมอบใบรับรอง SSL ฟรีเพื่อให้ผู้ใช้ของคุณได้รับประสบการณ์ที่ปลอดภัยเชื่อถือได้และมีเวลาแฝงต่ำไม่ว่าจะอยู่ที่ใดก็ตาม

- Realtime Database Realtime Database คือฐานข้อมูลดั้งเดิมของ Firebase เป็นโซลูชันที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปบนอุปกรณ์เคลื่อนที่ที่ต้องการสถานะการชิงคร่ระหว่างไคลเอนต์แบบเรียลไทม์ เราขอแนะนำ Cloud Firestore แทน Realtime Database สำหรับนักพัฒนาส่วนใหญ่ที่เริ่มโปรเจกต์ใหม่

## 2.) Improve app quality

- Crashlytics ลดเวลาในการแก้ไขปัญหาของคุณด้วยการเปลี่ยนข้อขัดข้องจากหิมะถล่มให้เป็นรายการปัญหาที่จัดการได้ รับข้อมูลเชิงลึกที่ชัดเจนและนำไปปฏิบัติได้ว่าปัญหาใดที่ต้องจัดการก่อน โดยเห็นผลกระทบของผู้ใช้ในแดชบอร์ด Crashlytics การแจ้งเตือนแบบเรียลไทม์จะช่วยให้คุณมีความเสถียรแม้ในขณะที่เดินทาง Crashlytics เป็นตัวรายงานข้อขัดข้องหลักของ Firebase

- Performance Monitoring วินิจฉัยปัญหาประสิทธิภาพของแอปที่เกิดขึ้นบนอุปกรณ์ของผู้ใช้ ใช้การติดตามเพื่อตรวจสอบประสิทธิภาพของบางส่วนของแอปและดูมุมมองสรุปในคอนโซล Firebase อยู่เหนือเวลาเริ่มต้นของแอปและตรวจสอบคำขอ HTTP โดยไม่ต้องเขียนโค้ดใด ๆ

- Test Lab เรียกใช้การทดสอบอัตโนมัติและกำหนดเองสำหรับแอปของคุณบนอุปกรณ์เสมือนและจริงที่โฮสต์โดย Google ใช้ Firebase Test Lab ตลอดวงจรการพัฒนาของคุณเพื่อค้นหาจุดบกพร่องและความไม่สอดคล้องกันเพื่อให้คุณสามารถนำเสนอประสบการณ์ที่ยืดเยื้อบนอุปกรณ์หลากหลายประเภท

## 3.) Grow your business

- Google Analytics วิเคราะห์คุณลักษณะและพฤติกรรมของผู้ใช้ในแดชบอร์ดเดียวเพื่อทำการตัดสินใจอย่างชาญฉลาดเกี่ยวกับแผนงานผลิตภัณฑ์ของคุณ รับข้อมูลเชิงลึกแบบเรียลไทม์จากรายงานหรือส่งออกข้อมูลเหตุการณ์ดิบไปยัง Google BigQuery สำหรับการวิเคราะห์ที่กำหนดเอง

- Remote Config กำหนดวิธีการแสดงผลแอปของคุณสำหรับผู้ใช้แต่ละคน เปลี่ยนรูปแบบเปิดตัวฟีเจอร์ที่ละเอียดน้อยเรียกใช้การทดสอบ A / B ส่งมอบเนื้อหาที่กำหนดเองให้กับผู้ใช้บางรายหรือทำการอัปเดตอื่น ๆ โดยไม่ต้องปรับใช้เวอร์ชันใหม่ทั้งหมดนี้ทำได้จากคอนโซล Firebase ตรวจสอบผลกระทบของการเปลี่ยนแปลงของคุณและทำการปรับเปลี่ยนในเวลาไม่กี่นาทีก่อน

- Cloud Messaging ส่งข้อความและการแจ้งเตือนไปยังผู้ใช้ข้ามแพลตฟอร์มทั้ง Android, iOS และเว็บได้ฟรี สามารถส่งข้อความไปยังอุปกรณ์เดี่ยวกลุ่มอุปกรณ์หรือหัวข้อเฉพาะหรือกลุ่มผู้ใช้ Firebase Cloud Messaging (FCM) ปรับขนาดเป็นแอปที่ใหญ่ที่สุดโดยส่งข้อความหลายแสนล้านข้อความต่อวัน

### 2.1.3 การประมวลผลภาพ (Image Processing)

การประมวลผลภาพ (Image Processing) [3]เป็นการนำภาพมาประมวลผลหรือคิดคำนวณด้วย คอมพิวเตอร์แล้วใช้กรรมวิธีใดๆมากระทำกับข้อมูลภาพเพื่อให้ได้ภาพที่มีคุณสมบัติตามต้องการทั้งในเชิง คุณภาพและปริมาณ โดยมีขั้นตอนต่างๆที่สำคัญ คือ การทำให้ภาพมีความคมชัดมากขึ้น การกำจัดสัญญาณรบกวนออกจากภาพ การแบ่งส่วนของวัตถุที่สนใจออกมาจากภาพ เพื่อนำภาพวัตถุที่ได้ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ

คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมากในเวลาอันสั้นจึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบ ลักษณะของการใช้งานจะมีเป็น

### 1. การรู้จำภาพ (Image Recognition)

การรู้จำภาพ (Image Recognition) การรู้จำภาพเป็นกระบวนการในการระบุและการตรวจจับวัตถุหรือคุณสมบัติที่มีอยู่ในระบบดิจิทัล ซึ่งได้จากภาพหรือวิดีโอแนวคิดนี้ได้ถูกนำมาใช้ในแอปพลิเคชันมากมายที่เหมือนกันในระบบสำหรับการทำงานของระบบอัตโนมัติในโรงงานการตรวจสอบด้านเก็บเงินด้วยการรักษาความปลอดภัยและการรักษาความปลอดภัย อัลกอริทึมการรู้จำภาพทั่วไป

### 2. การสกัดคุณลักษณะเด่น (Feature Extraction)

การสกัดคุณลักษณะเด่น (Feature Extraction) เป็นการหาคุณลักษณะเด่นที่ได้จากภาพ โดยผลลัพธ์ที่ได้จากการนำไปหาคุณลักษณะเด่นจะบ่งบอกถึงลักษณะเช่น ความสูง ความยาว สีหรือเวกเตอร์ที่เป็นคุณลักษณะ ที่สำคัญของรูปภาพ เป็นต้น นอกเหนือจากข้อมูลคุณลักษณะดังกล่าว ในปัจจุบันยังมีวิธีการการสกัดหา ลักษณะเด่นที่นิยมใช้กันอย่างแพร่หลายที่ใช้ในการต่อภาพคือ การวางตัวของเส้นขอบ(Edge) การหามุม (Corner) และการหาพื้นที่น่าสนใจ (Blob) เป็นต้น



ภาพประกอบที่ 2.2 การตรวจจับและประมวลผลใบหน้าจากภาพ [4]

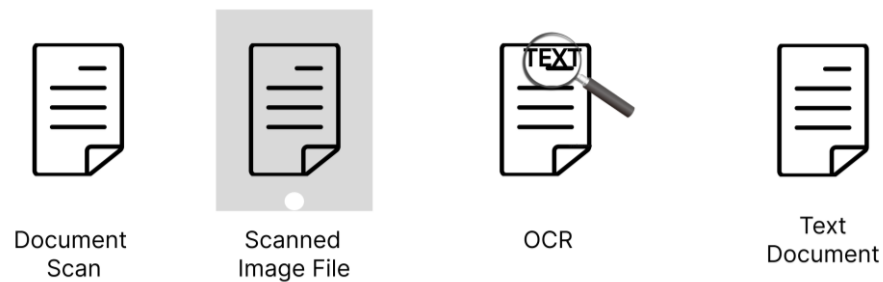
ตัวอย่างการ นำการประมวลผลภาพไปใช้งาน เช่น

- ระบบตรวจจับใบหน้าในกล้องดิจิทัล โดยกล้องจะมีระบบตรวจว่าส่วนไหนของภาพมี ลักษณะคล้าย ใบหน้า แล้วกล้องก็จะทำการโฟกัสตำแหน่งที่ตรวจจับเพื่อภาพมีความคมชัดมากขึ้น เช่น ระยะห่าง ระหว่างคิวมูปาก โหนกแก้ม จมูก โครงหน้า

- ระบบตรวจจับการเคลื่อนไหวเพื่อรักษาความปลอดภัย โดยกล้องจะเปรียบเทียบภาพก่อน หน้าและภาพปัจจุบันถ้ามีส่วนใดส่วนใดเปลี่ยนแปลงระบบจะบันทึกเฉพาะภาพที่มีการเปลี่ยนแปลงทำให้ประหยัด เนื้อที่ในการเก็บภาพและสามารถตรวจสอบได้ภาพหลัง
- ระบบอ่านคิวบาร์โค้ด โดยระบบนี้จะอ่านรหัสจากบาร์โค้ด แล้วแปลงเป็นข้อมูลซึ่งสามารถ นำมาใช้ งานได้ เช่น QR code, Microsoft tag ภาพที่นำมาประมวลผลนั้นเป็นได้ทั้งภาพนิ่งและ ภาพเคลื่อนไหว ในการนำภาพนิ่งเข้ามาประมวลผลนั้น โปรแกรมจะอ่านไฟล์ภาพขึ้นมาแล้วถอดรหัสจาก ค่าสีของภาพ แต่ละจุดเป็นตัวเลขเพื่อนำมาประมวลผลสำหรับการประมวลผลภาพเคลื่อนไหวนั้น โปรแกรมจะมองเห็นเป็นภาพนิ่งหลายภาพที่เรียงต่อกัน จึงมีประโยชน์อย่าง มากในการเพิ่ม ประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลได้จากภาพในระบบต่าง ๆดังกล่าวข้างต้น การประมวลผลภาพ สามารถจำแนกตาม วัตถุประสงค์ในการใช้งานดังนี้
- การปรับปรุงคุณภาพของภาพ (Image Enhancement and Restoration) เช่น ปรับภาพ ให้คมชัดมากขึ้นการปรับ Contrast หรือการปรับเน้นเส้นขอบภาพ (Edge Enhancement) หรือการ กรองสัญญาณรบกวน (Image Filtering) เพื่อกำจัดสัญญาณรบกวนได้
- การบีบอัดภาพ (Image Compression) เนื่องจากข้อมูลภาพนั้นมีขนาดใหญ่มากโดยเฉพาะ ภาพสี ซึ่งจะทำให้การจัดเก็บหรือรับส่งไฟล์ข้อมูลขนาดใหญ่เสียเวลามาก เพื่อทำให้ขนาดภาพเล็กลงจึง ทำให้ การบีบอัด ข้อมูลภาพ เช่น JPEG หรือ GIF และตัวอย่างการบีบอัดข้อมูลวีดีโอ เช่น MPEG หรือ AVI เป็นต้น
- การวิเคราะห์ข้อมูลภาพ (Image Analysis) เป็นการสังเคราะห์ ข้อมูลที่มีความหมายจากภาพ แล้วนำไปใช้งาน ซึ่งอาจต้องมีการรู้จำวัตถุ (object recognition) การแยกส่วนวัตถุ 317 The Tenth National conference on Computing and Information Technology NCCIT4 (segmentation) ตรวจจับ การ เคลื่อนไหวที่ของวัตถุ (motion detection)

#### 2.1.4 OCR หรือ Optical Character Recognition

OCR หรือ Optical Character Recognition [4] หรือ "การรู้จำอักขระด้วยแสง" เป็นเทคโนโลยีที่แปลงเอกสาร รูปภาพ หรือแม้แต่ ลายเซ็นอิเล็กทรอนิกส์ ให้อยู่ในรูปแบบตัวอักษร หรือข้อความ (Plain Text) ไฟล์ดิจิทัลที่สามารถสืบค้นได้ด้วยคำค้นหา (Keyword) ยกตัวอย่าง ไฟล์เอกสารที่ มาในรูปแบบ PDF มีข้อดีก็คือไม่สามารถปลอมแปลง แก้ไขได้ แต่ถ้าต้องการแก้ไขข้อความใด ๆ ในนั้นก็ เป็นเรื่องยากลำบาก วิธีแก้ไขปัญหาก็คือ แปลงไฟล์ PDF ให้เป็น OCR ซึ่งวิธีการที่ทุกคนคุ้นเคยก็คือ การนำแผ่นเอกสาร มาสแกนข้อความ ตัวอักษร เส้นตารางและอื่น ๆ ให้เป็นไฟล์ Word หรือ PDF ด้วย เครื่องสแกนเนอร์ หรือกล้องถ่ายภาพ



### ภาพประกอบที่ 2.3 ขบวนการการทำงานของ Optical Character Recognition

ส่วนการทำงานของ OCR นั้น หลัก ๆ คือ ใช้การจดจำรูปแบบ เพื่อกำหนดอักขระของแต่ละประเภทไฟล์ จากนั้น ซอฟต์แวร์จะทำการอ่านข้อความและอักขระ แล้วแปลงเป็นไฟล์ที่สืบค้นได้นอกจากนี้ การทำงานของ OCR ยังขึ้นอยู่กับซอฟต์แวร์ที่ใช้ประมวลผล เพราะนอกเหนือจากการแปลงรูปภาพ อักขระใด ๆ เป็นข้อความแล้ว ซอฟต์แวร์บางตัวสามารถจัดวางรูปแบบข้อความในไฟล์ OCR ได้

#### 2.1.1 Object Detection

Object Detection [5] การตรวจจับวัตถุ คือ เทคโนโลยีในทางคอมพิวเตอร์ หลักการที่เกี่ยวกับ Computer Vision และ Image Processing ที่ใช้ในงาน AI ตรวจจับวัตถุชนิดที่กำหนด เช่น มนุษย์ รถยนต์ อาคาร ที่อยู่ในรูปภาพ หรือวิดีโอ งาน Object Detection การตรวจจับวัตถุในรูปภาพสามารถเจาะลึกลงไปได้อีกหลายแขนง เช่น การทำ ตรวจจับหน้าคน ตรวจจับคนเดินถนน สามารถประยุกต์ใช้ได้หลากหลาย เช่น ใช้ในงานรักษาความปลอดภัย และรถยนต์ไร้คนขับ เป็นต้น

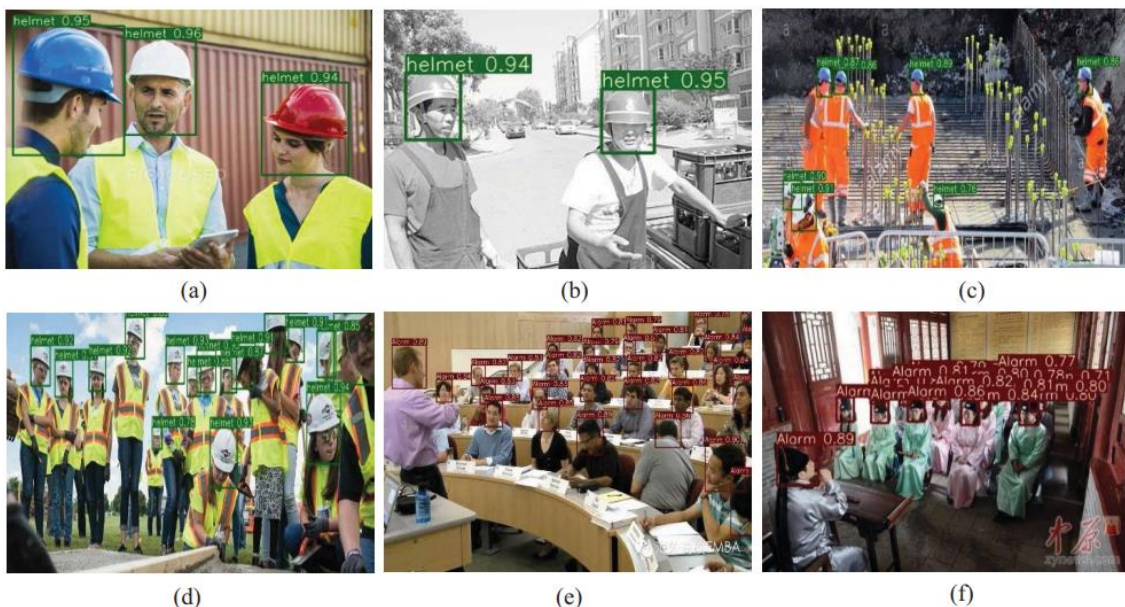
#### 2.1.1 YOLO (You Only Look Once)

เป็นอัลกอริทึม [6] ที่นำแนวความคิดของการทำนายตำแหน่งและขนาดของ กล่องจากความน่าจะเป็นที่กล่องนั้น จะเป็นกรอบล้อม วัตถุ แต่สิ่งที่ YOLO มีความสามารถและความเร็วเหนือกว่าอัลกอริทึมอื่น เช่น Faster R-CNN ที่มีการ ทำงานในลักษณะที่จะทำนายตำแหน่งของกรอบล้อม วัตถุและค่อยนำวัตถุในกล่องนั้น ไปผ่านแบบจำลองเพื่อ ทำนายวัตถุในกล่อง แต่ YOLO นั้นจะทำนายทั้งกรอบ ล้อมวัตถุ และความน่าจะเป็นของวัตถุบางส่วนที่อยู่ใน กรอบออกมาพร้อมกันทีเดียว YOLO จัดว่าเป็นเทคนิคการตรวจจับวัตถุในภาพซึ่ง เป็นซอฟต์แวร์เปิดสำหรับงานปัญญาประดิษฐ์แบบโครงข่ายประสาท (Neural Network) ที่พัฒนาด้วย โปรแกรมภาษา C++ และสามารถทำงานบนหน่วยประมวลผล CUDA ของ GPU ได้เป็นอย่างดี เหมาะกับ การประมวลผลภาพแบบ Real Time ภาพจากกล้องหรือ วิดีโอ ปัจจุบัน YOLO มี การพัฒนามาแล้ว 5 เวอร์ชันปัจจุบันคือ YOLO v5 โครงสร้างโครงข่ายประสาทเทียมของ YOLO แต่ละ เวอร์ชันจะมี Convolution Box ที่ แตกต่างกัน แต่หลักการโดยทั่วไปแล้ว YOLO จะแบ่งภาพออกเป็น Grid Cell เล็กๆ และแต่ละ Grid Cell จะถูกทำนายผ่าน แบบจำลอง เพื่อหาตำแหน่งจุดกึ่งกลางของวัตถุ และ ความน่าจะเป็นที่จะมีวัตถุใด ใน Grid Cell

## 2.2 ระบบงานที่เกี่ยวข้อง

### 2.2.1 Safety Helmet Detection Based on YOLOv5

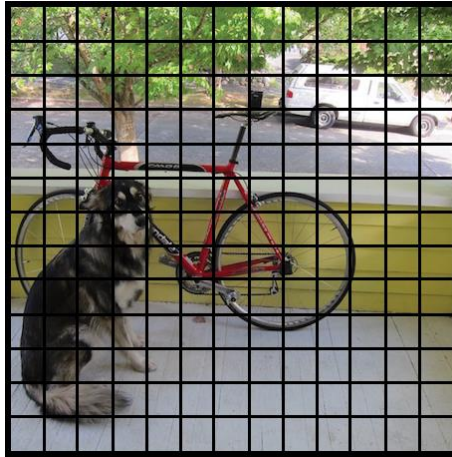
"Safety Helmet Detection Based on YOLOv5" [7]เป็นงานวิจัยของ Fangbo Zhou, Huailin Zhao, Zhen Nie มหาวิทยาลัย Shanghai Institute of Technology [7]ที่พัฒนาระบบตรวจสอบความปลอดภัยของพนักงาน ซึ่งระบบจะทำการตรวจจับภาพของพนักงาน และ จะแสดงกรอบที่มีคำอธิบาย โดยได้นำรูปภาพจากอินเทอร์เน็ตจำนวน 6045 ภาพมาทำการทดสอบ โดยสีระของพนักงานที่ไม่สวมหมวกนิรภัยจะมีคำอธิบายว่า"Alarm" และสำหรับสีระของพนักงานที่สวมหมวกนิรภัยจะมีคำอธิบายว่า"Helmet" โดยใช้ algorithm ที่ทันสมัยที่สุดในการตรวจสอบ algorithm งานตรวจจับวัตถุถูกใช้อย่างแพร่หลายในความเป็นจริง เป้าหมายของการตรวจจับคือการค้นหวัตถุทั้งหมดที่น่าสนใจในภาพ ซึ่งจะมี 2 งานย่อยคือการกำหนดหมวดหมู่และการระบุตำแหน่งของวัตถุ แม้ว่าอัลกอริธึมการตรวจจับแบบดั้งเดิมสามารถทำงานได้ผลลัพธ์ที่ดี ในบางสถานการณ์ในสภาพแวดล้อมที่แปรปรวน เช่น การเปลี่ยนแปลงของสภาพอากาศ การกระจายตัวของพนักงานที่ไม่สม่ำเสมอ และหมวกกันน็อคแบบต่างๆ ทำให้ความแม่นยำนั้นรับประกันได้ยากว่ามันสามารถทำงานได้ถูกต้อง จึงได้นำตัวYOLOv5มาใช้ซึ่งมีความเร็วและความแม่นยำสูง



ภาพประกอบที่ 2.4 ภาพการตรวจจับหมวกนิรภัยของพนักงาน [8]

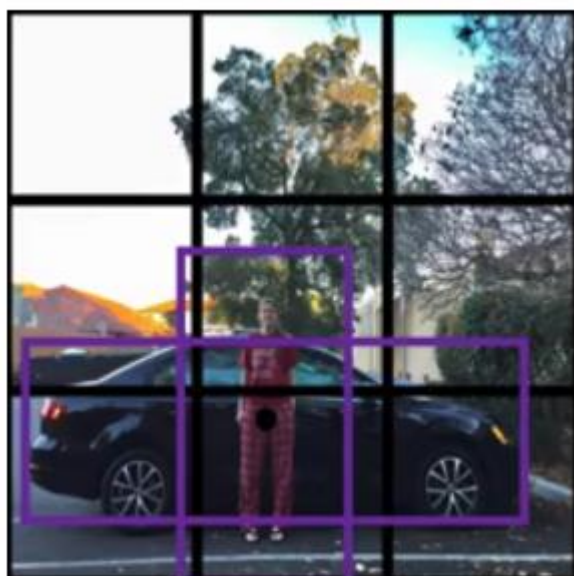


โดยการทำงานของYOLOv5 [8]คือจากรูป 1 รูปเต็มๆ จะทำการแบ่ง Grid cell ออกมาเป็น  $n \times n$  grid ยิ่งแบ่งมากก็จะละเอียด trade off กับการคำนวณ



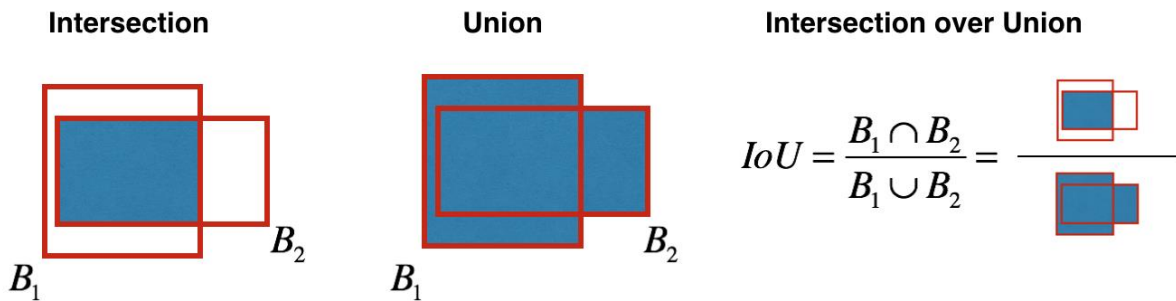
ภาพประกอบที่ 2.5 การแบ่งช่องของรูปภาพออกเป็น grid [9]

และในแต่ละ grid จะมี Label ทุกช่อง เช่น  $[P_c, b_x, b_y, b_h, b_w, c_1, c_2, \dots, c_n]$  โดยที่  $P_c$  คือ ความน่าจะเป็นที่มีวัตถุอยู่ใน Grid นั้นๆ ถ้าไม่มีคือจะได้ค่า 0 ถ้ามีจะได้ค่า 1  $b_x, b_y$  คือตำแหน่งตรงกลางของ Object ว่าอยู่พิกัดไหนส่วน  $b_h, b_w$  คือขนาดความสูงและกว้างของ Object ว่าสูง, กว้างขนาดไหน และ  $c_1, c_2, \dots, c_n$  คือ ผลลัพธ์ว่าเป็น class อะไร ถ้าโจทย์มีแค่ detect หมวกนิรภัย ก็จะมี class เดียว และ grid นั้นมีค่าเป็น 1 แต่ถ้ามีหลาย Object ก็จะมีเลขต่อไป



ภาพประกอบที่ 2.6 ภาพของเทคนิค Anchor Box [9]

จากภาพประกอบที่ 2.6 ภาพของเทคนิค Anchor Box [9] จะรับได้เฉพาะ 1 grid คือ 1 Object แต่กรณีถ้ามีหลาย Object เราจะใช้หลักการที่เรียกว่า Anchor Box โดยเราสามารถกำหนดจำนวน Box นี้ได้ ก็จะมี Label แบบด้านบน 2 อันใน 1 Grid ได้ และตัว YOLO ก็จะคำนวณให้ว่ารูปร่างใกล้เคียง Anchor อันไหนสุดจากค่า IOU ก็จะถูกกำหนดไปที่ Anchor นั้น



IOU จะมีส่วนหลักๆหลายส่วนใน YOLO เช่นเวลาโมเดลทำนายออกมา มันอาจจะทำนายรูป

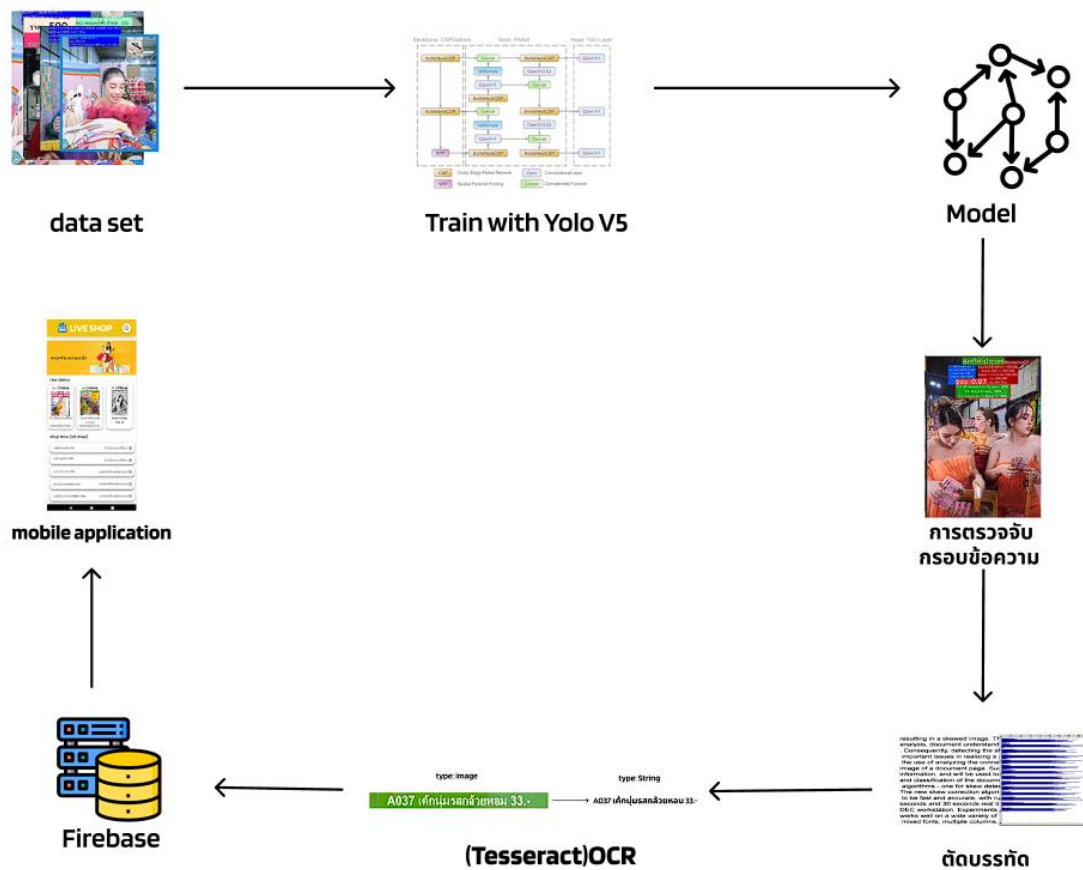
### ภาพประกอบที่ 2.7 การทำงานของขบวนการ IOU [9]

รถเดียวกันแต่หลายกล่องได้ จึงต้องใช้ metric IOU ในเลือกกล่องเดียวเป็นตัวแทนของ Object โดยหลักการคำนวณคือ หาส่วนที่ Intersect หาส่วนที่ Union กัน ถ้ากล่องใดมีค่านี้สูง เกินเกณฑ์ที่กำหนด แสดงว่ามันคือ Object เดียวกัน



### บทที่ 3

#### ขั้นตอนการดำเนินงาน

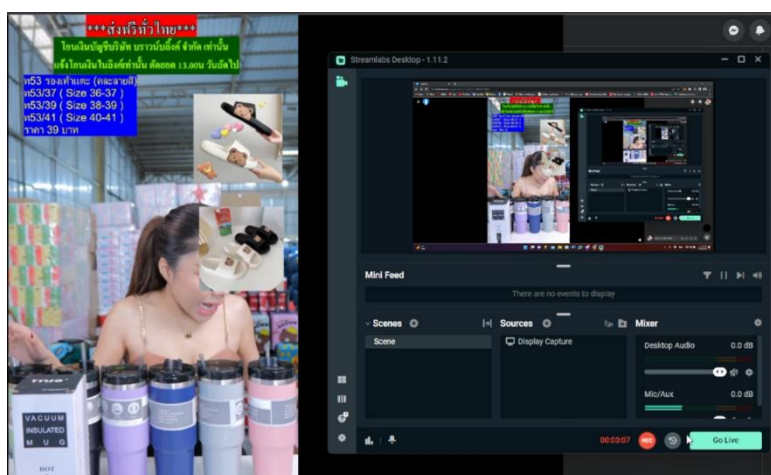


ภาพประกอบที่ 3.1 ขั้นตอนการดำเนินงานของระบบ

สำหรับในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงงานปริญญาโทซึ่งจะทำให้ทราบถึงการวิเคราะห์และการออกแบบแอปพลิเคชันโดยละเอียดว่ามีแนวทางในการดำเนินงานหรือมีขั้นตอนในการทำงานของแอปพลิเคชันอย่างไรบ้างโดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้ การรวบรวมข้อมูล การเทรนโมเดลด้วย YoloV5 การตัดบรรทัดข้อความ การแปลงรูปภาพเป็นตัวอักษร การเก็บข้อมูลลงในฐานข้อมูล และการนำข้อมูลไปแสดงผลใน mobile application

### 3.1 เก็บรวบรวมข้อมูล

เก็บข้อมูลโดยการบันทึกคลิปวิดีโอโดยใช้ Streamlabs Desktop ในการบันทึกคลิปวิดีโอ แล้วนำวิดีโอมาเปิดและบันทึกเป็นภาพนิ่ง ทำการบันทึกวิดีโอจำนวน 3 ร้านค้าได้แก่ 1.ร้าน FIRST SHOP V2 2.ร้านKANYA SHOP ขายถูกทุกอย่าง 3.ร้านมหัศจรรย์"วันของAuuM เนื่องไขในการเก็บ โดยมีการเก็บวิดีโอเป็น 1 ร้านจะมี 3 วิดีโอไลฟ์สด 1 วิดีโอจะแบ่งเป็น 6 คลิปวิดีโอ 1 คลิปวิดีโอ จะแบ่งเป็นคลิปละ 3 นาที



ภาพประกอบที่ 3.2 Streamlabs Desktop

ตารางที่ 2 ตารางจำนวนและเวลาเฉลี่ยของวิดีโอในแต่ละร้านค้า

ชื่อร้านค้า	เวลาเฉลี่ยของวิดีโอ	จำนวน
ร้าน FIRST SHOP V2	3 นาที	18 วิดีโอ
ร้านKANYA SHOP ขายถูกทุกอย่าง	3 นาที	18 วิดีโอ
ร้านมหัศจรรย์"วันของAuuM	3 นาที	18 วิดีโอ

### 3.2 การสร้างโมเดล

#### 3.2.1 การเตรียมข้อมูลในการ training

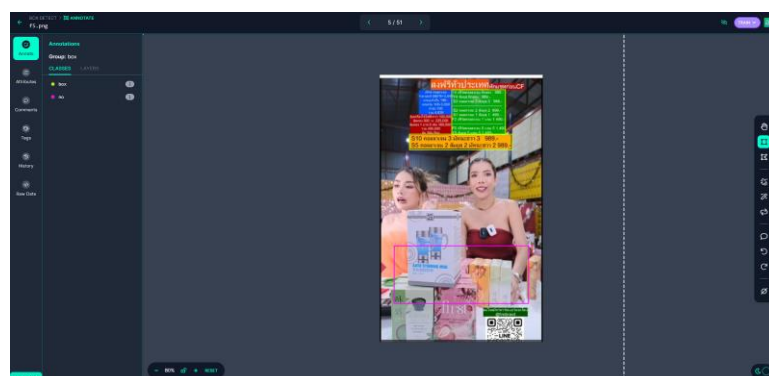
เตรียมข้อมูลในการ training โดยบันทึกภาพจาก Video ไฟล์สดที่เก็บมาได้โดยจะทำการบันทึกเฉพาะช่วงที่มีการเปลี่ยนแปลงของข้อความบนหน้าจอไลฟ์สดดัง โดยจะได้จากร้าน FIRST SHOP V2 จำนวน 33 ภาพ จากร้าน KANYA SHOP ขยายทุกทุกอย่าง จำนวน 28 ภาพ จากร้าน มหัทศจรย์"วันของAuuM" จำนวน 33 ภาพ รวมกันทั้งหมด 92 รูปภาพ



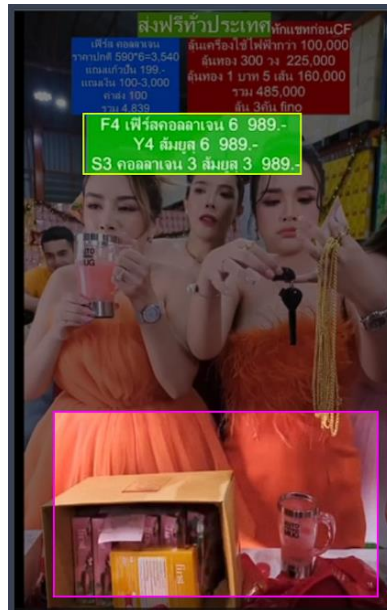
ภาพประกอบที่ 3.3 ตัวอย่างรูปภาพที่ทำการเก็บ

#### 3.2.2 การวาดภาพผลเฉลย

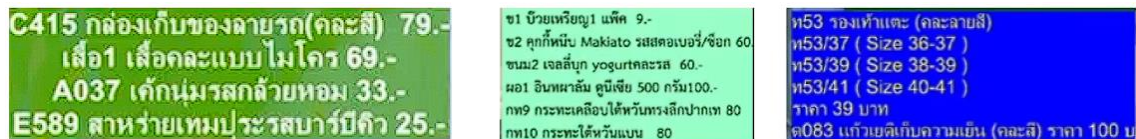
ทำการวาดภาพผลเฉลยรูปภาพผ่าน Roboflow framework สำหรับใช้จัดเก็บ เตรียมชุดข้อมูล และสร้างแบบจำลองต่างๆ ที่สามารถใช้งานผ่าน web browser โดยจะทำการวาดภาพผลเฉลยในรูปภาพที่เตรียมมาทั้งหมด โดยจะจัดหมวดหมู่ในที่ที่สนใจเป็น “box” และในพื้นที่ที่ไม่สนใจเป็น “no”



ภาพประกอบที่ 3.4 การวาดภาพผลเฉลยรูปภาพใน Roboflow



ภาพประกอบที่ 3.5 การวาดภาพผลเฉลยรูปภาพใน Roboflow (2)



ภาพประกอบที่ 3.6 ภาพผลเฉลยที่ถูกต้องจากทั้ง 3 ร้านค้า

เมื่อทำการวาดภาพผลเฉลยรูปภาพทั้งหมดจะทำการแบ่งส่วนของข้อมูลออกเป็นดังนี้

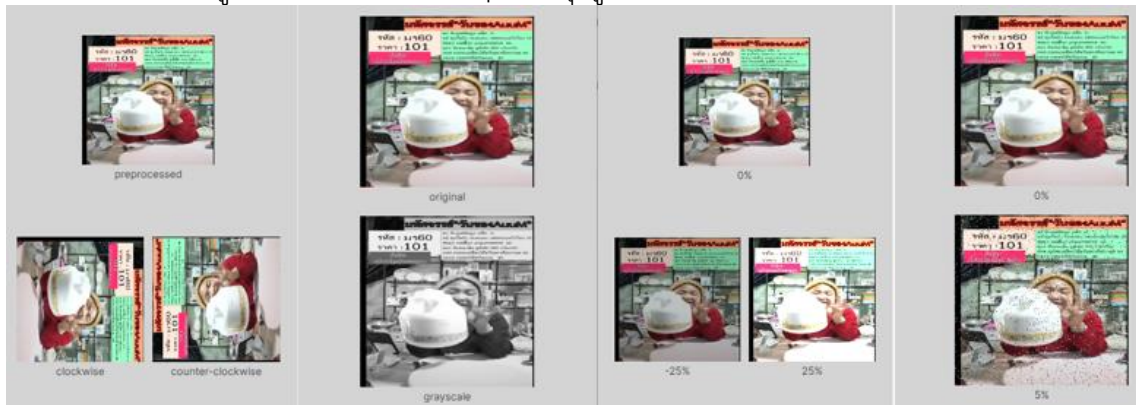
### ตารางที่ 3 จำนวนของข้อมูลในแต่ละส่วน

Training Set	Validation Set	Testing Set
62	17	13

โดยจำนวนของ Training Set จะแบ่งเป็น 62 รูปภาพ Validation Set 17 รูปภาพและ Testing Set มีจำนวน 13 รูปภาพ โดยจำนวนของ Training Set จะมีจำนวนที่มากกว่าเนื่องจาก ผลเฉลยที่สนใจมีรูปแบบที่คล้ายกัน

### 3.2.3 Data Augmentation

ในส่วนนี้จะนำข้อมูลไปทำการ Generate รูปภาพเพิ่มเพื่อเพิ่มจำนวนของข้อมูลที่จะทำไป train โดยจะทำการ หมุนภาพ 90 องศา,ทำให้เป็นภาพสีเทา,เพิ่มแสงสดแสง,เพิ่มสิ่งรบกวนในรูปภาพ และปรับขนาดของรูปภาพเป็น  $416 \times 416$  pixel ทุกรูปภาพ



ภาพประกอบที่ 3.7 การเพิ่มจำนวนรูปภาพใน Traing Set

จะทำการ Generate ทั้งหมด 3 ครั้ง โดยจะได้รูปภาพจากการ Generate ทั้งหมด 186 ภาพ จะมีรูปภาพที่ใช้ในการ train ทั้งหมด 558 โดยภาพที่ Generate ออกมาจะมีรูปแบบดัง ภาพประกอบที่ 3.8 ตัวอย่างรูปภาพที่ Generate ออกมา

ตารางที่ 4 จำนวนของข้อมูลก่อนและหลังทำการ Generate

	Training Set	Validation Set	Testing Set
ก่อน Generate	62	17	13
หลัง Generate	186	51	39



ภาพประกอบที่ 3.8 ตัวอย่างรูปภาพที่ Generate ออกมา

### 3.2.4 การสร้างโมเดล

ในส่วนของการสร้างโมเดลจะสร้างผ่าน google colab และใช้ YoloV5 ในการเทรนโมเดลโดยจะมีขั้นตอนดังต่อไปนี้

#### 1.การเตรียม Google Colab

**ตารางที่ 5** clone folder ของ yoloV5 และติดตั้งแพ็คเกจที่จำเป็น

```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
%pip install -qr requirements.txt # install
import torch
import utils

display = utils.notebook_init() # checks
```

ทำการ clone folder ของ yoloV5 และติดตั้งแพ็คเกจที่จำเป็นต่อการสร้างโมเดล

**ตารางที่ 6** การต่อ google colab เข้ากับ google drive

```
from google.colab import drive

drive.mount("/content/drive")
```

ทำการต่อ google colab เข้ากับ google drive เพื่อ ทำการนำข้อมูลที่เตรียมมาเข้ามาเทรนเพื่อสร้างโมเดล

### ตารางที่ 7 การกำหนด path ของไฟล์ข้อมูลที่เราเตรียมมาและทำการประกาศชื่อของ class

```
train: /content/drive/MyDrive/box_training
val: /content/drive/MyDrive/box_training

# Classes
names:
  0: box
  1: no
```

เป็นการกำหนด path ของไฟล์ข้อมูลที่เราเตรียมมาและทำการประกาศชื่อของ class ของกรอบข้อความเป็น 'box' และส่วนที่ไม่ใช่กรอบข้อความเป็น 'no' ในไฟล์ box.yaml ที่โคลนมา

### ตารางที่ 8 การกำหนด จำนวนของคลาส

```
# YOLOv5 by Ultralytics, GPL-3.0 license
# Parameters
nc: 2 # number of classes
```

เป็นการกำหนดจำนวนของคลาสจะใส่เป็น 2 โดยจะใช้ nc:2 คือคลาส 'box' และคลาส 'no' ในไฟล์ yolov5s.yaml

## 2.การเทรนข้อมูล

### ตารางที่ 9 การเทรนข้อมูลและพารามิเตอร์ที่ใช้ในการเทรน

```
!python train.py --img 416 --epochs 100 --data /content/yolov5/data/box.yaml
--weights yolov5s.pt
```

จากรูปภาพจะเป็นขั้นตอนในการเทรน โดยจะมีค่าพารามิเตอร์ดังนี้ ขนาดของรูปภาพที่ใช้เทรนจะเป็น 416 epochs เป็นจำนวนครั้งในการเทรนจะเทรนทั้งหมด 100 ครั้ง และ data จะเป็น path ของข้อมูลในการเทรน และ weights จะเป็นโมเดลของ yolo ที่เลือกใช้เทรนเป็น yolov5s

### ตารางที่ 10 ผลลัพธ์ในการเทรนโมเดล

```
100 epochs completed in 0.205 hours.
Optimizer stripped from runs/train/exp5/weights/last.pt, 14.3MB
Optimizer stripped from runs/train/exp5/weights/best.pt, 14.3MB
Validating runs/train/exp5/weights/best.pt...
Fusing layers...
```



Model summary: 157 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs							
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%
18/18 [00:06<00:00, 2.84it/s]							
	all	558	1152	0.999	0.999	0.995	0.911
	box	558	594	1	0.997	0.995	0.948
	False	558	558	0.997	1	0.995	0.874
Results saved to runs/train/exp5							

เมื่อทำการรันในส่วนของการเทรนข้อมูลเสร็จสิ้นผลลัพธ์จะได้ออกมาเป็น เป็นไฟล์ best.pt ซึ่งสามารถดาวโหลดมาใช้งานได้ตาม path ที่ระบุไว้ ในรูปภาพจะเก็บตัวโมเดลที่เทรนสำเร็จแล้วไว้ที่ 'runs/train/exp5'

#### ตารางที่ 11 การ detect ทดสอบโมเดล

```
!python detect.py --weights /content/yolov5/runs/train/exp5/weights/best.pt --img 416 --conf 0.25 --source data/images/kan10.png display.Image(filename='runs/detect/exp13/kan10.png', width=400)
```

ในส่วนของการทดสอบโมเดลจะทำการใช้คำสั่งในการ detect ส่วนที่สนใจในรูปภาพโดยจะทำการรันไฟล์ detect.py โดยมีพารามิเตอร์ดังนี้

- weights คือ path ของโมเดลที่เทรนได้
- img คือ ขนาดของรูปภาพ
- confi คือ การกำหนดค่าความเชื่อมั่นถ้าต่ำกว่าที่กำหนดจะถูกละเว้น
- source คือ path ของไฟล์รูปภาพ
- ผลลัพธ์ในการ detect ของแต่ละร้านหลังจากการใช้คำสั่ง detect



ภาพประกอบที่ 3.9 ผลลัพธ์ในการ detect ของรูปร้านค้าแต่ละร้าน



### 3.3 ในส่วนของ desktop application

เป็นส่วนที่รับข้อมูลไลฟ์สดของทางร้านค้ามาทำการประมวลผลโดยการ ตรวจจับข้อความจาก โมเดลที่เทรนมาได้และทำการนำข้อมูลที่ตรวจจับมาได้มาแปลงจากรูปภาพเป็นตัวอักษรเพื่อส่งไปเก็บใน firebase โดยจะมีขั้นตอนดังนี้

#### ตารางที่ 12 การ import ทรัพยากรที่จำเป็น

```
# importing the required packages
from importlib.resources import path
from statistics import mode
import torch
from PIL import Image, ImageTk
import pyautogui as pg
import cv2
import numpy as np
import selectinwindow
import sys
from time import time
sys.setrecursionlimit(10 ** 9)
```

ในส่วนของ class detection จะมี function ดังนี้

#### ตารางที่ 13 constructor

```
def __init__(self, capture_index, model_name):
    self.capture_index = capture_index
    self.model = self.load_model(model_name)
    self.classes = self.model.names
    self.device = "cuda" if torch.cuda.is_available() else "cpu"
```

ใน constructor จะมี พารามิเตอร์เป็น capture\_index,model\_name โดย capture\_indexจะเป็น ตำแหน่งแรกของ index ที่ทำการจับภาพหน้าจอมา ในส่วนของบรรทัดที่ 27-30จะเป็นการใส่ข้อมูลให้กับตัวแปร capture\_index,model,classes,device

**ตารางที่ 14** Function load\_model

```
def load_model(self,model_name):

    model = torch.hub.load('ultralytics/yolov5', 'custom', path=model_name,
    force_reload=False)
    return model
```

Function load\_model จะมี พารามิเตอร์ เป็น model\_name คือ path ของไฟล์โมเดล โดยจะให้ โหลดโมเดลมาไว้ในตัวแปรโมเดลผ่าน torch.hub.load() โดยจะมีพารามิเตอร์ดังนี้ torch.hub.load(,ประเภทของการ,path ของไฟล์โมเดล,เป็นการระบุว่าจะให้โมเดลที่ถูกโหลดไปก่อนหน้านี้โหลดซ้ำจาก path ที่ระบุ) และจะทำการ return model ออกไป

**ตารางที่ 15** Function score\_frame

```
def score_frame(self, frame):

    self.model.to(self.device)
    frame = [frame]
    results = self.model(frame)

    labels, cord = results.xyxy[0][:, -1], results.xyxy[0][:, :-1]
    return labels, cord
```

Function score\_frame จะเป็นฟังก์ชันที่ใช้ในการใส่คะแนนให้แก่ผลลัพธ์ในการตรวจจับแต่ละครั้ง

**ตารางที่ 16** Function plot\_box

```
def plot_boxes(self, results, frame):

    labels, cord = results
    n = len(labels)
    x_shape, y_shape = frame.shape[1], frame.shape[0]
    for i in range(n):
        row = cord[i]
        if row[4] >= 0.2:
            x1, y1, x2, y2 = (
                int(row[0] * x_shape),
                int(row[1] * y_shape),
                int(row[2] * x_shape),
                int(row[3] * y_shape),
            )
```

```

        print(x1, y1, x2, y2)
        bgr = (0, 255, 0)
        cv2.rectangle(frame, (x1, y1), (x2, y2), bgr, 2)

    return frame

```

Function `plot_box` จะเป็นการสร้างกรอบสี่เหลี่ยมล้อมรอบผลลัพธ์ที่ส่งเข้ามาในพารามิเตอร์

#### ตารางที่ 17 function `__call__` (1)

```

def __call__(self):
    i=0
    # Initialize the drag object#
    wName = "Live"
    resolution = (1920, 1080)
    # Create an Empty window
    cv2.namedWindow("Live", cv2.WINDOW_NORMAL)
    # Resize this window
    #cv2.resizeWindow("Live", 2000, 800)
    img = pg.screenshot()
    # Convert the screenshot to a numpy array
    frame = np.array(img)
    # Convert it from BGR(Blue, Green, Red) to
    # RGB(Red, Green, Blue)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    rectl = selectinwindow.DragRectangle(frame, wName, resolution[0],
    resolution[1])
    cv2.namedWindow(rectl.wname)
    cv2.setMouseCallback(rectl.wname, selectinwindow.dragrect, rectl)

```

ในส่วนของการเรียกใช้งานตัวคลาสนี้จะทำการสร้างหน้าต่างที่ครอบคลุมทั้งหน้าจอและทำการ capture หน้าจอและทำการแปลงจากรูปภาพไปเก็บ array และทำการเรียกใช้ function ในการ capture

#### ตารางที่ 18 function `__call__` (2)

```

while True:
    # display the image
    cv2.imshow(wName, rectl.image)

```

```

key = cv2.waitKey(1) & 0xFF
# if return flag is True, break from the loop
if rectl.returnflag:
    break

```

ทำการแสดงผลหน้าจอที่ capture มาได้เมื่อทำการลากกรอบส่วนที่ต้องการและกดคลิกที่พื้นที่ที่ลากกรอบจะทำการหยุดการทำงานรูป

#### ตารางที่ 19 function \_\_call\_\_ (3)

```

cv2.destroyAllWindows()
cv2.namedWindow("LiveCap")
cv2.resizeWindow("LiveCap", rectl.outRect.w, rectl.outRect.h)
results = []
while True:
    img =
pg.screenshot(region=(rectl.outRect.x,rectl.outRect.y,rectl.outRect.w,rectl.outRect.h));
    frame = np.array(img)
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    if(i%30 == 0):
        results = self.score_frame(frame)

    frame = self.plot_boxes(results, frame)

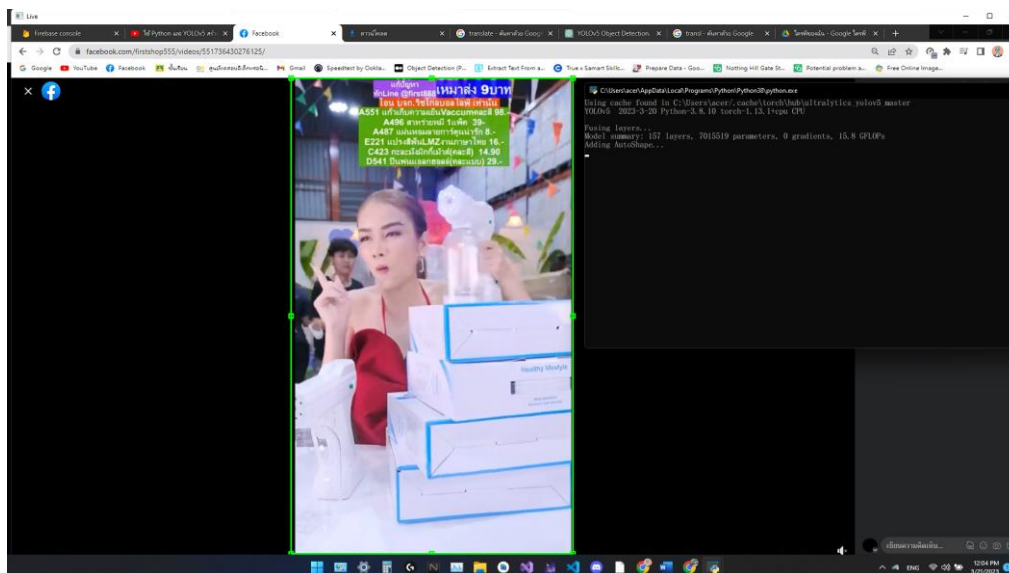
    #out.write(frame)
    cv2.imshow('LiveCap', frame)
    i=i+1
    if cv2.waitKey(1) == ord('q'):
        break

detector = detection(capture_index=0,model_name='D:/seneir/bestNew.pt')
detector()

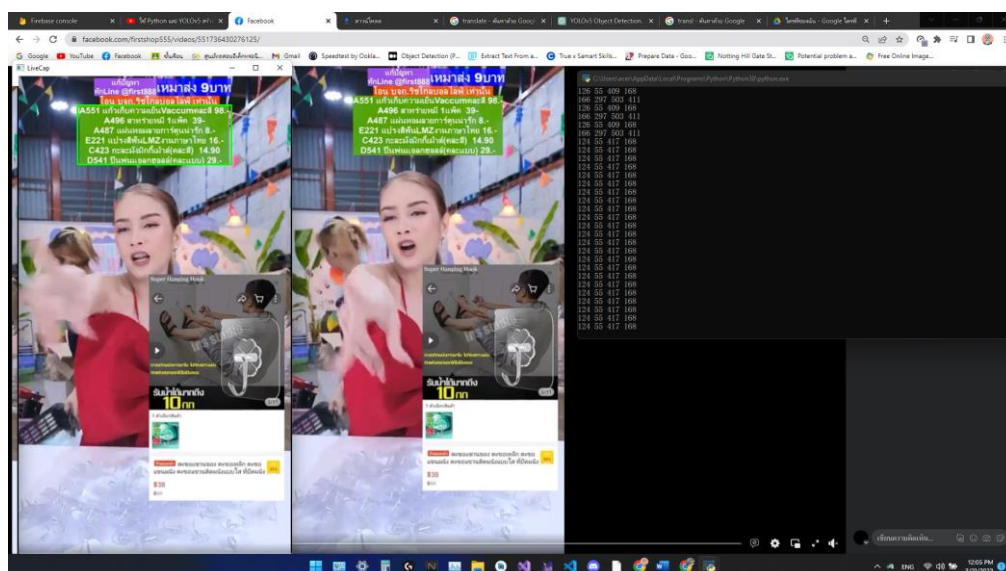
cv2.destroyAllWindows()

```

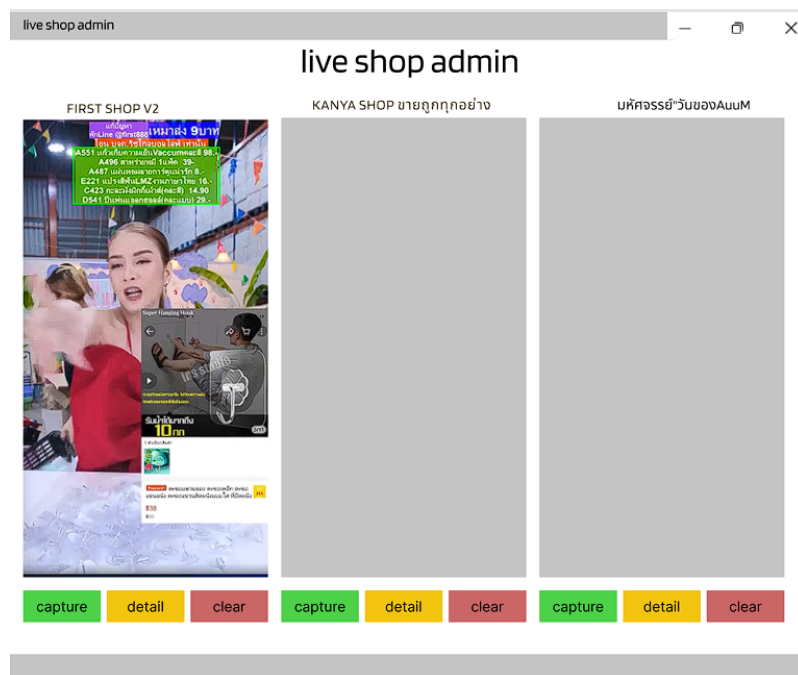
เมื่อออกจากลูปจะทำการปิดหน้าต่างทั้งหมดและทำการปรับขนาดหน้าจอตามทีลากกรอบเฉพาะส่วนและจะเข้าสู่การทำงานวนซ้ำโดยในลูปจะทำการ capture รูปภาพตามขนาดที่กำหนดและทำการแปลงรูปภาพเป็น array และเพื่อความถี่ของการแสดงผลจึงทำการ detect ทุกๆ 30 frame เพราะเนื่องจากรันค่าไม่ได้มีการเปลี่ยนแปลงในส่วนของคุณค่าข้อมูลตลอด และจะทำการแสดงผล frame นั้นๆออกมาทุก frame และหยุดการทำงานของโปรแกรมโดยการ กดตัว 'q'



ภาพประกอบที่ 3.10 การทำงานในส่วนของการเลือกส่วนที่ต้องการ



ภาพประกอบที่ 3.11 ผลลัพธ์ในการ detect และการเลือกพื้นที่ที่ต้องการ



ภาพประกอบที่ 3.12 ผลลัพธ์ในการ detect ใน desktop application

3.การเชื่อมต่อ python กับ firebase

ตารางที่ 20 การเชื่อมต่อ firebase กับ python

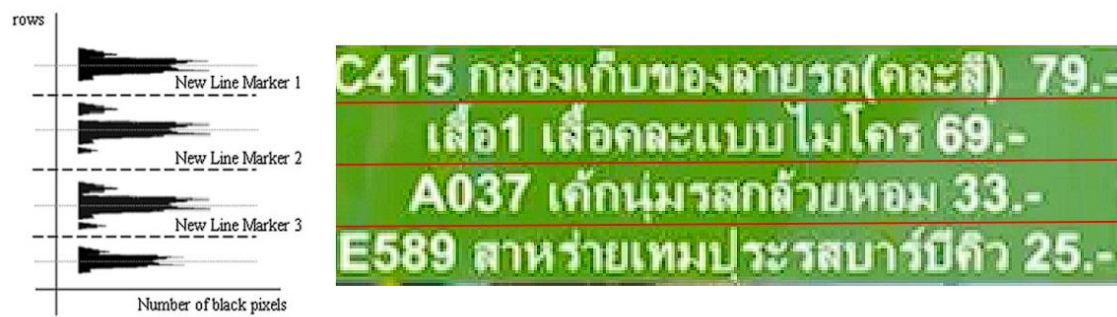
```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

# Fetch the service account key JSON file contents
cred = credentials.Certificate('secret key.json')
# Initialize the app with a service account, granting admin privileges
firebase_admin.initialize_app(cred, {
    'databaseURL': "URL to database"
})

ref = db.reference('Database reference')
print(ref.get())
```

### 3.4 การตัดบรรทัดด้วยเทคนิค projection profile

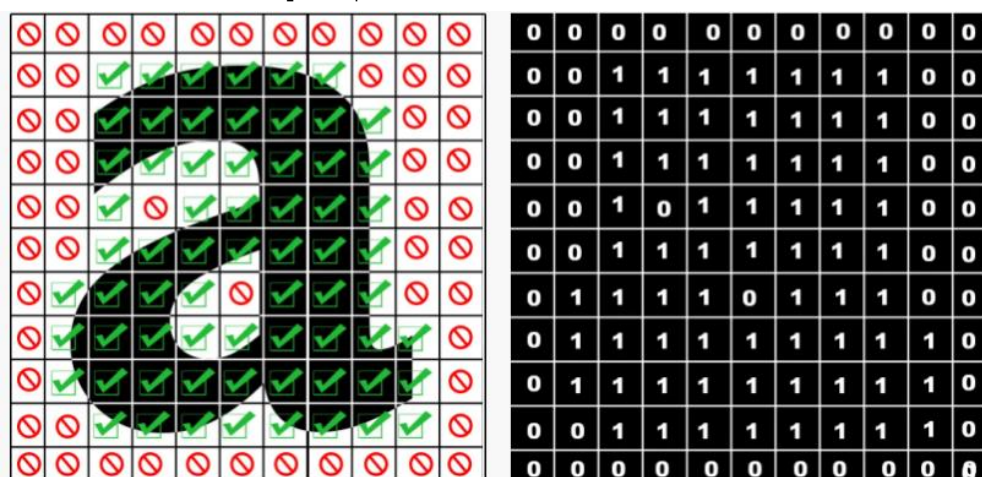
ทำการแยกบรรทัดของข้อมูลเพื่อทำการแยกเส้นค่าเป็นทีละชั้นโดยใช้เทคนิค Horizontal projection profile แยกบรรทัดของเอกสาร และ สร้าง Histogram ตามแนวนอนเพื่อหาจุดที่มีความหนาแน่นของ pixel ต่ำโดยจะ mark จุดที่มีความหนาแน่นของ pixel ต่ำเป็นเส้นคั่นบรรทัดและทำการตัดเป็นรูปภาพส่งไปยังขั้นตอนถัดไป



ภาพประกอบที่ 3.13 เทคนิคการทำ projection profile

### 3.5 อ่านข้อความ OCR

การรู้จำอักขระด้วยแสงทำงานโดยแบ่งรูปภาพของอักขระข้อความออกเป็นส่วนๆ และแยกความแตกต่างระหว่างพื้นที่ว่างและไม่ว่าง ขึ้นอยู่กับฟอนต์หรือสคริปต์ที่ใช้สำหรับตัวอักษร ผลรวมตรวจสอบของเมทริกซ์ผลลัพธ์จะถูกระบุในภายหลัง ตามอักขระในภาพ



ภาพประกอบที่ 3.14 ผลรวมตรวจสอบของเมทริกซ์

โดยขั้นตอนการทำงานในส่วนของโค้ดจะมีดังนี้

#### ตารางที่ 21 การแปลงรูปเป็นตัวอักษรด้วย tesserract

```
import pytesseract as tess
from PIL import Image
tess.pytesseract.tesseract_cmd = r'D:\ocr\tesseract.exe'
image = Image.open('D:\ocr\kan2.png')
text = tess.image_to_string(image, lang='tha+eng')
print(text)
```

บรรทัดที่ 1 import flies tesserract

บรรทัดที่ 2 import image เพื่อใช้ในการเพิ่มรูปภาพเข้ามา

บรรทัดที่ 3 คือการอ่าน path files ของ tesserract

บรรทัดที่ 4 การอ่าน files รูปภาพแล้วกับไว้ในตัวแปล image เพื่อนำไปประมวลผล

บรรทัดที่ 5 การเอารูปภาพมาแปลงเป็น text จะแปลงภาษาไทยและภาษาอังกฤษ

type: image

type: String

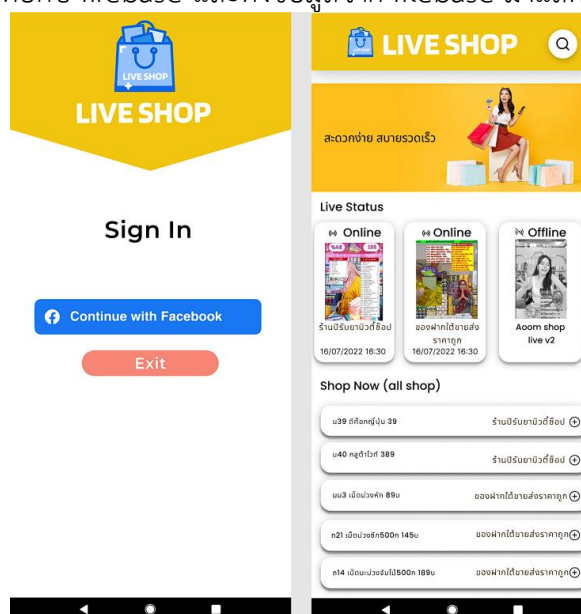
 → A037 เด็กนุ่มรสกล้วยหอม 33.-

ภาพประกอบที่ 3.15 การแปลงรูปเป็นตัวอักษร



### 3.6 Mobile application

ในส่วนของ mobile application จะใช้ Flutter ในการสร้าง application ที่ทำการล็อกอินด้วย facebook ได้และเชื่อมต่อกับ firebase และดึงข้อมูลจาก firebase มาแสดงผลได้

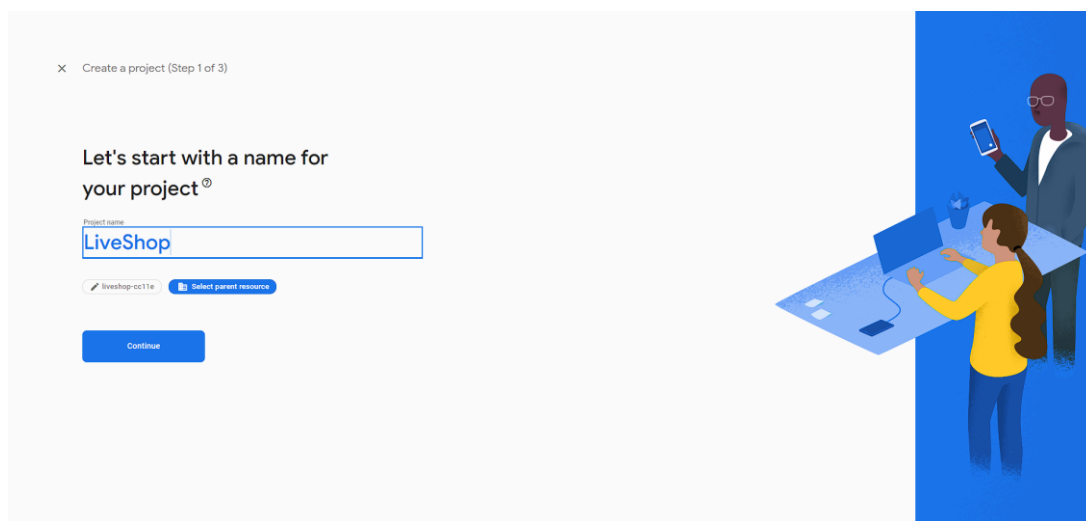


ภาพประกอบที่ 3.16 ตัวอย่างหน้าตา UI mobile application

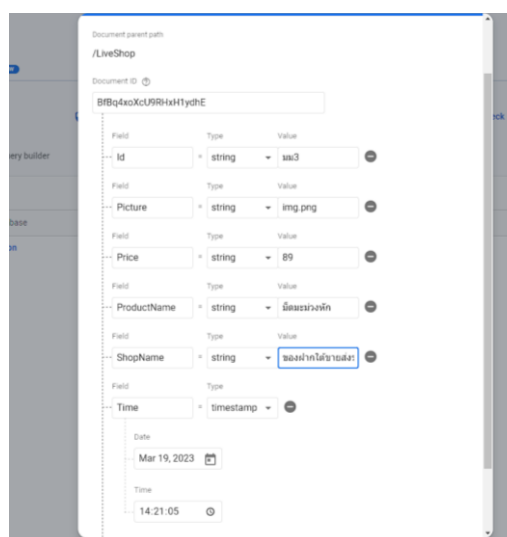
### 3.7 การจัดเก็บข้อมูล

ในส่วนของการจัดการข้อมูลจะใช้ Cloud Firestore จัดเก็บข้อมูล โดยใช้ฐานข้อมูล NoSQL ที่โฮสต์บนคลาวด์ Cloud Firestore โดยโครงสร้างจะมี 3 ส่วนคือ

1. Collection เป็น Folder ที่ไว้เก็บเอกสาร และมีชื่อบอกว่าเก็บเอกสารเกี่ยวกับอะไร
2. Document เป็นกระดาษไว้สำหรับเก็บข้อมูล และมีชื่อบอกว่าเก็บข้อมูลเกี่ยวกับอะไร
3. Data เป็นที่เก็บข้อมูล



ภาพประกอบที่ 3.17 การสร้างโปรเจ็ค firebase



ภาพประกอบที่ 3.18 การสร้าง Document ใน firebase

โดยข้อมูลที่ทำกรส่งไปเก็บในฐานข้อมูลจะประกอบไปด้วย

- ชื่อร้านค้า
- รหัสสินค้า
- ชื่อสินค้า
- ราคาสินค้า
- รูปภาพในการไลฟ์สด
- เวลาที่ capture รูปภาพ

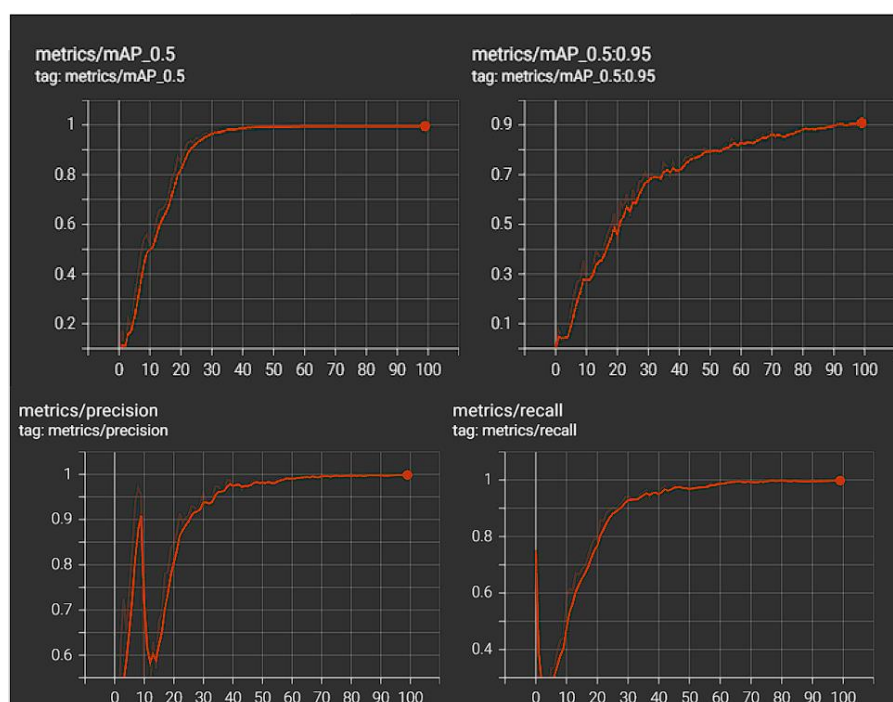
LiveShop	BfBq4xoXcU9RHxH1ydhE
+ Add document	+ Start collection
BfBq4xoXcU9RHxH1ydhE >	+ Add field
	Id: "mm3"
	Picture: "img.png"
	Price: "89"
	ProductName: "มีดมะม่วงหิ"
	ShopName: "ของฝากได้ขายส่งราคาถู"
	Time: March 19, 2023 at 2:21:05PM UTC+7

ภาพประกอบที่ 3.19 ตัวอย่างข้อมูลและประเภทของข้อมูล

### 3.8 วัดประสิทธิภาพ

#### 1. การวัดประสิทธิภาพการ Detect วัดด้วย Mean Average Precision (mAP)

ค่าเฉลี่ยความแม่นยำเฉลี่ย Mean Average Preciso (mAP) เป็นเมตริกที่ใช้ในการประเมินแบบจำลองการตรวจจับวัตถุ เช่น Fast R-CNN, YOLO, Mask R-CNN เป็นต้น ค่าเฉลี่ยของค่าความแม่นยำเฉลี่ย (AP) จะคำนวณจากค่าการเรียกคืนตั้งแต่ 0 ถึง 1 mAP คำนวณได้จากค่าดังนี้



ภาพประกอบที่ 3.20 กราฟการเปลี่ยนแปลงค่า mAP ของ model

## 2.การวัดประสิทธิภาพการ OCR วัดด้วย CER( Character Error Rate)

ค่า CER จะวัดเป็นเปอร์เซ็นต์ สังเกตว่ายิ่งค่า CER เยอะ ประสิทธิภาพของโมเดลก็จะยิ่งแย่มาก ความเหมาะสมของค่า CER ที่เหมาะสมสำหรับงานที่กำหนดไว้คือค่า CER ต้อง น้อยกว่า 10% โดยค่า CER สามารถหาค่าได้ดังนี้

$$CER = \frac{S + D + I}{N}$$

### ภาพประกอบที่ 3.21 สมการ CER

I (inserted words) คือ จำนวนตัวอักษรที่ถูกแทรกขึ้นมาจากข้อความเดิม  
D (deleted words) คือ จำนวนตัวอักษรที่หายไปจากข้อความเดิม  
S (substituted words) คือ จำนวนตัวอักษรที่ถูกแทนที่ไปจากคำเดิม  
N คือ จำนวนตัวอักษรทั้งหมด

A037 เด็กนุบรสกล้วยหอม 33.-	original words
A037 เด็กนุบรสกล้วยหอม 33.-	deleted words
A037 เด็กนุบรสกล้วยหอม 33.-	inserted words
A037 เด็กนุบรสกล้วยหอ 33.-	substituted words

### ภาพประกอบที่ 3.22 ตัวอย่างตัวแปรในสมการ CER

original words: A037 เด็กนุบรสกล้วยหอม 33.-  
OCR result: A037 เด็กนุบรสกล้วยหอ 33.-  
CER = (1+1+0)/28

### ภาพประกอบที่ 3.23 ตัวอย่างการหาค่า CER

จาก ภาพประกอบที่ 3.23 ตัวอย่างการหาค่า CER เมื่อนำค่าจากผลลัพธ์จากการ OCR มา แทนค่าในตัวแปรทั้ง 4 จะมีผลลัพธ์ดังนี้

S (substituted words) จะมีค่าเท่ากับ 1

D (deleted words) จะมีค่าเท่ากับ 1

I (inserted words) จะมีค่าเท่ากับ 0

N จะมีค่าเท่ากับ 28

เมื่อแทนค่าทั้งหมดลงในสมการ  $CER = (S, D, I) / N$  จะได้  $(1+1+0) / 28 = 0.07$

เมื่อนำมาคิดเป็นเปอร์เซ็นต์จะได้ค่า CER = 7%

## อ้างอิง

- [1] Hizoka, "Flutter," [Online]. Available: <https://shorturl.asia/zPKJZ>. [Accessed 15 07 2022].
- [2] J. Saengow, "Firebase," [Online]. Available: <https://shorturl.asia/g6H0J>. [Accessed 07 16 2022].
- [3] N. Jesadapatrakul, "Image Processing," [Online]. Available: <https://shorturl.asia/4LirK>. [Accessed 10 06 2022].
- [4] NUMKINGSTON, "Optical Character Recognition," [Online]. Available: <https://shorturl.asia/nmq7h>. [Accessed 12 07 2022].
- [5] S. KANOKTIPSATHARPORN. [Online]. Available: <https://shorturl.asia/twFLQ>. [Accessed 14 07 2022].
- [6] S. D. R. G. a. A. F. J. Redmon, "ieeexplore," [Online]. Available: <https://ieeexplore.ieee.org/document/7780460>. [Accessed 14 07 2022].
- [7] H. Z. a. Z. N. F. Zhou, "Safety Helmet Detection Based on YOLOv5," [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9362711>. [Accessed 07 15 2022].
- [8] I. Kunakornum, "medium," [Online]. Available: <http://surl.li/ctfuc>. [Accessed 15 07 2022].
- [9] E. BARRETT, "fortune," [Online]. Available: <https://fortune.com/2018/10/28/in-china-facial-recognition-tech-is-watching-you/>. [Accessed 14 07 2022].