

แอปพลิเคชันรวมสินค้าไลฟ์สดด้วยการประมวลผลทางภาพ
application live shop with image processing

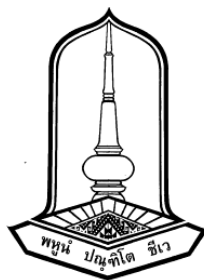
โครงการปริญญานิพนธ์
ของ
นายกฤตเมธ บัวสิงห์
นายกิตติศักดิ์ มนพรหมมา

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
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 มนพรมมา แล้วเห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบโครงการปริญญานิพนธ์

..... ประธานสอบ
(.....)

..... กรรมการ
(.....)

.....  ที่ปรึกษาโครงการปริญญานิพนธ์หลัก
(ดร.พัฒนพงษ์ ชมพูวิเศษ)

หลักสูตรวิทยาการคอมพิวเตอร์อนุมัติให้รับโครงการปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

.....
(อาจารย์พระ พฤกษ์ศรี)

.....
(ผู้ช่วยศาสตราจารย์พิมลรัตน์ อ้วนศรีเมือง)

อาจารย์ผู้ประสานงานวิชาโครงการปริญญานิพนธ์

วันที่ เดือน พ.ศ.

สารบัญ

หน้า

สารบัญ.....	ก
สารบัญตาราง.....	1
สารบัญรูปภาพ.....	2
สารบัญรูปภาพ(ต่อ).....	3
สารบัญรูปภาพ(ต่อ).....	4
บทที่ 1 บทนำ.....	5
1.1 หลักการและเหตุผล.....	5
1.2 วัตถุประสงค์ของโครงการ.....	5
1.3 ขอบเขตของโครงการ.....	5
1.3.1 ผู้ใช้งาน (ใช้งานผ่าน Mobile Application).....	6
1.3.2 ผู้ดูแลระบบ (ใช้งานผ่าน Desktop Application ที่พัฒนาด้วย Python).....	8
1.3.3 ระบบ (ระบบพัฒนาเป็น Module ด้วย Python).....	8
1.4 ภาพรวมของระบบ.....	10
จากภาพรวมระบบจะการทำงานดังนี้.....	10
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	11
1.5.1 ช่วยให้ผู้ใช้งานที่เป็นผู้ค้าขายมีช่องทางการขายสินค้ามากขึ้น.....	11
1.5.2 ช่วยให้ผู้ใช้งานที่เป็นลูกค้ามีความสะดวกมากขึ้น.....	11
1.6 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน.....	11
1.6.1 ฮาร์ดแวร์.....	11
1.6.2 ซอฟต์แวร์.....	12
1.7 แผนการดำเนินงาน.....	12
1.8 ตัวอย่างโปรแกรม.....	13
1.8.1 แอปพลิเคชันบนอุปกรณ์ไร้สายแบบเคลื่อนที่.....	13
1.8.2 แอปพลิเคชันบนคอมพิวเตอร์สำหรับผู้ดูแลระบบ.....	18
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	21

สารบัญ (ต่อ)

	หน้า
2.1 ทฤษฎีที่เกี่ยวข้อง.....	21
2.1.1 Flutter	21
2.1.2 Firebase	22
2.1.3 การประมวลผลภาพ (Image Processing)	24
2.1.4 OCR หรือ Optical Character Recognition.....	26
2.1.1 Object Detection.....	27
2.1.1 YOLO (You Only Look Once).....	27
2.2 ระบบงานที่เกี่ยวข้อง	27
2.2.1 Safety Helmet Detection Based on YOLOv5	27
บทที่ 3.....	31
3.1 เก็บรวบรวมข้อมูล	32
3.2 การเตรียมข้อมูล	33
3.2.1 การเตรียมข้อมูลในการ training	33
3.2.2 การวาดภาพผลเฉลย.....	33
3.2.3 การทำ Data Augmentation.....	35
3.3 การสร้างโมเดลโดยใช้ yoloV5.....	37
3.3.1 Convolution Layer	37
3.3.2 Rectified Linear Unit (ReLU).....	38
3.3.3 ReLU leaky.....	38
3.3.4 Pooling layer	39
3.3.5 Fully Connected Layer	39
3.3.6 การแทนค่าความสูงและความกว้างของ Bounding Box.....	43
3.3.7 การทำ anchor box	44
3.3.8 Intersection over Union	45
3.3.9 Non-Max Suppression	47
3.4 การตัดบรรทัดด้วยเทคนิค projection profile.....	49
3.4.1 การเตรียมภาพ	49
3.4.2 แปลงภาพสี RGB ไปเป็นภาพระดับเทา.....	50
3.4.3 การแปลงภาพระดับเทา ไปเป็นภาพขาวดำ (Binary image)	51
3.4.4 การวาดเส้นในจุดที่เป็นที่ว่าง	53

สารบัญ (ต่อ)

	หน้า
3.5 ขั้นตอนการเรียกใช้ Tesseract OCR.....	54
3.6 Mobile application.....	55
3.7 การจัดเก็บข้อมูล	56
3.8 วัดประสิทธิภาพ	58
อ้างอิง.....	63

สารบัญตาราง

	หน้า
ตารางที่ 1 แผนการดำเนินงาน	12
ตารางที่ 2 ตารางจำนวนและเวลาเฉลี่ยของวิดีโอในแต่ละร้านค้า	32
ตารางที่ 3 จำนวนของข้อมูลในแต่ละส่วน	34
ตารางที่ 4 จำนวนของข้อมูลก่อนและหลังทำการ Generate	35
ตารางที่ 5 ค่า parameter ใน ค่าผลเฉลย y ทั้ง 7	40
ตารางที่ 6 การแทนค่า b_x , b_y , b_h , b_w	43
ตารางที่ 7 parameter ของการทำ anchor box.....	44
ตารางที่ 8 การแปลงรูปเป็นตัวอักษรด้วย tesserract	55

สารบัญรูปภาพ

	หน้า
ภาพประกอบที่ 1.1 ตัวอย่างข้อมูลที่นำมาแยก.....	9
ภาพประกอบที่ 1.2 ภาพรวมของระบบ.....	10
ภาพประกอบที่ 1.3 หน้าแรกของโปรแกรม.....	13
ภาพประกอบที่ 1.4 หน้าล็อกอินเข้าสู่ระบบ.....	14
ภาพประกอบที่ 1.5 หน้าหลักของแอปพลิเคชัน.....	15
ภาพประกอบที่ 1.6 หน้าค้นหาสินค้า.....	16
ภาพประกอบที่ 1.7 หน้ารายละเอียดสินค้า.....	17
ภาพประกอบที่ 1.8 หน้าแรกโปรแกรม.....	18
ภาพประกอบที่ 1.9 หน้าเลือกพื้นที่ภาพ.....	19
ภาพประกอบที่ 1.10 หน้าผลลัพธ์ในการถ่ายภาพหน้าจอ.....	19
ภาพประกอบที่ 1.11 หน้ารายละเอียดข้อมูล.....	20
ภาพประกอบที่ 2.1 ตัวอย่างโครงสร้าง Flutter.....	21
ภาพประกอบที่ 2.2 การตรวจจับและประมวลผลใบหน้าจากภาพ [4].....	25
ภาพประกอบที่ 2.3 ขบวนการการทำงานของ Optical Character Recognition.....	26
ภาพประกอบที่ 2.4 ภาพการตรวจจับหมวดนิรภัยของคนงาน [8].....	28
ภาพประกอบที่ 2.5 การแบ่งช่องของรูปภาพออกเป็น grid [9].....	28
ภาพประกอบที่ 2.6 ภาพของเทคนิค Anchor Box [9].....	29
ภาพประกอบที่ 2.7 การทำงานของขบวนการ IOU [9].....	29
ภาพประกอบที่ 3.1 ขั้นตอนการดำเนินงานของระบบ.....	31
ภาพประกอบที่ 3.2 Streamlabs Desktop.....	32
ภาพประกอบที่ 3.3 ตัวอย่างรูปภาพที่ทำการเก็บ.....	33
ภาพประกอบที่ 3.4 การวาดภาพผลเฉลยรูปภาพใน Roboflow.....	33
ภาพประกอบที่ 3.5 การวาดภาพผลเฉลยรูปภาพใน Roboflow (2).....	34
ภาพประกอบที่ 3.6 ภาพผลเฉลยที่ถูกตัดจากทั้ง 3 ร้านค้า.....	34
ภาพประกอบที่ 3.7 การเพิ่มจำนวนรูปภาพใน Traing Set.....	35
ภาพประกอบที่ 3.8 ตัวอย่างรูปภาพที่ Generate ออกมา.....	36
ภาพประกอบที่ 3.9 ผลเฉลยของรูปภาพที่ Generate ออกมา.....	36
ภาพประกอบที่ 3.10 การทำ Convolution.....	37
ภาพประกอบที่ 3.11 การคำนวณRuLE.....	38
ภาพประกอบที่ 3.12 การคำนวณRuLE leaky.....	38
ภาพประกอบที่ 3.13 Pooling layer.....	39
ภาพประกอบที่ 3.14 Fully Connected Layer.....	39

สารบัญรูปภาพ(ต่อ)

	หน้า
ภาพประกอบที่ 3.15 รูปภาพ input ที่ใส่ grid	40
ภาพประกอบที่ 3.16 ค่าผลเฉลี่ย y ในแต่ละ grid cell	41
ภาพประกอบที่ 3.17 ตัวอย่าง grid ที่มีวัตถุในภาพและการแทนค่า	42
ภาพประกอบที่ 3.18 ตัวอย่าง grid ที่ไม่มีวัตถุในภาพและการแทนค่า	42
ภาพประกอบที่ 3.19 จุดกึ่งกลางของ Bounding Box	43
ภาพประกอบที่ 3.20 การแทนค่าใน anchor box และภาพรวมของ anchor box	45
ภาพประกอบที่ 3.21 IOU	45
ภาพประกอบที่ 3.22 ตัวอย่างการคำนวณค่า IoU	46
ภาพประกอบที่ 3.23 ตัวอย่างของ ค่า IoU ในรูปภาพ	46
ภาพประกอบที่ 3.24 ตัวอย่างการทำ Non-Max Suppression	47
ภาพประกอบที่ 3.25 ตัวอย่างการทำ Non-Max Suppression	47
ภาพประกอบที่ 3.26 การทำงานในส่วนของการเลือกส่วนที่ต้องการ	48
ภาพประกอบที่ 3.27 ผลลัพธ์ในการ detect และการเลือกพื้นที่ที่ต้องการ	48
ภาพประกอบที่ 3.28 ผลลัพธ์ในการ detect ใน desktop application	49
ภาพประกอบที่ 3.29 ตัวอย่างภาพนำเข้า	49
ภาพประกอบที่ 3.30 ตารางค่าสี R G B ตามลำดับ	50
ภาพประกอบที่ 3.31 การแปลงภาพสี RGB ไปเป็นระดับเทา	50
ภาพประกอบที่ 3.32 ตัวอย่างผลลัพธ์หลังจากการแปลงเป็นภาพระดับเทา	50
ภาพประกอบที่ 3.33 ตัวอย่างหลังจากการแปลงเป็นภาพระดับเทา	51
ภาพประกอบที่ 3.34 ตัวอย่างการแปลงภาพระดับ เทา เป็นภาพ Binary	53
ภาพประกอบที่ 3.35 ตัวอย่างสแกนรูปภาพจากซ้ายไปขวา	53
ภาพประกอบที่ 3.36 ตัวอย่างการมาร์คเส้น	54
ภาพประกอบที่ 3.37 ผลลัพธ์ในการวางเส้นแยกบรรทัด	54
ภาพประกอบที่ 3.38 ตัวอย่างของผลลัพธ์ของการแปลงที่ผิดพลาด	55
ภาพประกอบที่ 3.39 ตัวอย่างหน้าต่าง UI mobile application	56
ภาพประกอบที่ 3.40 การสร้างโปรเจ็ค firebase	56
ภาพประกอบที่ 3.41 การสร้าง Document ใน firebase	57
ภาพประกอบที่ 3.42 ตัวอย่างข้อมูลและประเภทของข้อมูล	57
ภาพประกอบที่ 3.43 Confusion Matrix	58
ภาพประกอบที่ 3.44 สมการหาค่า IoU	58
ภาพประกอบที่ 3.45 ตัวอย่างการหาค่า IoU	59
ภาพประกอบที่ 3.46 Predicted box และ Ground truth box	59
ภาพประกอบที่ 3.47 การหาค่า Precision	60

สารบัญรูปภาพ(ต่อ)

	หน้า
ภาพประกอบที่ 3.48 ตัวอย่างผลลัพธ์ในการทำนาย	60
ภาพประกอบที่ 3.49 การหาค่า recall.....	60
ภาพประกอบที่ 3.50 กราฟการเปลี่ยนแปลงค่า mAP precision และ recall ของ model.....	61
ภาพประกอบที่ 3.51 สมการ CER	62
ภาพประกอบที่ 3.52 ตัวอย่างตัวแปรในสมการ CER.....	62
ภาพประกอบที่ 3.53 ตัวอย่างการหาค่า CER.....	62

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

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

ปัจจุบันการที่จะขายพัสดุออนไลน์ในการไลฟ์สดนั้น 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
 - วันที่เวลาที่ capture รูปภาพ

ตัวอย่าง เช่น ปรีนย่า บิวตี้ ซุป คือชื่อร้าน ,น39 คือรหัสสินค้า, ดิท็อกญี่ปุ่น คือชื่อสินค้า ,39 คือราคาสินค้า,16/07/2022 16:30 คือ วันที่เวลาที่ capture รูปภาพ
4. ผู้ใช้งานสามารถเลือกที่สินค้าเพื่อดูรายละเอียดเพิ่มเติมได้โดยรายการที่ปรากฏในไลฟ์สดประกอบด้วย
 - ชื่อร้านค้า
 - รหัสสินค้า
 - ชื่อสินค้า
 - ราคาสินค้า
 - รูปภาพในการไลฟ์สด
 - วันที่เวลาที่ capture รูปภาพ

ตัวอย่าง เช่น FIRST SHOP V2คือชื่อร้าน ,๐7 คือรหัสสินค้า, สดรับ ขมิ้น คือชื่อสินค้า ,59 คือราคาสินค้า,14/06/2022 12:30 คือ วันที่เวลาที่ capture รูปภาพ
5. เมื่อผู้ใช้เลือกสินค้าผู้ใช้งานสามารถกดปุ่ม “chat”ในหน้ารายละเอียดสินค้าเพื่อลิงไปยังแชท ของร้านค้าได้
6. เมื่อผู้ใช้เลือกสินค้าผู้ใช้งานสามารถกดปุ่ม “buy”ในหน้ารายละเอียดสินค้าเพื่อลิงไปยังแชท 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

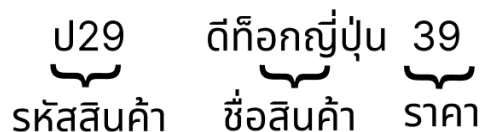
ตัวอย่าง เช่น ๐61 คือรหัสสินค้า, ลิป KO(เบอร์ 8) คือชื่อสินค้า, 59 คือราคาสินค้า

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, ดีทีเอกญ่ปุ่นคือชื่อสินค้า,39 คือราคาสินค้า



ภาพประกอบที่ 1.1 ตัวอย่างข้อมูลที่นำมาแยก

7. ระบบสามารถนำข้อมูลไปจัดเก็บในฐานข้อมูลได้โดยรายการที่ปรากฏในไลฟ์

สดประกอบด้วย

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

ตัวอย่าง เช่น ของฝากใต้ชายส่งราคาถูก คือชื่อร้าน,มม3 คือรหัสสินค้า,เมืมมะม่วงหัท คือชื่อสินค้า,89บ คือราคาสินค้า,14/06/2022 14:30 คือ วันที่เวลาที่ capture รูปภาพ

8. ระบบจะทำงานแบบ real-time เป็น Database ของ Firebase

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

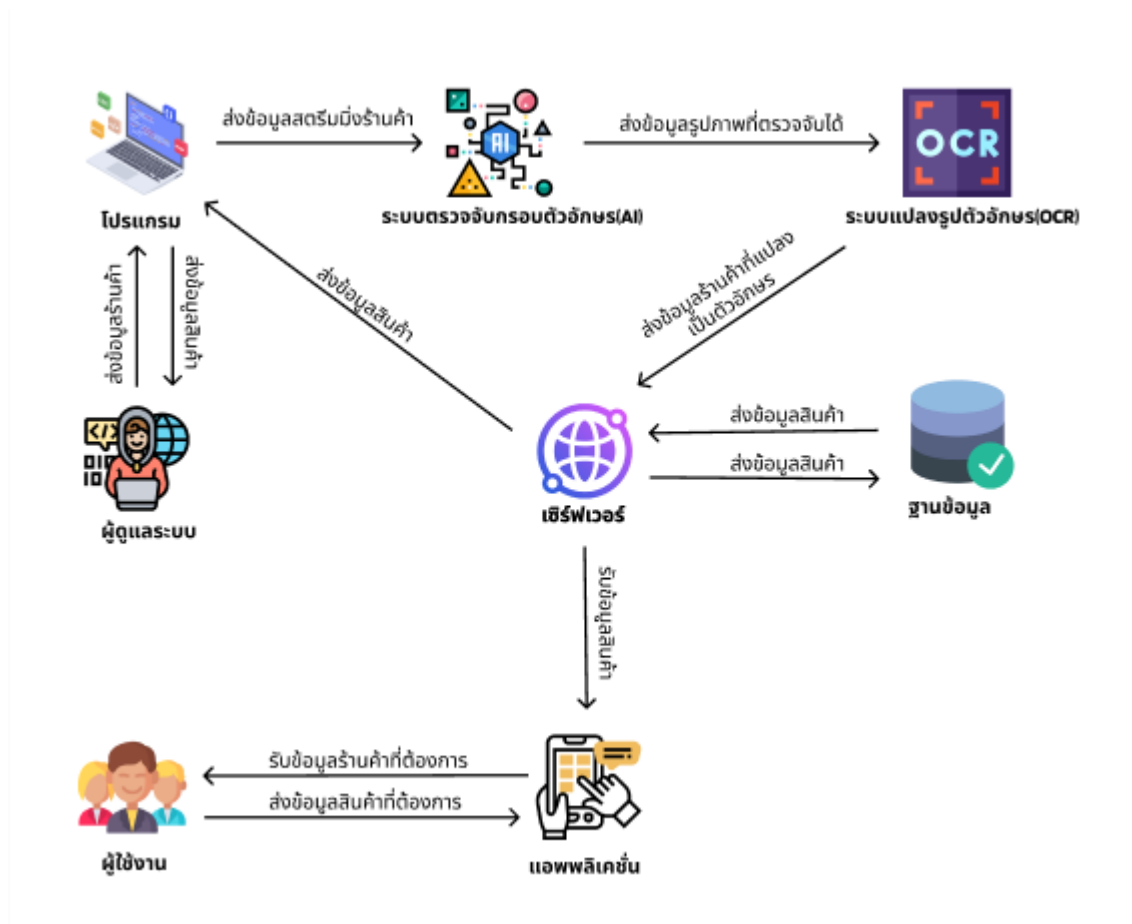
10. ระบบสามารถแจ้งเตือนผู้ใช้งานได้ เมื่อมีร้านค้าที่มีสถานะ online

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 ชื่อสินค้า ราคาสินค้า รูปภาพของโลโก้สติกเกอร์ที่ทำกรบันทึกมา ตัวอย่างของข้อมูลชื่อร้านค้า คือ ปิรันย่า บิวตี้ซ้อป รหัสสินค้าคือ ก22, ชื่อสินค้าคือ ดีท็อกซ์ญี่ปุ่น, ราคาสินค้าคือ 39 บาท เป็นต้น

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

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

7. ผู้ใช้ต้องเข้าสู่ระบบผ่านเฟสบุ๊คจึงจะสามารถเข้า application เพื่อเลือกสินค้าที่ต้องการและรอแจ้งเตือนร้านที่กำลังโลโก้สติกเกอร์ได้

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

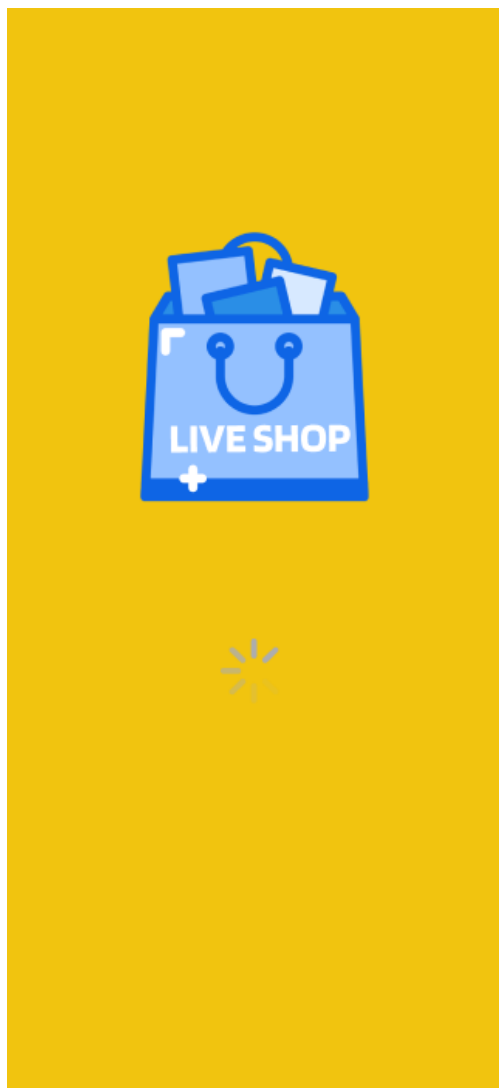
- ระบบปฏิบัติการ Windows 11
- Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz
- Installed RAM 16.0 GB (15.9 GB usable)

โครงการปริญญานิพนธ์ฉบับนี้ ดำเนินงาน ณ คณะวิทยาการสารสนเทศ
มหาวิทยาลัยมหาสารคาม ระหว่างเดือน พ.ค. 2565 ถึง ก.พ. 2566

	เดือน										
กิจกรรม	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษา และ รวบรวมข้อมูล											
2. กำหนดขอบเขต											
3. วิเคราะห์และ ออกแบบ											
4. พัฒนาโปรแกรม											
5. ทดสอบการใช้งาน โปรแกรม											
6. ทำรายงานสรุป											
7. นำเสนอโครงการงาน											

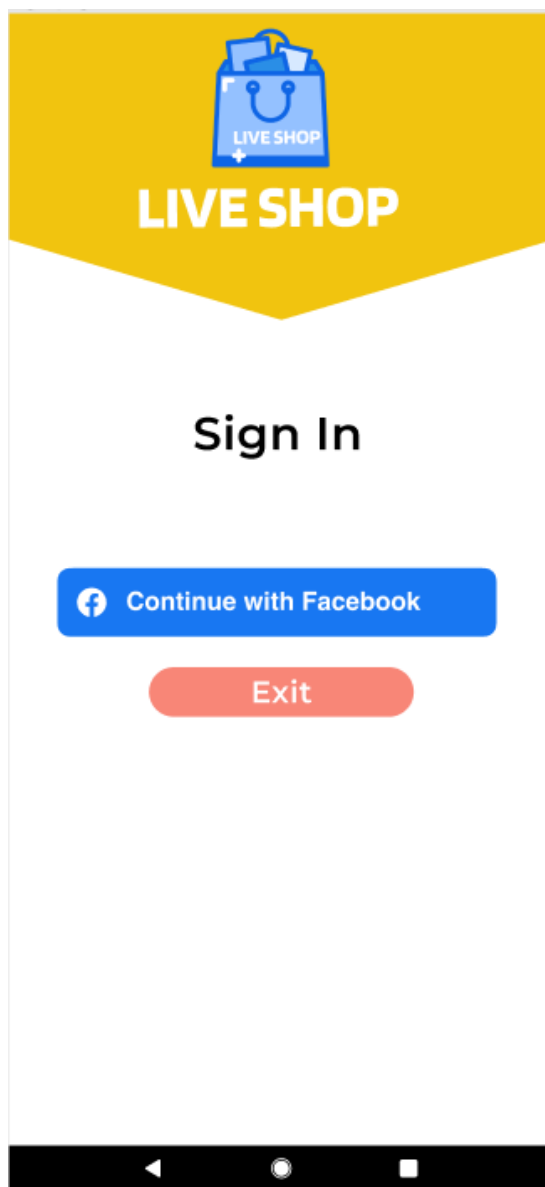
1.8 ตัวอย่างโปรแกรม

1.8.1 แอปพลิเคชันบนอุปกรณ์ไร้สายแบบเคลื่อนที่



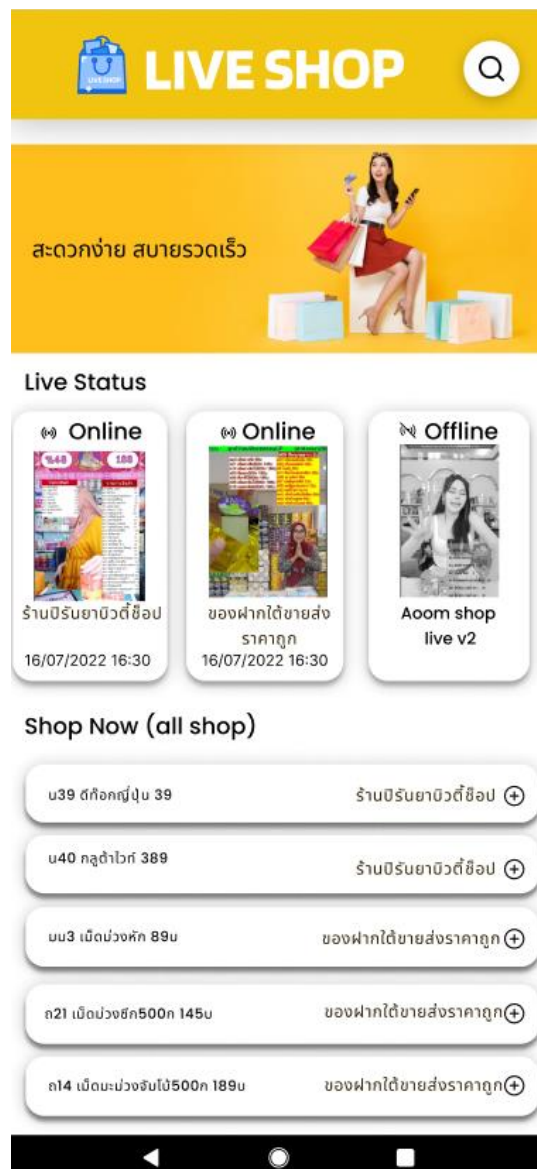
ภาพประกอบที่ 1.3 หน้าแรกของโปรแกรม

หน้าแรกของโปรแกรม จะมีการรอโหลดข้อมูลจาก Database เมื่อทำการโหลดข้อมูลเสร็จจะนำพาไปยังหน้าสื่อกอินทันที



ภาพประกอบที่ 1.4 หน้าล็อกอินเข้าสู่ระบบ

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



ภาพประกอบที่ 1.5 หน้าหลักของแอปพลิเคชัน

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



ภาพประกอบที่ 1.6 หน้าค้นหาสินค้า

หน้าค้นหาสินค้า จะมีช่องเพื่อใส่ข้อมูลชื่อของสินค้าหรือชื่อร้านค้าที่ต้องการค้นหา และเมื่อใส่ข้อมูลลงไปจะมีส่วนในการแสดงผลลัพธ์ ของการค้นหาเป็นรายการสินค้าขึ้นมาและเมื่อเลือกรายการสินค้าจะไปยังหน้ารายละเอียดสินค้าทันที



ภาพประกอบที่ 1.7 หน้ารายละเอียดสินค้า

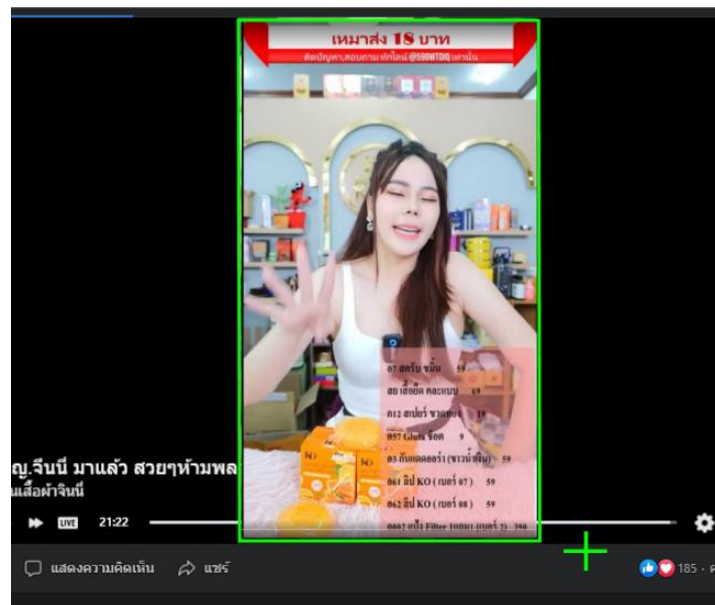
หน้ารายละเอียดสินค้า จะแสดงรูปภาพที่แคปมาจากไลฟ์สดและวันเวลาอัปเดตรูปภาพและแสดงชื่อร้านค้า ข้อมูลของสินค้าประกอบไปด้วย รหัสสินค้า ชื่อสินค้า ราคา และจะมีปุ่ม 2 ปุ่ม “Chat” เพื่อลิงก์ไปยังแชทของร้านค้าและ “Buy” เพื่อลิงก์ไปยังแชทของร้านค้าและนำข้อมูลสินค้าส่งไปอัตโนมัติ

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



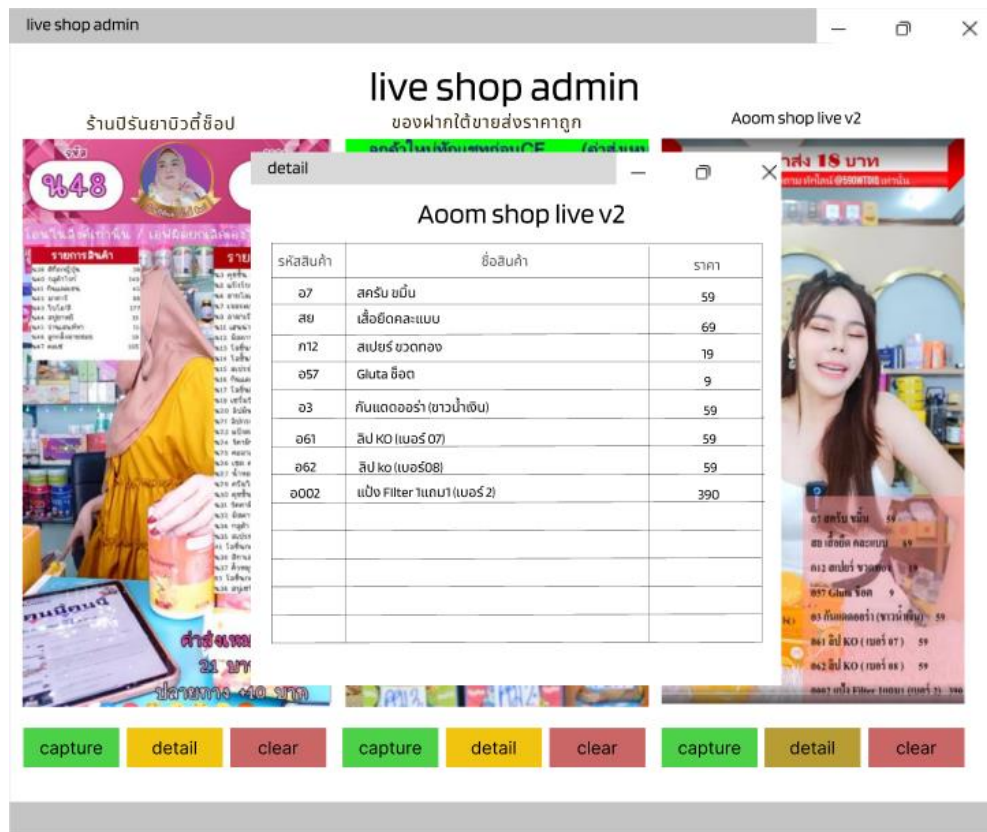
ภาพประกอบที่ 1.9 หน้าเลือกพื้นที่ภาพ

เมื่อกดปุ่ม “capture” แล้วต้องลาก cursor ครอบพื้นที่ที่ต้องการสตรีมจากไลฟ์สดเพื่อนำไปประมวลผล



ภาพประกอบที่ 1.10 หน้าผลลัพธ์ในการถ่ายภาพหน้าจอ

ผลลัพธ์ในการถ่ายภาพหน้าจอ ของปุ่ม Capture



ภาพประกอบที่ 1.11 หน้ารายละเอียดข้อมูล

หน้ารายละเอียดข้อมูล เมื่อกดที่ปุ่ม "detail" ในส่วนของสตริมมิ่งนั้นจะทำการแสดงข้อมูล รหัสสินค้า ชื่อสินค้า และราคาสินค้า ที่ระบบทำการประมวลผลแปลงรูปภาพเป็นข้อความ ออกมาโชว์ เป็น 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, Icon หรือรูปภาพที่ไม่มี 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 มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึงEmailและpasswordผู้ให้บริการบุคคลที่สามเช่น Google หรือ Facebook และใช้ระบบบัญชีที่คุณมีอยู่โดยตรง สร้างอินเทอร์เฟซของคุณเองหรือใช้ประโยชน์จากโอเพ่นซอร์ส UI ที่ปรับแต่งได้อย่างเต็มที่

- Hosting ลดความซับซ้อนของเว็บไซต์ของคุณด้วยเครื่องมือที่สร้างขึ้นเฉพาะสำหรับเว็บแอปสมัยใหม่ เมื่อคุณอัปโหลดเนื้อหาเว็บของคุณเราจะส่งเนื้อหาเหล่านั้นไปยัง CDN ทั่วโลกของเราโดยอัตโนมัติและมอบใบรับรอง SSL ฟรีเพื่อให้ผู้ใช้ของคุณได้รับประสบการณ์ที่ปลอดภัยเชื่อถือได้และมีเวลาแฝงต่ำไม่ว่าจะอยู่ที่ใดก็ตาม

- Realtime Database Realtime Database คือฐานข้อมูลดั้งเดิมของ Firebase เป็นโซลูชันที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับแอปบนอุปกรณ์เคลื่อนที่ที่ต้องการสถานะการชิงช้าระหว่างไคลเอนต์แบบเรียลไทม์ 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 ตลอดวงจรการพัฒนาของคุณเพื่อค้นหาจุดบกพร่องและความไม่สอดคล้องกันเพื่อให้คุณสามารถนำเสนอประสบการณ์ที่ยอดเยี่ยมบนอุปกรณ์หลากหลายประเภท

3.) Grow your business

- Google Analytics วิเคราะห์คุณลักษณะและพฤติกรรมของผู้ใช้ในแดชบอร์ดเดียวเพื่อทำการตัดสินใจอย่างชาญฉลาดเกี่ยวกับแผนงานผลิตภัณฑ์ของคุณ รับข้อมูลเชิงลึกแบบเรียลไทม์จากรายงานหรือส่งออกข้อมูลเหตุการณ์ไปยัง Google BigQuery สำหรับการวิเคราะห์ที่กำหนดเอง

- Remote Config กำหนดวิธีการแสดงผลแอปของคุณสำหรับผู้ใช้แต่ละคน เปลี่ยนรูปลักษณ์เปิดตัวฟีเจอร์ที่ละน้อยเรียกใช้การทดสอบ A / B ส่งมอบเนื้อหาที่กำหนดเองให้กับผู้ใช้บางรายหรือทำ

การอัปเดตอื่น ๆ โดยไม่ต้องปรับใช้เวอร์ชันใหม่ทั้งหมดนี้ทำได้จากคอนโซล Firebase ตรวจสอบผลกระทบของการเปลี่ยนแปลงของคุณและทำการปรับเปลี่ยนในเวลาไม่กี่ปาที

- 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 เพื่อนำภาพวัตถุที่ได้ ไปวิเคราะห์หาข้อมูลเชิงปริมาณ เช่น ขนาด รูปร่าง และทิศทางการเคลื่อนของวัตถุในภาพ คอมพิวเตอร์มีความสามารถในการคำนวณและประมวลผลข้อมูลจำนวนมหาศาลได้ในเวลาอันสั้นจึงมีประโยชน์อย่างมากในการเพิ่มประสิทธิภาพการประมวลผลภาพและวิเคราะห์ข้อมูลที่ได้จากภาพในระบบ ลักษณะของการใช้งานจะมีเป็น

1. การรู้จำภาพ (Image Recognition)

การรู้จำภาพ (Image Recognition) การรู้จำภาพเป็นกระบวนการในการระบุและการตรวจจับวัตถุหรือคุณสมบัติที่มีอยู่ในระบบดิจิทัล ซึ่งได้จากภาพหรือวิดีโอแนวคิดนี้ได้ถูกนำมาใช้ในแอปพลิเคชันมากมายที่เหมือนกันในระบบสำหรับการทำงานของระบบอัตโนมัติในโรงงานการตรวจสอบด้านเก็บเงินด้วยการรักษาความปลอดภัยและการรักษาความปลอดภัย อัลกอริทึมการรู้จำภาพทั่วไป

2. การสกัดคุณลักษณะเด่น (Feature Extraction)

การสกัดคุณลักษณะเด่น (Feature Extraction) เป็นการหาลักษณะเด่นที่ได้จากภาพ โดยผลลัพธ์ที่ได้จากการนำไปหาลักษณะเด่นจะบ่งบอกถึงลักษณะเช่น ความสูง ความยาว สีหรือเวกเตอร์ ที่เป็นคุณลักษณะ ที่สำคัญของรูปภาพ เป็นต้น นอกเหนือจากข้อมูลคุณลักษณะดังกล่าว ในปัจจุบันยังมีวิธีการการสกัดหา ลักษณะเด่นที่นิยมใช้กันอย่างแพร่หลายที่ใช้ในการต่อภาพคือ การวางตัวของเส้นขอบ(Edge) การหามุม (Corner) และการหาพื้นที่ที่น่าสนใจ (Blob) เป็นต้น



ภาพประกอบที่ 2.2 การตรวจจับและประมวลผลใบหน้าจากภาพ [4]

ตัวอย่างการ นำการประมวลผลภาพไปใช้งาน เช่น

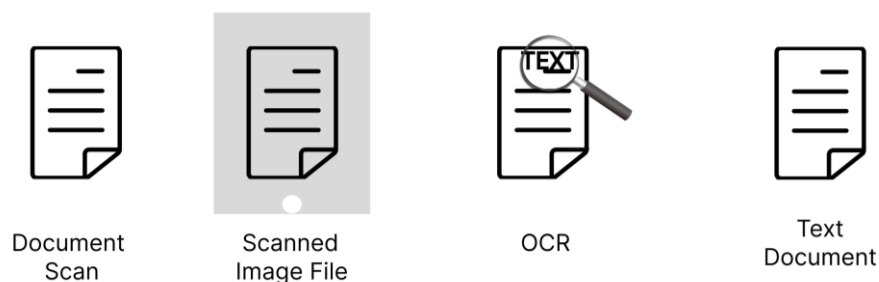
- ระบบตรวจจับใบหน้าในกล้องดิจิทัล โดยกล้องจะมีระบบตรวจว่าส่วนไหนของภาพมี ลักษณะคล้าย ใบหน้า แล้วกล้องก็จะทำการโฟกัสตำแหน่งที่ตรวจจับเพื่อภาพมีความคมชัดมากขึ้น เช่น ระยะห่าง ระหว่างคิ้วมูปาก โหนกแก้ม จมูก โครงหน้า
- ระบบตรวจจับการเคลื่อนไหวเพื่อรักษาความปลอดภัย โดยกล้องจะเปรียบเทียบภาพก่อน หน้าและ ภาพปัจจุบันถ้ามีส่วนใดส่วนใดเปลี่ยนแปลงระบบจะบันทึกเฉพาะภาพที่มีการเปลี่ยนแปลงทำให้ ประหยัด เนื้อที่ในการเก็บภาพและสามารถตรวจสอบได้ภาพหลัง
- ระบบอ่านคิวบาร์โค้ด โดยระบบนี้จะอ่านรหัสจากบาร์โค้ด แล้วแปลงเป็นข้อมูลซึ่งสามารถ นำมาใช้ งานได้ เช่น QR code, Microsoft tag ภาพที่นำมาประมวลผลนั้นเป็นได้ทั้งภาพนิ่งและ ภาพเคลื่อนไหว ในการนำภาพนิ่งเข้ามาประมวลผลนั้น โปรแกรมจะอ่านไฟล์ภาพขึ้นมาแล้วถอดรหัสจาก ค่าสีของภาพ แต่ละจุดเป็นตัวเลขเพื่อนำมาประมวลผลสำหรับการประมวลผลภาพเคลื่อนไหวนั้น โปรแกรมจะ มองเห็นเป็นภาพนิ่งหลายภาพที่เรียงต่อกัน จึงมีประโยชน์อย่าง มากในการเพิ่ม ประสิทธิภาพการ ประมวลผลภาพและวิเคราะห์ข้อมูลได้จากภาพในระบบต่าง ๆ ดังกล่าวข้างต้น การประมวลผลภาพ สามารถจำแนกตาม วัตถุประสงค์ในการใช้งานดังนี้
- การปรับปรุงคุณภาพของภาพ (Image Enhancement and Restoration) เช่น ปรับภาพ ให้คมชัด มากขึ้นการปรับ Contrast หรือการปรับเน้นเส้นขอบภาพ (Edge Enhancement) หรือการ กรอง สัญญาณรบกวน (Image Filtering) เพื่อกำจัดสัญญาณรบกวนได้
- การบีบอัดภาพ (Image Compression) เนื่องจากข้อมูลภาพนั้นมีขนาดใหญ่มากโดยเฉพาะ ภาพสี ซึ่งจะ ทำให้การจัดเก็บหรือรับส่งไฟล์ข้อมูลขนาดใหญ่เสียเวลามาก เพื่อทำให้ขนาดภาพเล็กลงจึง ทำให้

การบีบอัด ข้อมูลภาพ เช่น JPEG หรือ GIF และตัวอย่างการบีบอัดข้อมูลวิดีโอ เช่น MPEG หรือ AVI เป็นต้น

- การวิเคราะห์ข้อมูลภาพ (Image Analysis) เป็นการสังเคราะห์ ข้อมูลที่มีความหมายจากภาพ แล้วนำไปใช้งาน ซึ่งอาจต้องมีรู้จำวัตถุ (object recognition) การแยกส่วนวัตถุ 317 The Tenth National conference on Computing and Information Technology NCCIT4 (segmentation) ตรวจจับการ 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 ไฟล์เอกสารที่มาในรูปแบบ PDF มีข้อดีก็คือไม่สามารถปลอมแปลง แก้ไขได้ แต่ถ้าต้องการแก้ไขข้อความใด ๆ ในนั้นก็เป็นเรื่องยากลำบาก วิธีแก้ไขปัญหาก็คือ แปลงไฟล์ PDF ให้เป็น OCR ซึ่งวิธีการที่ทุกคนคุ้นเคยก็คือการนำแผ่นเอกสาร มาสแกนข้อความ ตัวอักษร เส้นตารางและอื่น ๆ ให้เป็นไฟล์ Word หรือ PDF ด้วยเครื่องสแกนเนอร์ 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] ที่พัฒนาระบบตรวจสอบความปลอดภัยของคนงาน ซึ่งระบบจะทำการตรวจจับภาพของคนงาน และ จะแสดงกรอบที่มีคำอธิบาย โดยได้นำรูปภาพจากอินเทอร์เน็ตจำนวน 6045 ภาพมาทำการทดสอบ โดยสีระของ คนงานที่ไม่สวมหมวกนิรภัยจะมีคำอธิบายว่า "Alarm" และสำหรับสีระของคนงานที่สวมหมวกนิรภัยจะมีคำอธิบายว่า "Helmet" โดยใช้ algorithm ที่ทันสมัยที่สุดในการตรวจสอบ algorithm งานตรวจจับวัตถุถูกใช้อย่างแพร่หลายในความเป็นจริง เป้าหมายของการตรวจจับคือการค้นหาวัตถุทั้งหมด

ที่น่าสนใจในภาพ ซึ่งจะมี 2 งานย่อยคือการกำหนดหมวดหมู่และการระบุตำแหน่งของวัตถุ แม้ว่าอัลกอริธึมการตรวจจับแบบดั้งเดิมสามารถทำงานได้ผลลัพธ์ที่ดี ในบางสถานการณ์ในสภาพแวดล้อมที่แปรปรวน เช่น การเปลี่ยนแปลงของสภาพอากาศ การกระจายตัวของคนงานที่ไม่สม่ำเสมอ และหมวกกันน็อคแบบต่างๆ ทำให้ความแม่นยำนั้นรับประกันได้ยากว่ามันสามารถทำงานได้ถูกต้อง จึงได้นำตัวYOLOv5มาใช้ซึ่งมีความเร็วและความแม่นยำสูง



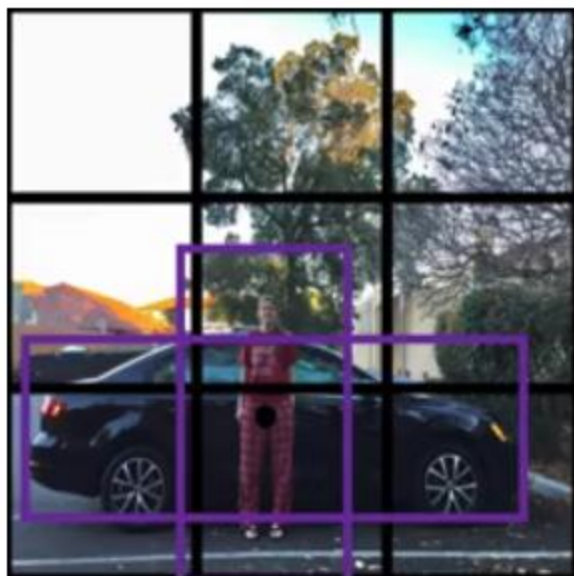
ภาพประกอบที่ 2.4 ภาพการตรวจจับหมวกนิรภัยของคนงาน [8]

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



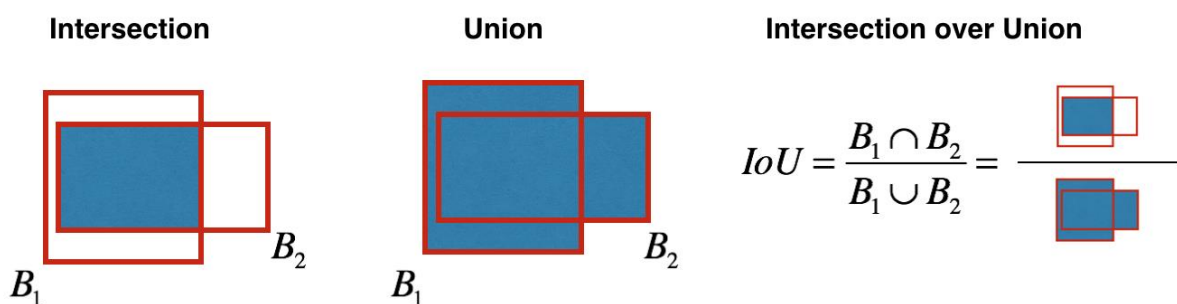
ภาพประกอบที่ 2.5 การแบ่งช่องของรูปภาพออกเป็น grid [9]

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 ว่าอยู่ที่พิกัดไหนส่วน 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 โดยเราสามารถกำหนด



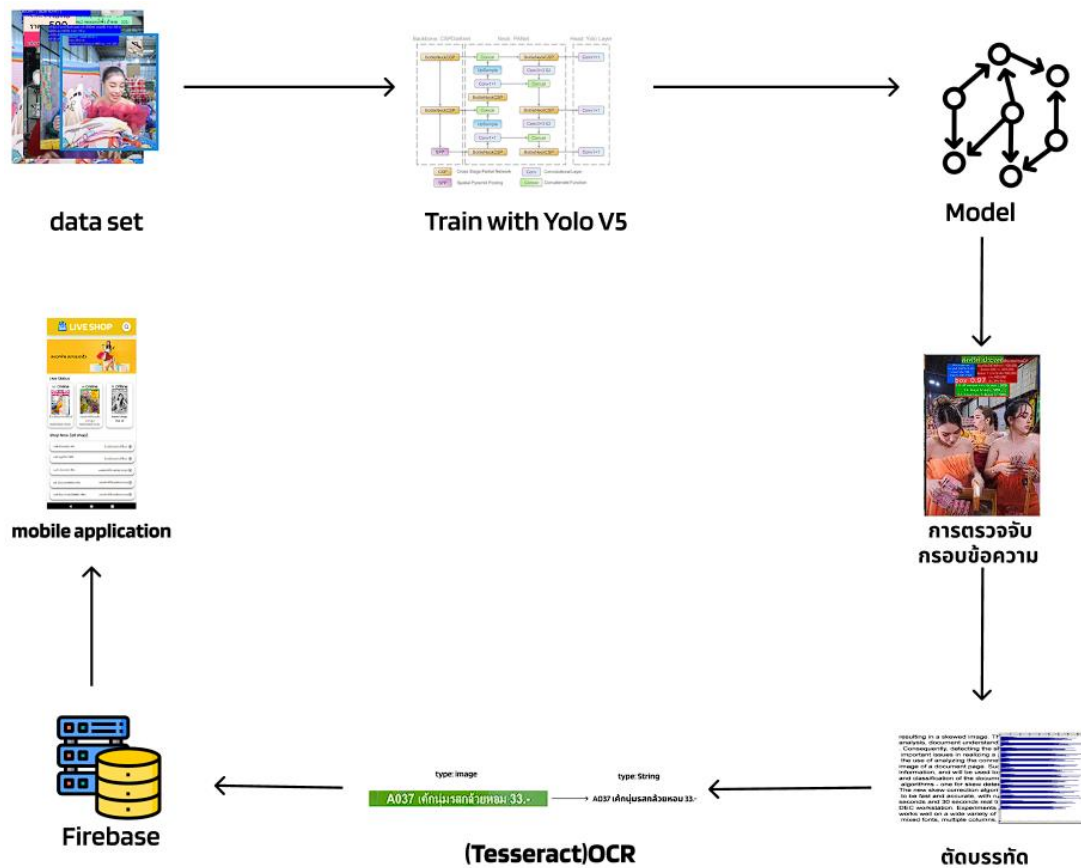
ภาพประกอบที่ 2.7 การทำงานของขบวนการ IOU [9]

จำนวน Box นี้ได้ ก็จะมี Label แบบด้านบน 2 อันใน 1 Grid ได้ และตัว YOLO ก็จะคำนวณให้ว่ารูปนั้นใกล้ Anchor อันไหนสุดจากค่า IOU ก็จะถูกกำหนดไปที่ Anchor นั้น

IOU จะมีส่วนหลักๆหลายส่วนใน YOLO เช่นเวลาโมเดลทำนายออกมา มันอาจจะทำนายรูปรถเดียวกันแต่หลายกล่องได้ จึงต้องใช้ metric IOU ในเลือกกล่องเดียวเป็นตัวแทนของ Object โดยหลักการคำนวณคือ หาส่วนที่ Intersect หาส่วนที่ Union กัน ถ้ากล่องใดมีค่านี้สูง เกินเกณฑ์ที่กำหนด แสดงว่ามันคือ Object เดียวกัน

บทที่ 3

ขั้นตอนการดำเนินงาน

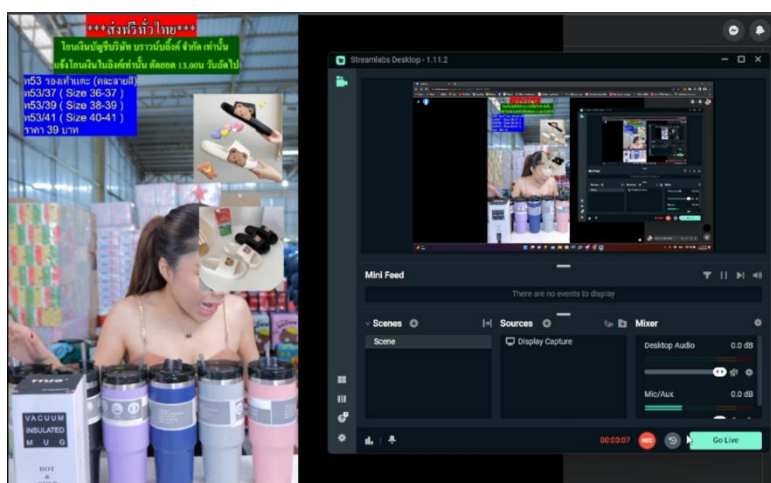


ภาพประกอบที่ 3.1 ขั้นตอนการดำเนินงานของระบบ

สำหรับในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงการปริญญาโทซึ่งจะทำให้ทราบถึงการวิเคราะห์และการออกแบบแอปพลิเคชันโดยละเอียดว่ามีแนวทางในการดำเนินงานหรือมีขั้นตอนในการทำงานของแอปพลิเคชันอย่างไรบ้างโดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้ 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.ร้าน KANYA SHOP ขายถูกทุกอย่าง 3.ร้านมหัศจรรย์"วันของAuuM" เจื่อนใจในการเก็บ โดยมีการเก็บวิดีโอเป็น 1 ร้านจะมี 3 วิดีโอไลฟ์สด 1 วิดีโอจะแบ่งเป็น 6 คลิปวิดีโอ 1 คลิปวิดีโอ จะแบ่งเป็น คลิปละ 3 นาที



ภาพประกอบที่ 3.2 Streamlabs Desktop

ตารางที่ 2 ตารางจำนวนและเวลาเฉลี่ยของวิดีโอในแต่ละร้านค้า

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

3.2 การเตรียมข้อมูล

3.2.1 การเตรียมข้อมูลในการ training

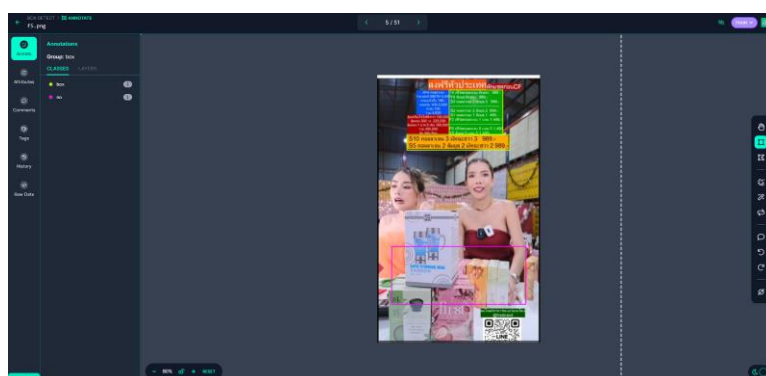
เตรียมข้อมูลในการ training โดยบันทึกรูปภาพจาก Video โลโก้สดที่เก็บมาได้โดยจะทำการบันทึกเฉพาะช่วงที่มีการเปลี่ยนแปลงของข้อความบนหน้าจอโลโก้สดดังกล่าว โดยจะได้จาก ร้าน FIRST SHOP V2 จำนวน 33 ภาพ จากร้าน KANYA SHOP ขยายทุกทุกอย่าง จำนวน 28 ภาพ จากร้าน มหัทศจรย์"วันของAuuM" จำนวน 33 ภาพ รวมกันทั้งหมด 92 รูปภาพ



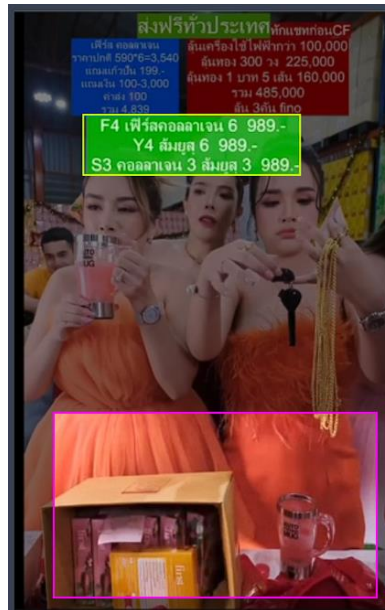
ภาพประกอบที่ 3.3 ตัวอย่างรูปภาพที่ทำการเก็บ

3.2.2 การวาดภาพผลเฉลย

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



ภาพประกอบที่ 3.4 การวาดภาพผลเฉลยรูปภาพใน Roboflow



ภาพประกอบที่ 3.5 การวาดภาพผลเฉลยรูปภาพใน Roboflow (2)

C415 กล้องเก็บของลายรถ(คละสี) 79.-
เสื้อ1 เสื้อคละแบบไมโคร 69.-
A037 เสื้อกันหนาวสลายหอม 33.-
E589 สานห่วยเหมประสบาร์ปิกว 25.-

ข1 บิวเวอรี่ยู1 แพ็ค 9.-
ข2 ชุดกีฬาม Makiatto รสสตอเบอร์รี่/จ๊อก 60
ขนม2 เจลลี่บุก yogurtคละรส 60.-
ผอ1 อินทผาลัม ดูนีเซีย 500 กรัม100.-
กท9 กระดาษเคลือบได้พื้นทรงลึกปากท 80
กท10 กระดาษได้พื้นแบน 80

ท53 รองเท้าแตะ (คละลายสี)
ท53/37 (Size 36-37)
ท53/39 (Size 38-39)
ท53/41 (Size 40-41)
ราคา 39 บาท
ด083 แก้วยัดเก็บถวามเย็น (คละสี) ราคา 100 บ

ภาพประกอบที่ 3.6 ภาพผลเฉลยที่ถูกตัดจากทั้ง 3 ร้านค้า

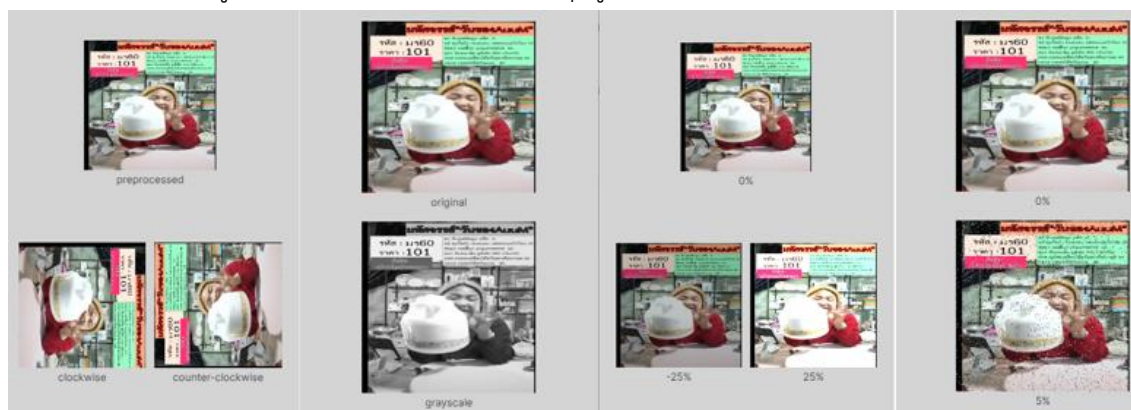
ตารางที่ 3 จำนวนของข้อมูลในแต่ละส่วน

Training Set	Validation Set	Testing Set
62	17	13

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

3.2.3 การทำ Data Augmentation

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



ภาพประกอบที่ 3.7 การเพิ่มจำนวนรูปภาพใน Traing Set

จะทำการ Generate ทั้งหมด 3 ครั้ง โดยจะได้รูปภาพจากการ Generate ทั้งหมด 186 ภาพจะมีรูปภาพที่ใช้ในการ train ทั้งหมด 558

ตารางที่ 4 จำนวนของข้อมูลก่อนและหลังทำการ Generate

	Training Set	Validation Set	Testing Set
ก่อน Generate	62	17	13
หลัง Generate	186	17	13
หลัง Generate 3 ครั้ง	558	17	13



ภาพประกอบที่ 3.8 ตัวอย่างรูปภาพที่ Generate ออกมา

0 0.6838942307692307 0.12259615384615384 0.4951923076923077 0.07211538461538461
1 0.28846153846153844 0.8161057692307693 0.5 0.1875

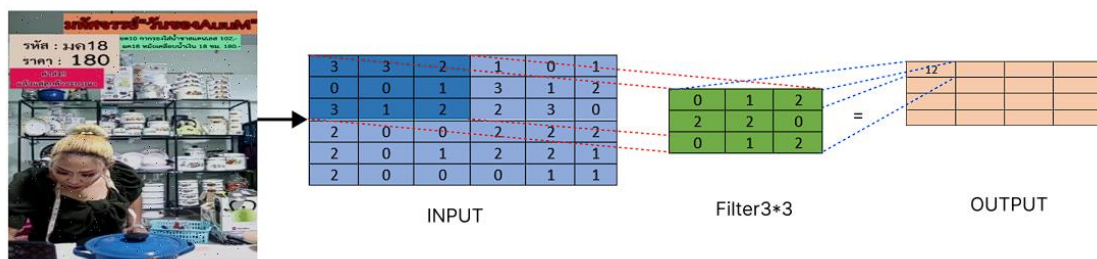
ภาพประกอบที่ 3.9 ผลเฉลยของรูปภาพที่ Generate ออกมา

โดยในสิ่งที่ผลเฉลยอธิบายจะมีดังนี้ 1.คลาสของสิ่งที่สนใจในรูป 2.ตำแหน่งเริ่มต้นของกรอบสิ่งที่ detect เจอ 3.ตำแหน่งสิ้นสุดของกรอบสิ่งที่ detect เจอ โดยจะนำตำแหน่งที่ได้มาหารกับค่า ความกว้างและความสูงของรูปภาพ คือ 416×416 จึงได้ค่าออกมาเป็นทศนิยมเพื่อจะได้นำไปปรับใช้กับหน้าจอได้หลายขนาด

3.3 การสร้างโมเดลโดยใช้ yoloV5

ในส่วนนี้จะเป็นการอธิบายถึงโครงสร้างของ yoloV5 และการเทรนข้อมูลโดยจะมีขั้นตอนการทำงานแบ่งเป็นส่วนหลักๆดังนี้

ในส่วนที่ 1 จะเป็นการคัดกรองภาพเพื่อดึงลักษณะเด่นของรูปภาพออกมาโดยจะมีขั้นตอนการทำงานดังนี้



ภาพประกอบที่ 3.10 การทำ Convolution

3.3.1 Convolution Layer

การทำ Convolution Layer เพื่อสกัดเอาส่วนต่างๆ ของภาพออกมา เช่น เส้นขอบของวัตถุต่างๆ เพื่อให้โมเดลสามารถเรียนรู้ลักษณะของภาพได้อย่างมีประสิทธิภาพและแม่นยำโดยขั้นตอนจะมีรูปภาพที่รับเข้ามาเป็น Matrix input ขนาด 6x6 เป็นรูปภาพของเราและมี Filter ขนาด 3x3 จะนำเฉพาะ Filter 3x3 กับ ช่องแรกๆของ Matrix input แรก มาคูณ กับ Filter matrix แล้วนำผลที่ได้แต่ละค่า มาบวกกัน แล้วนำไปใส่ในแถวแรกคอลัมน์แรกของ Matrix output ซึ่งเป็นผลลัพธ์ โดยในภาพผลลัพธ์ในช่องแรกโดย output จะเท่ากับ

$$Output = (I1 * f1) + (I2 * f2) + (I3 * f3) + (I4 * f4) + (I5 * f5) + (I6 * f6) + (I7 * f7) + (I8 * f8) + (I9 * f9)$$

เมื่อแทนค่าในสมการจะได้ดังต่อไปนี้

$$Output = (3 * 0) + (3 * 1) + (2 * 2) + (0 * 2) + (0 * 2) + (1 * 0) + (3 * 0) + (1 * 1) + (2 * 2) = 12$$

โดยนำถัดมาจะเลื่อนกรอบขนาด 3x3 ใน Matrix แรกไปทางขวา 1 ช่อง แล้วทำแบบเดิม ผลลัพธ์ที่ได้นำไปใส่ในแถว 1 ช่อง 2 ของ Matrix output โดยจะทำได้เรื่อยๆ จนสุดแล้วเลื่อน กรอบ 3x3 ลงมาด้านล่าง 1 ช่อง แล้วทำแบบเดิม จนกระทั่งเติมค่าใน Matrix ผลลัพธ์จนเต็ม

3.3.2 Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) เป็นฟังก์ชัน Activation Function ที่นิยมใช้ใน Deep Learning เพื่อปรับค่าผลลัพธ์ให้เป็นค่าบวก โดยถ้า x มีค่าเป็นลบจะแทนค่า x ตัวนั้นกลายเป็น 0 โดยหาได้จากสมการต่อไปนี้

$$f(x) = \max(0, x)$$

ตัวอย่างการคำนวณReLU

-1	1	→	0	1
0	1		0	1

ภาพประกอบที่ 3.11 การคำนวณReLU

3.3.3 ReLU leaky

ฟังก์ชัน ReLU นั้นมีจุดขาดที่เรียกว่า ReLU leaky โดยจุดขาดนี้เกิดจากเมื่อค่าอินพุตที่ผ่านมามีค่าติดลบมาก ๆ จะทำให้ gradient หายไปและไม่สามารถปรับค่าได้ จึงมีการพัฒนา ReLU leaky เพื่อแก้ไขปัญหานี้ ReLU leaky จะไม่ให้ค่าลบเป็น 0 แต่จะแทนที่ด้วยค่าที่เล็กๆ โดยที่ค่านี้จะกำหนดได้ โดยทั่วไปจะใช้ค่าเดียวกับ alpha ซึ่งเป็นค่าเล็กๆ อยู่ในช่วง 0 ถึง 1 ซึ่งช่วยให้ gradient ยังคงมีค่าได้ และช่วยให้โมเดลมีความสามารถในการเรียนรู้ได้ดีขึ้นโดยมีสมการดังต่อไปนี้

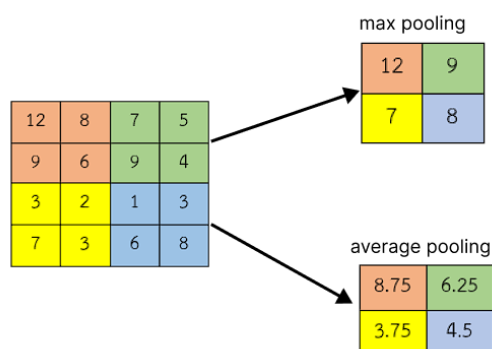
$$f(x) = \{ x \text{ (if } x > 0), \alpha * x \text{ (if } x \leq 0) \}$$

ตัวอย่างการคำนวณ ReLU leaky

-5	1	→	0.05	1
0	1		0	1

ภาพประกอบที่ 3.12 การคำนวณReLU leaky

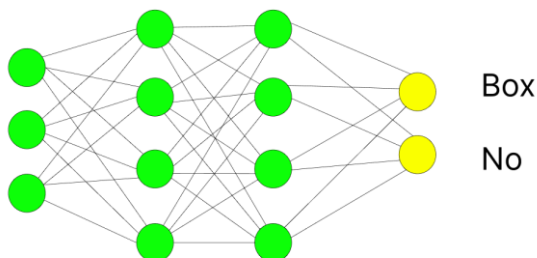
หลังจากทำการ ReLU เพื่อทำให้ค่าเป็นบวกแล้วจะทำการ Pooling layer เพื่อทำการการสกัดหรือลดขนาดของข้อมูล เพื่อเอาส่วนที่สำคัญที่สุดของข้อมูลโดยทั่วไปมักจะเลือกใช้ max pooling หรือ average pooling



ภาพประกอบที่ 3.13 Pooling layer

3.3.4 Pooling layer

การทำ Pooling layer คือการสกัดหรือลดขนาดของข้อมูล เพื่อเอาส่วนที่สำคัญที่สุดของข้อมูล และเพิ่มประสิทธิภาพการประมวลผลให้รวดเร็วยิ่งขึ้น โดย Max Pooling layer คือการสกัดเอาเฉพาะค่าสูงสุดของ Matrix เก็บไว้ใน Output ส่วน Average Pooling layer คือการสกัดเอาเฉพาะค่าเฉลี่ยของ Matrix เก็บไว้ใน Output เช่นจากภาพ Max Pooling layer ที่มีขนาด 2x2

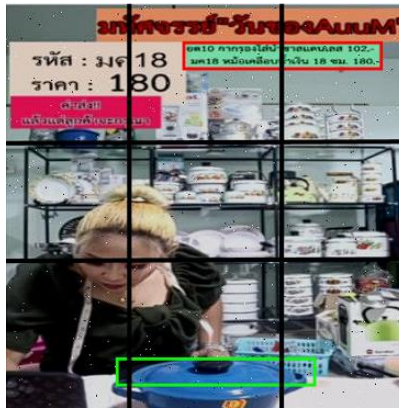


ภาพประกอบที่ 3.14 Fully Connected Layer

3.3.5 Fully Connected Layer

Fully Connected Layer คือการรวมผลลัพธ์ของตัวแปร input จาก Layer ก่อนหน้าทั้งหมด และคูณด้วยน้ำหนัก (weight) ของแต่ละโหนด (neuron) ใน Fully-Connected Layer นี้ โดยที่แต่ละโหนดจะมี weight และ bias ของตัวเองที่แตกต่างกันไป การคูณนี้จะทำให้ได้ผลลัพธ์เป็นเวกเตอร์หรือเมทริกซ์ขนาดเล็กกว่า input และเมทริกซ์นี้จะถูกส่งต่อไปยัง Layer ถัดไป

ส่วนที่ 2 เป็นส่วนที่ทำหน้าที่ Detect Object โดยใช้การสร้าง bounding box โดยการคำนวณพิกัดของ object ที่ Convolution Layer เพื่อทำนายความน่าจะเป็นของแต่ละ Object Class และ bounding box coordinates



ภาพประกอบที่ 3.15 รูปภาพ input ที่ใส่ grid

การกำหนดข้อมูลเทรนจะมีขั้นตอนดังต่อไปนี้ เราจะต้องส่งข้อมูลที่วาดภาพผลเฉลยแล้วไปยังโมเดลเพื่อฝึกฝน และจะแบ่งภาพออกเป็นตารางขนาด 3×3 และมีทั้งหมด 2 คลาสที่ต้องการให้วัตถุถูกจัดประเภท ซึ่งใน 2 คลาสจะมี คลาส 'box' ที่เป็นจุดที่เป็นกรอบของข้อความและคลาส 'no' เป็นจุดที่ไม่สนใจ ดังนั้นสำหรับแต่ละเซลล์ใน grid จะมี ค่าผลเฉลย y เจ็ดค่า ดังต่อไปนี้

ตารางที่ 5 ค่า parameter ใน ค่าผลเฉลย y ทั้ง 7

y	pc
	bx
	by
	bh
	bw
	c1
	c2

pc คือ ค่าที่กำหนดว่าวัตถุอยู่ในตารางหรือไม่

bx คือ ค่าที่ระบุตำแหน่งในแกน x ของ Bounding Box ของวัตถุเมื่อมีวัตถุภายในภาพ

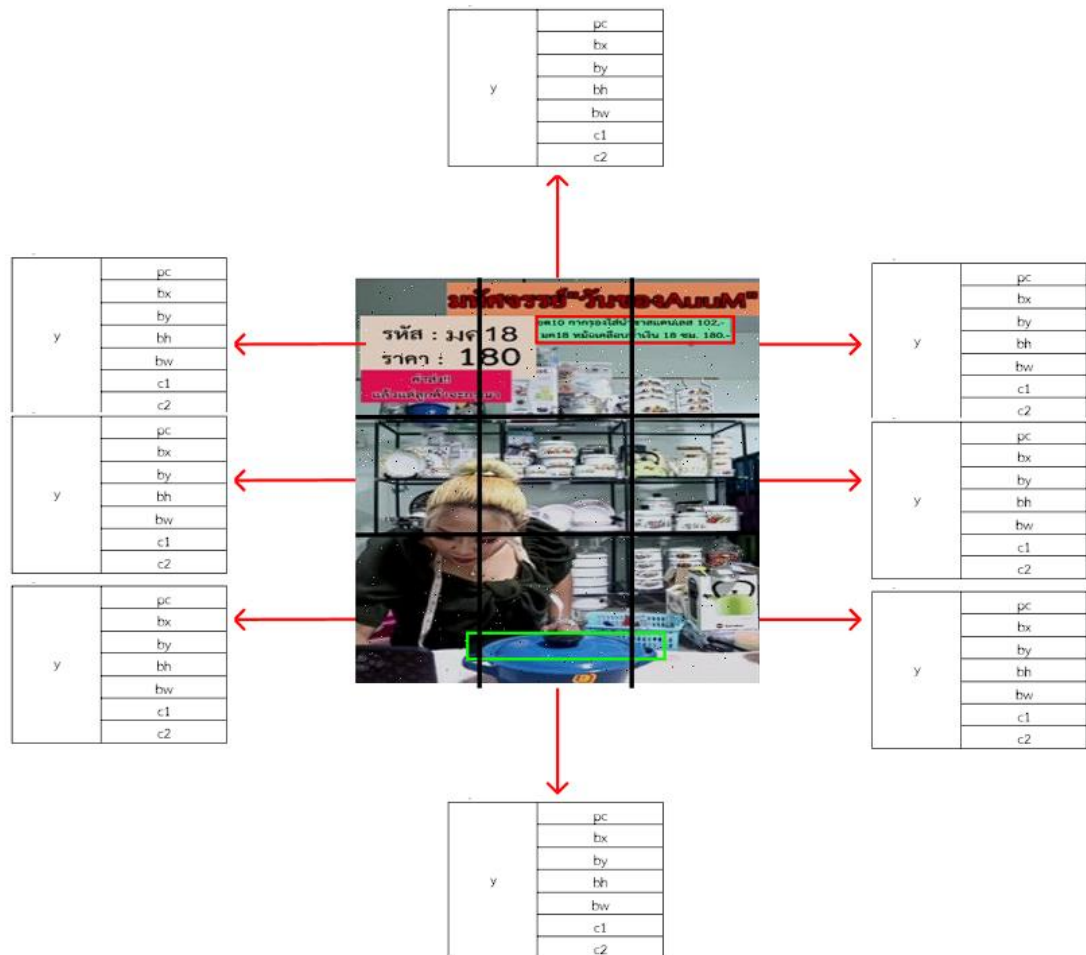
by คือ ค่าที่ระบุตำแหน่งในแกน y ของ Bounding Box ของวัตถุเมื่อมีวัตถุภายในภาพ

bh คือ ค่าที่ระบุตำแหน่งความสูงของ Bounding Box ของวัตถุเมื่อมีวัตถุภายในภาพ

bw คือ ค่าที่ระบุตำแหน่งความกว้างของ Bounding Box ของวัตถุเมื่อมีวัตถุภายในภาพ

c1 คือ คลาสของวัตถุที่เจอในภาพที่เจอโดยคลาสนี้จะมีชื่อว่า 'box' ถ้าเจอวัตถุจะดังกล่าวน่าจะเป็น 1 ถ้าไม่ใช่จะเป็น 0

c2 คือ คลาสของวัตถุที่เจอในภาพที่เจอโดยคลาสนี้จะมีชื่อว่า 'no' ถ้าเจอวัตถุจะดังกล่าวน่าจะเป็น 1 ถ้าไม่ใช่จะเป็น 0



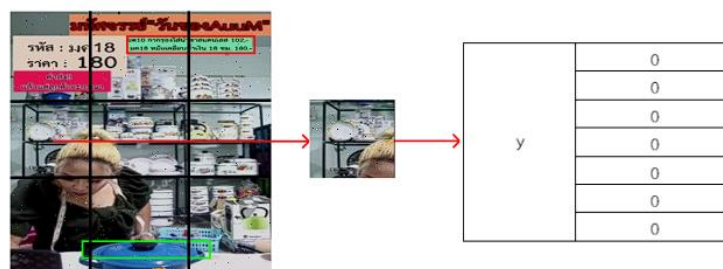
ภาพประกอบที่ 3.16 ค่าผลเฉลี่ย y ในแต่ละ grid cell

การกำหนด input ของรูปภาพสำหรับ grid นี้ YOLO จะตัดสินใจว่ามีวัตถุอยู่ใน grid จริงหรือไม่โดย YOLO จะใช้จุดกึ่งกลางของวัตถุและวัตถุเหล่านั้นจะถูกกำหนดให้กับ grid ที่มีจุดกึ่งกลางของวัตถุนั้น มี ค่าผลเฉลย y สำหรับ grid ช่องนั้นๆ ถ้าเจอวัตถุในภาพ



ภาพประกอบที่ 3.17 ตัวอย่าง grid ที่มีวัตถุในภาพและการแทนค่า

ในกรณีที่เจอวัตถุใน grid ในตารางนี้ $pc = 1$ และในส่วน bx, by, bh, bw จะถูกคำนวณเทียบกับเซลล์ grid ที่ทำการคำนวณอยู่ในภายหลัง และเนื่องจากเจอกรอบข้อความในรูปภาพ ค่า $c1 = 1$ และ $c2 = 0$



ภาพประกอบที่ 3.18 ตัวอย่าง grid ที่ไม่มีวัตถุในภาพและการแทนค่า

ในกรณีที่ไม่มีวัตถุใน grid ในตารางนี้ $pc = 0$ เมื่อมีค่า $pc = 0$ เท่ากับว่าไม่มีวัตถุที่สนใจภายในภาพเราจะไม่แทนค่าต่อใน grid cell นี้



ภาพประกอบที่ 3.19 จุดกึ่งกลางของ Bounding Box

3.3.6 การแทนค่าความสูงและความกว้างของ Bounding Box

Bounding Box คือการ วาดกล่องรอบวัตถุ ดังที่ได้กล่าวไว้ก่อนหน้านี้โดย bx , by , bh , bw จะถูกคำนวณเทียบกับเซลล์ grid ที่ทำการคำนวณอยู่ พิจารณา grid ตรงที่มีกรอบของตัวอักษร bx คือพิกัด x ของจุดกึ่งกลางของวัตถุที่อยู่ใน grid นี้ ในกรณีนี้ $bx = 0.9$ โดยจะวัดจากจุดกึ่งกลาง by คือพิกัด y ของจุดกึ่งกลางของวัตถุที่อยู่ใน grid นี้ ในกรณีนี้ $by = 0.4$ โดยจะวัดจากจุดกึ่งกลาง bh คืออัตราส่วนของความสูงของ Bounding Box กับความสูงของเซลล์ grid ที่เกี่ยวข้องซึ่งในกรณีของรูปภาพในตัวอย่างจะมีค่าคือประมาณ $bh = 0.3$ โดยประมาณ bw คืออัตราส่วนของความกว้างของ Bounding Box กับความกว้างของเซลล์ grid ที่เกี่ยวข้องซึ่งในกรณีของรูปภาพในตัวอย่างจะมีค่าคือประมาณ $bw = 1.5$ โดยประมาณ โดยค่า y สำหรับ grid นี้จะมีค่าดังนี้

ตารางที่ 6 การแทนค่า bx , by , bh , bw

y	1
	0.9
	0.4
	0.3
	1.5
	1
	0

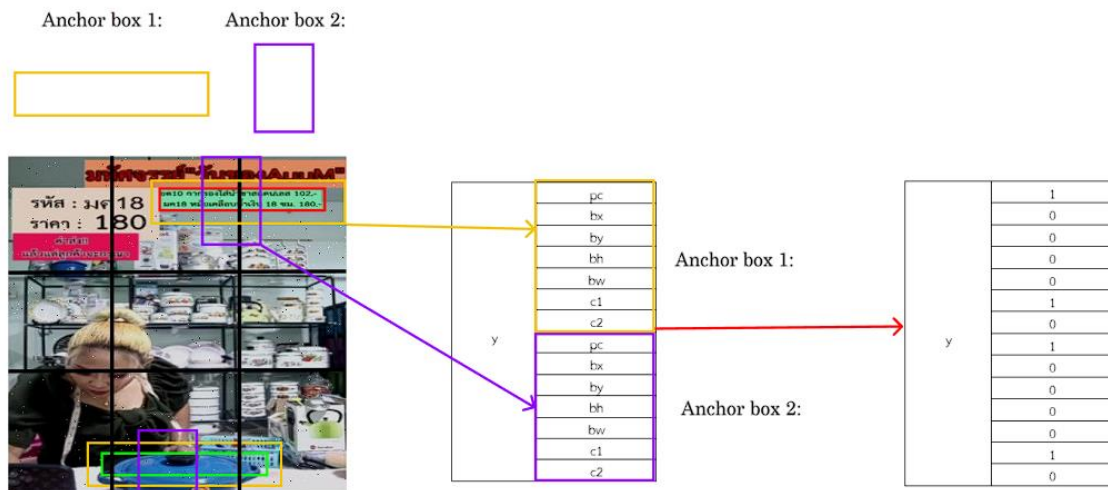
bx และ by จะอยู่ในช่วงระหว่าง 0 ถึง 1 เสมอเนื่องจากจุดกึ่งกลางจะอยู่ภายใน grid เสมอ ขณะที่ bh และ bw สามารถมากกว่า 1 ในกรณีที่ Bounding Box มากกว่าขนาดของ grid

3.3.7 การทำ anchor box

เป็นการสร้าง anchor box ที่มีรูปร่างต่างกันเพื่อเพิ่มความแม่นยำในการตรวจจับของโมเดลโดยถ้ามี anchor box 2 กล่อง ค่า label y โดยที่ค่า parameter 7 แถวแรกเป็นของ anchor box ที่ 1 และอีก 7 แถวที่เหลือเป็นของ anchor box ที่ 2 โดยจะเปลี่ยนแปลงไปดังนี้

ตารางที่ 7 parameter ของการทำ anchor box

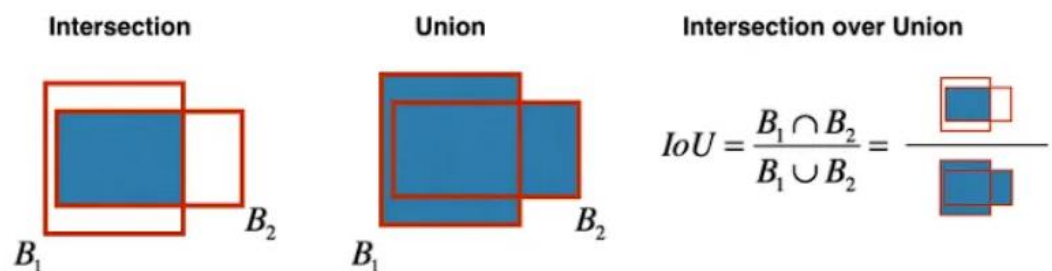
y	pc
	bx
	by
	bh
	bw
	c1
	c2
	pc
	bx
	by
	bh
	bw
	c1
	c2



ภาพประกอบที่ 3.20 การแทนค่าใน anchor box และภาพรวมของ anchor box

3.3.8 Intersection over Union

Intersection over Union คือ วิธีการวัดความเหมือนหมายถึงการเปรียบเทียบระหว่างกรอบสี่เหลี่ยมที่รอบตัวของวัตถุ (bounding box) ที่ระบุโดยโมเดลกับกรอบสี่เหลี่ยมที่ถูกต้อง (ground truth bounding box) โดยการคำนวณ IoU จะได้ผลลัพธ์ออกมาเป็นเลขระหว่าง 0 ถึง 1 โดยจะมีค่ามากเมื่อกรอบสี่เหลี่ยมที่ระบุโดยโมเดลมีการครอบคลุมวัตถุที่ถูกต้องมากขึ้น



ภาพประกอบที่ 3.21 IOU

ตัวอย่างการหาค่า IOU

1	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	1	1	0	0	0	0	1

0	0	1	1	0	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	0	0	0	0	0	0	0

1	1	1	1	0	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	1	1	0	0	0	0	1

ภาพประกอบที่ 3.22 ตัวอย่างการคำนวณค่า IoU

โดยภาพ P คือ ภาพ Matrix แรก และภาพ G คือ ภาพ Matrix ที่สอง และในภาพ P union G คือสีเทา ส่วน P intersection G คือสีเขียว สีเหลืองคือหายเกิน สีแดงคือหายขาด โดยสมการดังนี้

$$\begin{aligned}\text{union} &= 54 + 54 - 31 = 77 \\ \text{IOU} &= 31/77 = 0.4\end{aligned}$$



ภาพประกอบที่ 3.23 ตัวอย่างของ ค่า IoU ในรูปภาพ

3.3.9 Non-Max Suppression

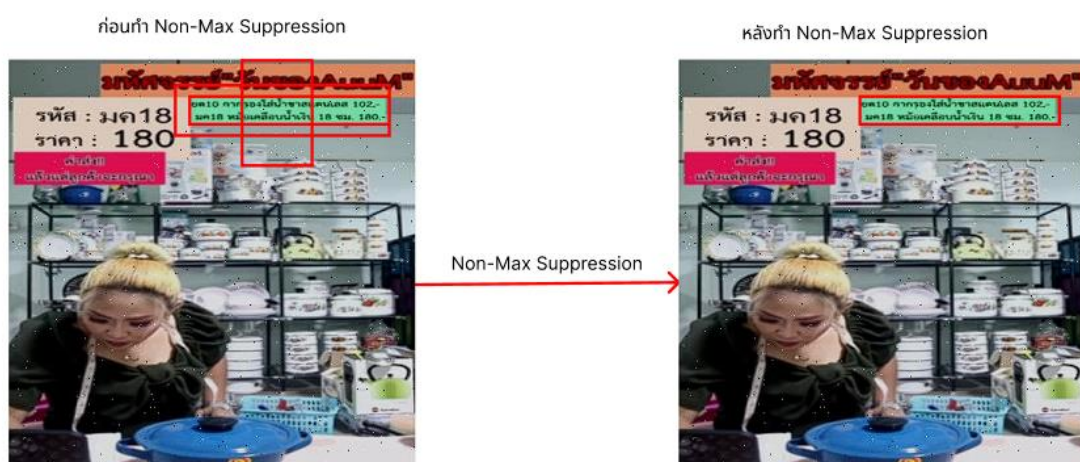
Non-Max Suppression เป็นเทคนิคหนึ่งที่ใช้ในการลดจำนวนของ bounding boxes หรือ detections ที่ซ้อนทับกันในการตรวจจับวัตถุ ซึ่งเป็นส่วนสำคัญในการประมวลผลภาพการฝึกฝน ข้อมูล วิธีการทำงานของ Non-Max Suppression คือ การเลือก bounding box ที่มีค่า confidence score สูงสุด และกำจัด bounding box อื่น ๆ ที่ซ้อนทับอยู่ในพื้นที่เดียวกัน โดยมีขั้นตอนดังนี้

- 1.จัดเรียง bounding box ตามค่า confidence score จากมากไปน้อย
- 2.เลือก bounding box ที่มีค่า confidence score สูงสุด และเก็บไว้
- 3.ลบ bounding box ที่มีค่า IoU หรือ threshold น้อยกว่าหรือเท่ากับค่าที่กำหนด
- 4.ทำขั้นตอนที่ 2 และ 3 จนกว่าจะไม่มี bounding box ที่เหลือให้ตรวจสอบ

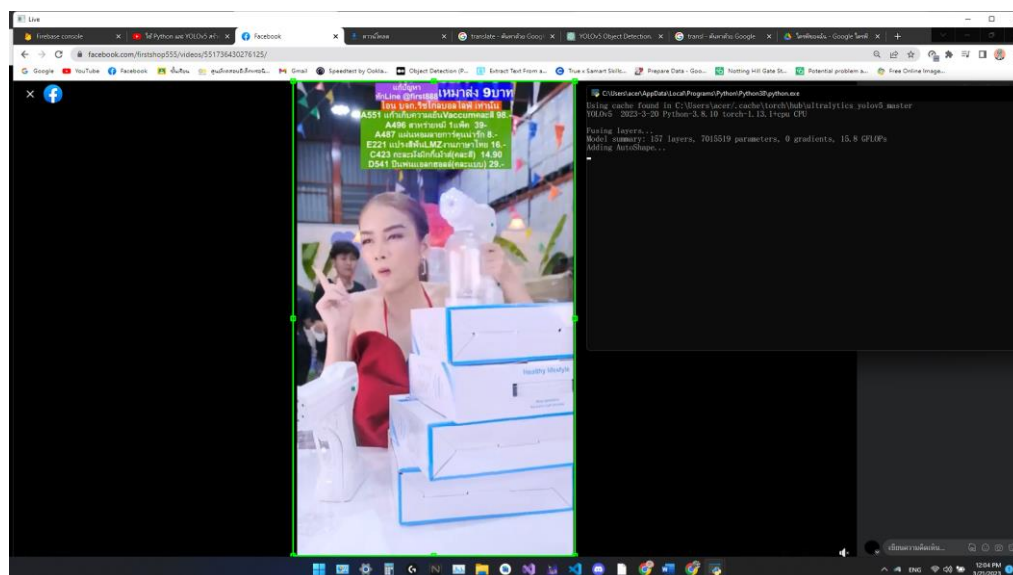
ตัวอย่างเมื่อกำหนดหนดค่า IoU ให้เท่ากับ 0.5

กล่องที่	confidence score		กล่องที่	confidence score
1	0.9	→	1	0.9
2	0.6			
3	0.75			

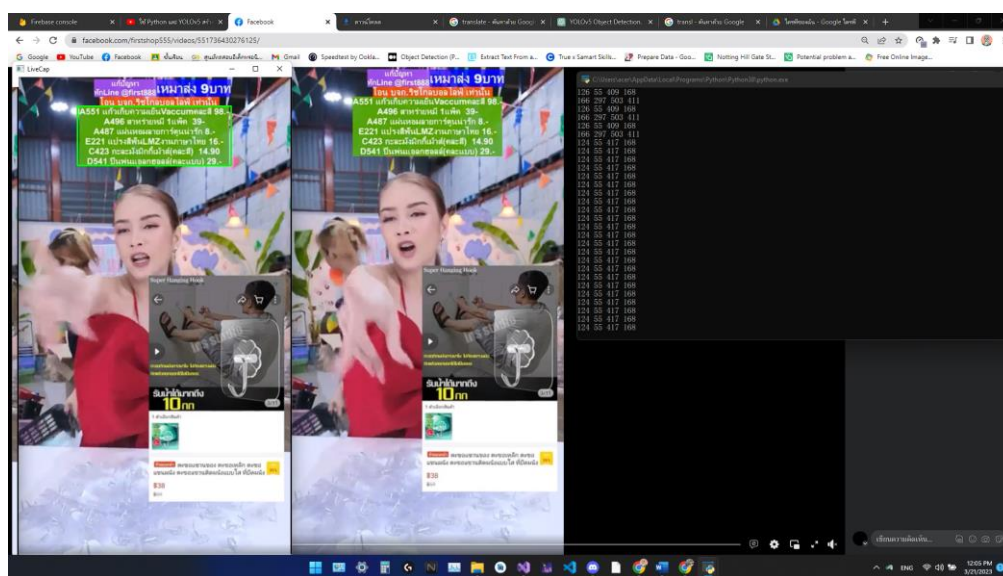
ภาพประกอบที่ 3.24 ตัวอย่างการทำ Non-Max Suppression



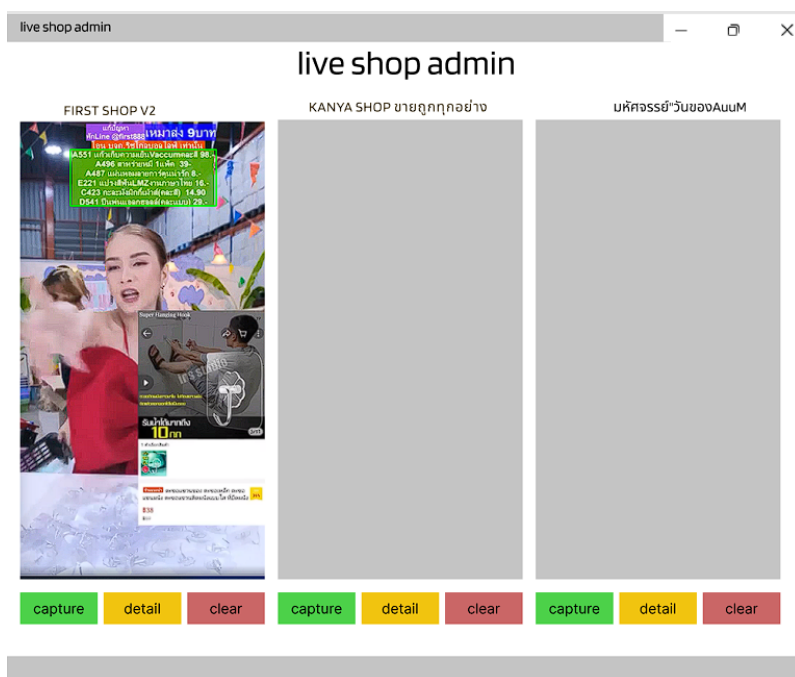
ภาพประกอบที่ 3.25 ตัวอย่างการทำ Non-Max Suppression



ภาพประกอบที่ 3.26 การทำงานในส่วนของการเลือกส่วนที่ต้องการ



ภาพประกอบที่ 3.27 ผลลัพธ์ในการ detect และการเลือกพื้นที่ที่ต้องการ

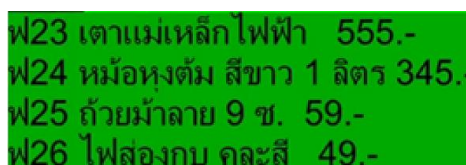


ภาพประกอบที่ 3.28 ผลลัพธ์ในการ detect ใน desktop application

3.4 การตัดบรรทัดด้วยเทคนิค projection profile

ในส่วนของงานในส่วนนี้จะเป็นการนำรูปภาพจากที่ทำการ detect มาได้มาทำการแปลงเป็นภาพ binary และทำการใช้เทคนิค horizontal projection profile มาทำการแบ่งบรรทัดเพื่อให้สินค้าแยกขึ้นกันอย่างชัดเจน

3.4.1 การเตรียมภาพ



ภาพประกอบที่ 3.29 ตัวอย่างภาพนำเข้า

การเตรียมพร้อมภาพ(Pre-Processing) เป็นขั้นตอนการนำเอาภาพเข้าเพื่อให้พร้อมต่อการนำไปประมวลผลต่อไป โดยประกอบด้วย ขั้นตอนการแปลงเป็นภาพระดับเทา และ การแปลงให้เป็นไบนารี ขาว-ดำ และตัดรูปภาพให้มีเฉพาะตรงส่วนที่มีข้อความอยู่ โดยขั้นตอนต่อไปนี้

3.4.2 แปลงภาพสี RGB ไปเป็นภาพระดับเทา

ขั้นตอนการแปลงภาพทำได้โดยอาศัยค่าของ RGB ที่อยู่ในแต่ละพิกเซลของภาพต้นฉบับมาแปลงเป็นภาพระดับเทา โดยช่องสี R จะมีค่าอยู่ที่ 0 – 255 ช่องสี G มีค่าอยู่ที่ 0-255 และช่องสี B จะมีค่าอยู่ที่ 0-255

		0		
	255	0	0	
		0		

		0		
	0	0	250	
		250		

		0		
	255	0	0	
		0		

ภาพประกอบที่ 3.30 ตารางค่าสี R G B ตามลำดับ

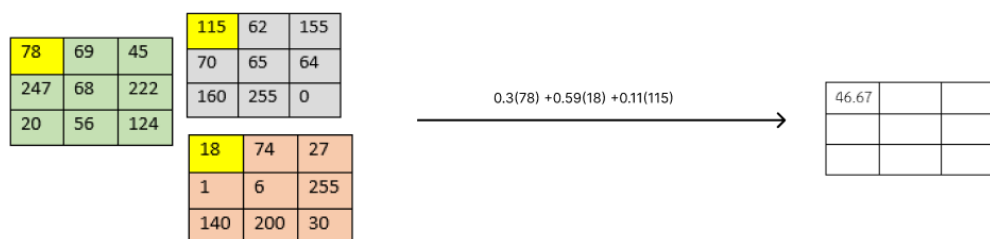
วิธีการนำพิกเซลตำแหน่งที่(3,4)จากตารางมาแปลงเป็นภาพระดับเทาสามารถทำได้ดังนี้

จากสูตร $\text{Gray} = 0.3R + 0.59G + 0.11B$

แทนค่า $\text{Gray} = 0.3(255) + 0.59(0) + 0.11(255)$

จะได้ค่า $\text{Gray} = 76.5 + 0 + 28.5 = 104.55$

โดยค่าคำตอบของ Gray จะถูกแทนลงในตำแหน่ง (3,4)ของภาพระดับเทา ตัวอย่างการแปลงภาพสี RGB เป็นภาพระดับเทา ของภาพที่มีขนาด 3x3 พิกเซล



ภาพประกอบที่ 3.31 การแปลงภาพสี RGB ไปเป็นระดับเทา

46.67	71.18	46.48
82.39	31.09	224.09
106.2	162.85	54.9

ภาพประกอบที่ 3.32 ตัวอย่างผลลัพธ์หลังจากการแปลงเป็นภาพระดับเทา

พ23 เต้าแม่เหล็กไฟฟ้า 555.-	พ23 เต้าแม่เหล็กไฟฟ้า 555.-
พ24 หม้อหุงต้ม สีขาว 1 ลิตร 345.-	พ24 หม้อหุงต้ม สีขาว 1 ลิตร 345.-
พ25 ถ้วยมัลลาล 9 ข. 59.-	พ25 ถ้วยมัลลาล 9 ข. 59.-
พ26 ไฟส่องกบ กละสี 49.-	พ26 ไฟส่องกบ กละสี 49.-

ภาพประกอบที่ 3.33 ตัวอย่างหลังจากการแปลงเป็นภาพระดับเทา

3.4.3 การแปลงภาพระดับเทา ไปเป็นภาพขาวดำ (Binary image)

เป็นการแปลงค่าสีจากระดับเทาที่มีค่าอยู่ระหว่าง 0 – 255 ให้เป็นภาพแบบไบนารีที่มีค่า 0 และ 255 โดยจะหาค่าขีดแบ่ง (Threshold) ที่เหมาะสม โดยใช้ Otsu จะคำนวณหาความแปรปรวนระหว่างกลุ่มเพื่อหาค่าขีดแบ่งที่ดีที่สุดในการแยกวัตถุออกจากพื้นหลังออกจากพื้นหลัง โดยพิจารณาจากค่า t ที่เป็นไปได้ ทั้งหมดโดย โดยมีวิธีการคำนวณดังนี้

ขั้นตอนที่ 1 กำหนดค่า L ให้เท่ากับ 4 และ $n_i = \{29, 21, 37, 18\}$

ขั้นตอนที่ 2 คำนวณหาค่าความน่าจะเป็น ดังสมการ

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=0}^{L-1} p_i = 1 \quad (3.2)$$

โดยที่ p_i คือ จำนวนจุดภาพที่ระดับความเข้มที่ i

N_i คือจำนวนพิกเซล ณ ระดับที่ i

N คือผลรวมของพิกเซลทั้งหมด ในแต่ละระดับ

L คือ ระดับความเข้มของภาพระดับสีเทา

- คำนวณความน่าจะเป็น p_i

$$p_0 = \frac{29}{105} = 0.2761$$

$$p_1 = \frac{21}{105} = 0.2$$

$$p_2 = \frac{37}{105} = 0.3524$$

$$p_3 = \frac{18}{105} = 0.1714$$

ขั้นตอนที่ 3 คำนวณหาค่าน้ำหนัก ω_0 และ ω_1 โดยค่าน้ำหนักคำนวณได้ตามสมการดังนี้

$$\omega_0 = \sum_{i=1}^K P_i = \omega(k) \quad (3.3)$$

โดยที่ ω_0 คือ ความน่าจะเป็นของกลุ่มที่ 1

$$k = 1$$

$$p = 0.2761$$

$$\omega_0 = \sum_{i=1}^K P_i = 0.2761$$

$$\omega_1 = \sum_{i=1}^L P_i \quad (3.4)$$

โดยที่ ω_1 คือ ความน่าจะเป็นของกลุ่มที่ 2

$$k = 1$$

$$p = \{0.2 + 0.3524 + 0.1714\}$$

$$\omega_1 = \sum_{i=1}^k P_i = 0.7238$$

- คำนวณหาค่า ω_0 และ ω_1

$$\omega_0 = \{0.2761, 0.3524, 0.1714\}$$

$$\omega_1 = \{0.7238, 0.3524, 0.1714\}$$

ขั้นตอนที่ 4 คำนวณหาค่าเฉลี่ย ดังสมการ

$$\mu_0 = \frac{\mu(k)}{\omega(k)} \quad (3.5)$$

โดยที่ μ_0 คือ ค่าเฉลี่ยของกลุ่มที่ 1

$\mu_k \sum_{i=1}^k i * p_i$ คือผลรวมของ $i * p_i$ ตั้งแต่ $i = k + 1$ ถึง L

$$\mu_1 = \sum_{i=k+1}^L \frac{i * p_i}{\omega_1} \quad (3.6)$$

โดยที่ μ_1 คือ ค่าเฉลี่ยของกลุ่มที่ 2

$\mu_1 = \sum_{i=k+1}^L \frac{i * p_i}{\omega_1}$ คือผลรวมของ $\frac{i * p_i}{\omega_1}$ ตั้งแต่ $i = k + 1$ ถึง L

- คำนวณหาค่า μ_0 μ_1 และ μ_k แทนค่าในสูตร กำหนด $k = 3$

$$\mu_k = (1 * 0.2761) = 0.2761$$

$$\mu_0 = (1 * 0.2761) + (2 * 0.2) + (3 * 0.3524) = 1.7333$$

$$\mu_1 = \left(\frac{2 * 0.2}{0.7238} \right) + \left(\frac{3 * 0.3524}{0.7238} \right) + \left(\frac{4 * 0.1714}{0.7238} \right) = 2.9604$$

- คำนวณหาค่า μ_0 μ_1 และ μ_k ทุกตำแหน่ง กำหนด $k = 1$

$$u_k = \{0.2761, 0.4, 1.0572\}$$

$$u_0 = \{1, 0.8402, 1.2760\}$$

$$u_1 = \{0.5526, 2.0132, 2.9604\}$$

ขั้นตอนที่ 5 คำนวณหาค่าความแปรปรวน ดังสมการ

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_0 - \mu_1)^2 \quad (3.7)$$

โดยที่ σ_B^2 คือค่าความแปรปรวนของกลุ่มตั้งแต่รอบที่ 1-255

$$u_1 = \sum_{i=k+1}^L \frac{i * p_i}{\omega_1} \quad \text{คือผลรวมของ } \frac{i * p_i}{\omega_1} \quad \text{ตั้งแต่ } i = k + 1 \text{ ถึง } L$$

- คำนวณหาค่า σ_B^2 จะได้

$$\sigma_B^2 = \{0.04, 0.6545, 2.3492\}$$

ขั้นตอนที่ 6 คำนวณหาค่าความแปรปรวนสูงสุด ดังสมการ

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \quad (3.8)$$

โดยที่ $\sigma_B^2(k^*)$ คือ ค่าความแปรปรวนสูงสุดของกลุ่มตั้งแต่รอบที่ $k = 1-255$

- คำนวณ หาค่า $\sigma_B^2(k^*)$ จะได้

$$\sigma_B^2(k^*) = 2.3492$$

โดยจะเลือกค่า k ที่ให้ค่าความแปรปรวนสูงสุด คือ 2 ที่เท่ากับ 2.3492 กำหนดให้เป็นค่า Threshold จะได้ $t=2$ จะได้ผลลัพธ์เป็นภาพ binary



ภาพประกอบที่ 3.34 ตัวอย่างการแปลงภาพระดับ เทา เป็นภาพ Binary

3.4.4 การวาดเส้นในจุดที่เป็นที่ว่าง

ในขั้นตอนนี้จะทำการวาดเส้นในส่วนที่เป็นช่องว่างระหว่างข้อความเพื่อแบ่งสินค้าออกเป็นแต่ละชิ้นโดยทำการหาพิกัด (x,y) เริ่มต้นของภาพและไล่เช็คไปที่ละพิกเซลจากซ้ายไปขวาและเมื่อเจอจุดที่เป็นช่องว่างจะนำมาทำการวาดเส้นแบ่งไว้โดยมีรายละเอียดและขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 เริ่มสแกนรูปภาพจากซ้ายไปขวาจนกว่าจะพบพิกเซลวัตถุ โดยหาค่าพิกเซลที่พบมีค่ามากกว่าหรือเท่า 1 แปลว่าพิกเซลที่พบมีข้อความอยู่

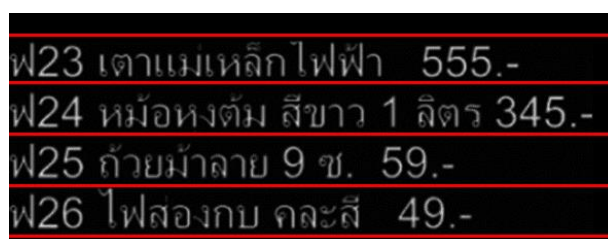
0	0	0	0	0
1	0	0	0	0
1	1	1	1	1
1	0	0	1	0
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0

ภาพประกอบที่ 3.35 ตัวอย่างสแกนรูปภาพจากซ้ายไปขวา

ขั้นตอนที่ 2 หลังจากได้จุดที่น่าจะเป็นขอบข้อความแล้วจะทำตามมาร์คเส้น ใน index ที่ว่างแล้วใส่เลข 5 ลงไปแทน แล้วเริ่มสแกนไปยังแถวถัดไป โดยในตำแหน่งที่เป็นเลข 5 จะกำหนดให้เป็นสีแดง และจะทำการวาดเส้นในตำแหน่งที่เป็นเลข 5 เพื่อทำการระบุจุดตรงนี้คือช่องว่างระหว่างบรรทัด

0	0	0	0	0
1	0	0	0	0
1	1	1	1	1
1	0	0	1	0
5	5	5	5	5
1	1	1	1	1
5	5	5	5	5

ภาพประกอบที่ 3.36 ตัวอย่างการมาร์คเส้น



ภาพประกอบที่ 3.37 ผลลัพธ์ในการวางเส้นแยกบรรทัด

3.5 ขั้นตอนการเรียกใช้ Tesseract OCR

ในกระบวนการ Optical Character Recognition (OCR) นั้นจำเป็นจะต้องมีข้อมูลลักษณะ (Feature) ของตัวอักษรนั้นๆ ก่อน เพื่อนำมาประมวลผลเทียบเคียงกับข้อมูลที่ได้จากภาพ ข้อมูล Feature ที่ได้มาจากการ (Train) ซึ่งค่อนข้างมีความซับซ้อนในการพัฒนา ดังนั้นในโครงการนี้จะใช้เครื่องมือที่ช่วยลดความยุ่งยาก ที่มีชื่อว่า Tesseract OCR เพื่อนำมาเปรียบเทียบผลลัพธ์ที่ได้

วิธีการดำเนินการ การสกัดตัวหนังสือภาษาไทยและภาษาอังกฤษ โดยขั้นตอนแรกนำภาพเข้าไป จากนั้นเขียน Code เพื่อทดสอบการทำงานและตรวจสอบผลลัพธ์

ตารางที่ 8 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)
```

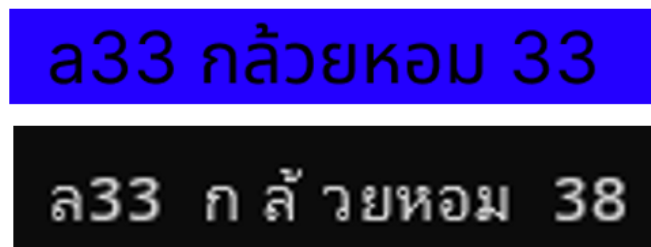
บรรทัดที่ 1 import flies tesserract

บรรทัดที่ 2 import image เพื่อใช้ในการเพิ่มรูปภาพเข้ามา

บรรทัดที่ 3 คือการอ่าน path files ของ tesserract

บรรทัดที่ 4 การอ่าน files รูปภาพแล้วกับไว้ในตัวแปล image เพื่อนำไปประมวลผล

บรรทัดที่ 5 การเอารูปภาพมาแปลงเป็น text จะแปลงเป็นภาษาไทยและภาษาอังกฤษ

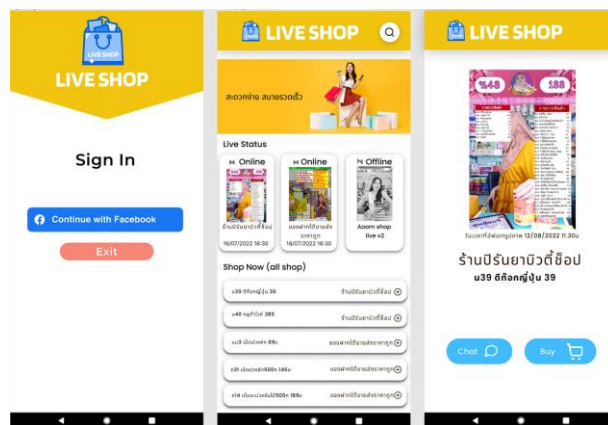


ภาพประกอบที่ 3.38 ตัวอย่างของผลลัพธ์ของการแปลงที่ผิดพลาด

การใช้ Training Dataset ในการสกัดตัวอักษรไทยและอังกฤษจากภาพ จะพบว่าสามารถสกัดข้อความออกมาได้ถูกบางคำ เพื่อป้องกันข้อผิดพลาดโครงการนี้จึงจะส่งรูปภาพให้ผู้ช่วย เพื่อลดข้อผิดพลาดลง

3.6 Mobile application

ในส่วนของ mobile application จะใช้ Flutter ในการสร้าง application ที่ทำการล็อกอินด้วย facebook ได้และเชื่อมต่อกับ firebase และดึงข้อมูลจาก filebase มาแสดงผลได้และเพื่อลดความผิดพลาดในการแสดงผลข้อมูลจะมีรูปภาพของร้านค้าในขนาดที่ทำการไลฟ์สดขึ้นโชว์ในหน้ารายละเอียดสินค้าด้วย

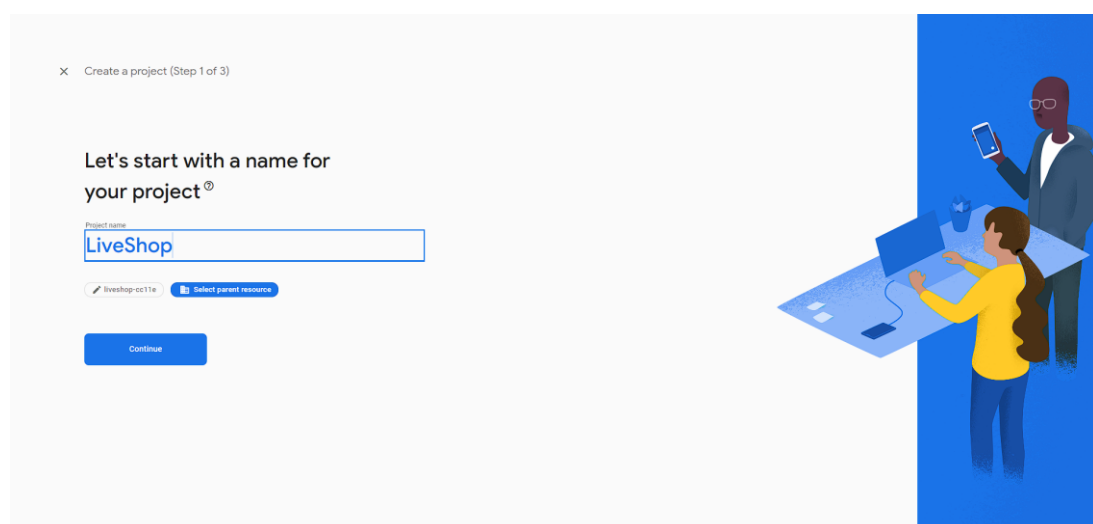


ภาพประกอบที่ 3.39 ตัวอย่างหน้าตา UI mobile application

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 3.40 การสร้างโปรเจ็ค firebase

ภาพประกอบที่ 3.41 การสร้าง Document ใน firebase

โดยข้อมูลทำการส่งไปเก็บในฐานข้อมูลจะประกอบไปด้วย

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

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

ภาพประกอบที่ 3.42 ตัวอย่างข้อมูลและประเภทของข้อมูล

3.8 วัดประสิทธิภาพ

1.การวัดประสิทธิภาพการ Detect วัดด้วย Mean Average Precision (mAP)

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

Confusion Matrix ถือเป็นเครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย หรือ Prediction ที่ทำนายจาก Model ที่เราสร้างขึ้น ใน Machine learning โดยมีไอดีจากการวัดว่า สิ่งที่เราคิด (Model ทำนาย) กับ สิ่งที่เกิดขึ้นจริง มีสัดส่วนเป็นอย่างไร

	Actually Positive (1)	Actually Negative (0)
Predicted Positive(1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

ภาพประกอบที่ 3.43 Confusion Matrix

True Positive (TP)=สิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้นจริงในกรณีทำนายว่าจริงและสิ่งที่เกิดขึ้นก็คือ จริง
 True Negative (TN)=สิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้น ในกรณีทำนายว่าไม่จริงและสิ่งที่เกิดขึ้นก็คือไม่จริง
 False Positive (FP)=สิ่งที่ทำนายไม่ตรงกับสิ่งที่เกิดขึ้น คือทำนายว่า จริง แต่สิ่งที่เกิดขึ้น คือ ไม่จริง
 False Negative (FN)=สิ่งที่ทำนายไม่ตรงกับที่ที่เกิดขึ้นจริง คือทำนายว่าไม่จริง แต่สิ่งที่เกิดขึ้น คือ จริง

Intersection over Union(IoU) เป็นวิธีทางสถิติที่ใช้วัดความสอดคล้องของข้อมูลสองชุด โดยมีข้อมูลสองเซตคือ P ซึ่งแทนเซตของพิกเซลในกรอบที่โมเดลเลือกมา และ G คือเซตของพิกเซลในกรอบที่เป็นเฉลย ใช้ P แทนคำว่า Predicted ส่วน G คือ Ground truth

$$IoU(P, G) = \frac{|P \cap G|}{|P \cup G|}$$

ภาพประกอบที่ 3.44 สมการหาค่า IoU

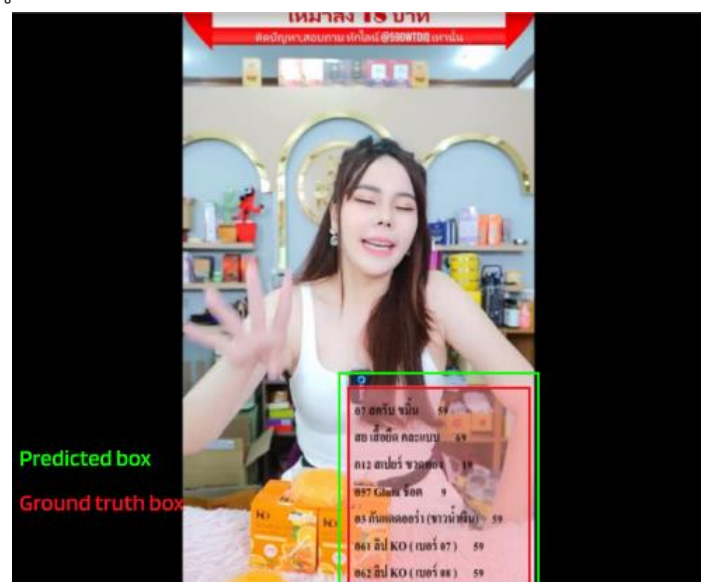
1	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	1	1	0	0	0	0	1

0	0	1	1	0	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	0	0	0	0	0	0	0

1	1	1	1	0	0	0	0	0
0	0	1	1	1	0	1	1	0
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
0	0	1	1	1	1	0	0	1
0	0	1	1	0	0	0	0	1

ภาพประกอบที่ 3.45 ตัวอย่างการหาค่า IoU

โดยภาพ P คือ ภาพ Matrix แรก และภาพ G คือ ภาพ Matrix ที่สอง และในภาพ P union G คือสีเทา ส่วน P intersection G คือสีเขียว สีเหลืองคือหายไป สีแดงคือหายขาดรวมทุกสี ในที่นี้ค่า $\text{IoU}(P, G) = 22/31$ จุดตัดบนยูเนียนบ่งชี้การซ้อนทับของฟิวด์กล่องขอบเขตที่คาดการณ์ไว้กับ Ground truth box IoU ที่สูงขึ้นบ่งชี้ว่าฟิวด์กล่องขอบเขตที่คาดการณ์ไว้ใกล้เคียงกับฟิวด์กล่องของผลเฉลย



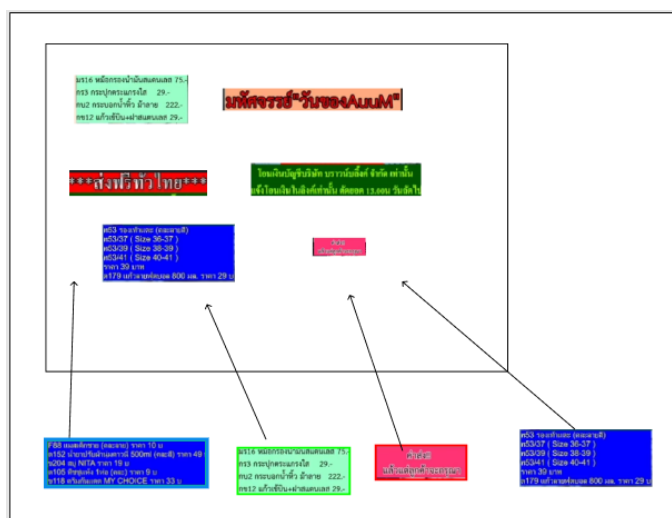
ภาพประกอบที่ 3.46 Predicted box และ Ground truth box

Precision คือ ค่าความแม่นยำจะวัดว่าสามารถค้นหาผลบวกที่แท้จริง (TP) จากการคาดการณ์เชิงบวกทั้งหมดได้ดีเพียงใด (TP+FP)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

ภาพประกอบที่ 3.47 การหาค่า Precision

Recall คือ จำนวนที่หายถูกต้องจำนวนของ GroundTruth ทั้งหมด ตัวอย่าง Model เราหามาเป็นลักษณะนี้ เราจะให้ Model เราหายุเฉพาะแค่ กรอบข้อความ หมายถึง เอาแต่ข้อความ มาให้เราเอาอย่างอื่นมาให้ถือว่าผิด

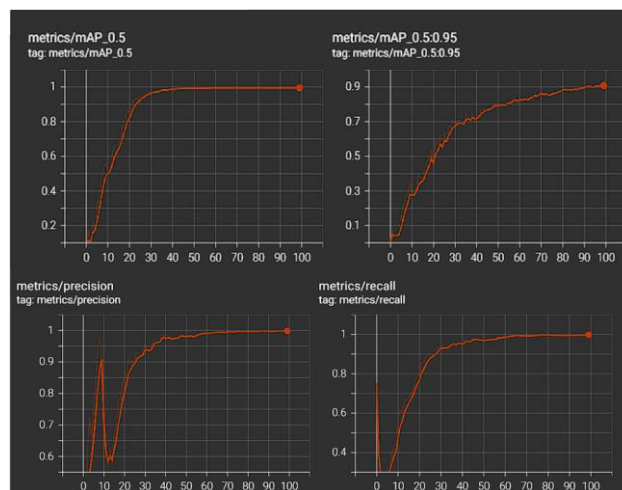


ภาพประกอบที่ 3.48 ตัวอย่างผลลัพธ์ในการทำนาย

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

ภาพประกอบที่ 3.49 การหาค่า recall

จาก model ของเราทำการเลือกภาพมาให้เราดังภาพที่อยู่นอกกรอบ จะเห็นว่า ได้รูปกรอบข้อความ ได้มา 3 รูป ส่วน สีเหลี่ยม รูป ลองคำนวณ Precision และ Recall ดูจากตัวอย่างจะได้ 3 TP และ 1 FP ดังนั้น precision คือ $3/4 = 0.75$ และ recall คือ $3/6 = 0.5$



ภาพประกอบที่ 3.50 กราฟการเปลี่ยนแปลงค่า mAP precision และ recall 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 3.51 สมการ CER

I (inserted words) คือ จำนวนตัวอักษรที่ถูกแทรกขึ้นมาจากข้อความเดิม

D (deleted words) คือ จำนวนตัวอักษรที่หายไปจากข้อความเดิม

S (substituted words) คือ จำนวนตัวอักษรที่ถูกแทนที่ไปจากคำเดิม

N คือ จำนวนตัวอักษรทั้งหมด

A037 เล็กนุ่มสกล้วยหอม 33.-	original words
A037 เล็กนุ่มสกล้วยหอม 33.-	deleted words
A037 เล็กนุ่มสกล้วยหอม 33.-	inserted words
A037 เล็กนุ่มสกล้วยหอม 33.-	substituted words

ภาพประกอบที่ 3.52 ตัวอย่างตัวแปรในสมการ CER

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

ภาพประกอบที่ 3.53 ตัวอย่างการหาค่า CER

จาก ภาพประกอบที่ 3.53 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. Kunakornlum, "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].