

ระบบการนับน้ำแข็งของแบบเรียลไทม์  
โดยใช้โครงข่ายประสาทเทียม  
REALTIME ICE COUNTING  
USING ARTIFICIAL NEURAL NETWORK



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

**REALTIME ICE COUNTING  
USING ARTIFICIAL NEURAL NETWORK**



**MR. SAHARAT RATTANAMOKA  
MR. SUWIJAK CHAIJAROENLAK**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING  
SCHOOL OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**ACADEMIC YEAR 2020**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.


คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบการนับน้ำแข็งของแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียม  
REALTIME ICE COUNTING USING ARTIFICIAL NEURAL  
NETWORK

นักศึกษา นายสหรัฐ รัตนโมคา รหัสประจำตัว 60011044  
นายสุวิจักขณ์ ชัยเจริญลักษณ์ รหัสประจำตัว 60011114

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

อาจารย์ผู้ควบคุมปริญญาานิพนธ์

  
(ผศ.ดร.อุตม จันท์จรีสสุข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

หัวข้อปริญญานิพนธ์	ระบบการนับน้ำแข็งของแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียม
นักศึกษา	นายสหรัฐ รัตนโมคา นายสุวิจักขณ์ ชัยเจริญลักษณ์
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2563
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ผศ.ดร. อุดม จันทร์จรัสสุข

### บทคัดย่อ

โรงงานน้ำแข็งทิพย์ธารา ใช้พนักงานในการบันทึกข้อมูลการส่งออกน้ำแข็งทำให้เกิดความผิดพลาดอยู่บ่อยครั้ง ทางโรงงานจึงต้องมีการตรวจนับน้ำแข็งย้อนหลังผ่านวิดีโอจากกล้องวงจรปิดย้อนหลังที่ถูกบันทึกไว้ของทุกวัน โดยใช้พนักงานตรวจสอบ 2 คน ซึ่งเป็นการทำงานที่ไม่ก่อให้เกิดคุณค่า ดังนั้น ผู้วิจัยจึงเสนอแนวทางในการใช้ระบบการตรวจนับน้ำแข็งอัตโนมัติแบบเรียลไทม์ โดยใช้เทคโนโลยีการตรวจจับวัตถุที่เข้ากับกล้องวงจรปิดเพื่อตรวจจับน้ำแข็งที่อยู่ในวิดีโอเป็นตัวตั้งต้นในการนับน้ำแข็ง ผู้วิจัยได้ใช้เทคนิค Transfer Learning ในการนำแบบจำลองการตรวจจับที่ผ่านการฝึกฝนมาแล้วล่วงหน้าชื่อ EfficientDet-D0 มาใช้เพื่อต่อยอดการเรียนรู้กับชุดข้อมูลที่ทางผู้วิจัยได้จัดเตรียมไว้ผ่านโปรแกรม Tensorflow โดยระบบการตรวจจับแบบเรียลไทม์จะทำการนับน้ำแข็งของที่ผ่านรางลำเลียงน้ำแข็งแต่ละก้อนที่ถูกลบและบันทึกเป็นข้อมูลเชิงตัวเลขพร้อมกับระบุวันที่และเวลา เพื่อเป็นฐานข้อมูลแบบเรียลไทม์ผ่าน Firebase จากผลการทดสอบระบบการตรวจนับมีค่าความแม่นยำอยู่ที่ 95.39%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

ก

Forbidden to modify the content, and cite the document when use.

<b>Thesis Title</b>	Realtime Ice Counting Using Artificial Neural Network
<b>Student</b>	Mr. Saharat Rattanamoka Mr. Suwijak Chaijaroenlak
<b>Degree</b>	Bachelor of Engineering in Industrial Engineering King Mongkut's Institute of Technology Ladkrabang
<b>Academic Year</b>	2020
<b>Thesis Advisor</b>	Asst.Prof.Dr. Udom Janjarassuk

### ABSTRACT

Thiphara Ice factory uses worker to record the number of ices shipped to customers, resulting in mismatch of the data record. Therefore, the factory has to recounting the number of ices from CCTV playback by using two inspectors, which is a non value added activity. Therefore, the researcher has proposed a way to build a real-time ice counting system by using object detection technology to count the number of ices from the CCTV video stream. The researchers used transfer learning techniques to implement a pre-trained detection model called EfficientDet-D0 to extend the learning with the data set provided by researcher by using Tensorflow.

The real-time ice counting system will count the number of ices that has passed through the conveyor. Each counted ice will be recorded as numerical data with the date and time through Firebase. The tested result has an average accuracy of 95.39%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

๗

Forbidden to modify the content, and cite the document when use.

## กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง ระบบการนับน้ำแข็งของแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียม  
ได้ประสบผลสำเร็จ ด้วยความช่วยเหลือจากทุกท่านที่เกี่ยวข้อง

ผศ.ดร. อุดม จันทร์จรัสสุข กรรมการและอาจารย์ที่ปรึกษาปริญญานิพนธ์ ผู้วิจัยขอขอบพระคุณ  
เป็นอย่างสูงสำหรับการชี้แนะแนวทางการศึกษา การประสานงาน การแก้ไขปัญหา และความช่วยเหลือ  
ในทุกๆ ด้านตลอดปีการศึกษาที่ผ่านมา

ผู้วิจัยขอขอบคุณคณาจารย์ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ สถาบัน  
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ความรู้ในสาขาวิชาอันเป็นส่วนสำคัญที่ทำให้  
ปริญญานิพนธ์เรียบร้อยสมบูรณ์

สุดท้ายนี้ ผู้วิจัยขอกราบขอบพระคุณบิดา มารดา ตลอดจนกัลยาณมิตรทุกท่าน ที่ให้กำลังใจและ  
การสนับสนุนเสมอมา อันทำให้ปริญญานิพนธ์สำเร็จไปด้วยดี

นายสหรัฐ รัตนโมคา

นายสุวิจักขณ์ ชัยเจริญลักษณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

ค

Forbidden to modify the content, and cite the document when use.

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ก
บทคัดย่อภาษาอังกฤษ .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ .....	ง
สารบัญตาราง .....	ฉ
สารบัญรูป .....	ช

### บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญ .....	1
1.2 วัตถุประสงค์ .....	2
1.3 ขอบเขตของปริญญาานิพนธ์ .....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ .....	2
1.5 ขั้นตอนดำเนินงานวิจัย .....	3

### บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 โครงข่ายประสาทเทียม .....	4
2.2.1 โครงข่ายประสาทแบบคอนโวลูชัน .....	5
2.2.2 คอมพิวเตอร์วิทัศน์ .....	7
2.2 การตรวจจับวัตถุ .....	8
2.2.1 ขอบข่ายการทำงานทั่วไปของการตรวจจับวัตถุ .....	9
2.2.2 การตรวจจับวัตถุโดยใช้ Tensorflow Object Detection API .....	10
2.2.3 Transfer Learning .....	10
2.2.4 Pre-Trained Model ที่ใช้ในงานวิจัย .....	12
2.3 ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนา .....	14
2.3.1 ภาษา Python .....	14

2.3.2 ไลบรารีที่ใช้ในการพัฒนา .....

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ในบริบทของการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

## สารบัญ

	หน้า
<b>บทที่ 3 วิธีการดำเนินงาน</b>	
3.1 การเตรียมชุดข้อมูล.....	17
3.1.1 การรวบรวมข้อมูล (Data Collecting) .....	17
3.1.2 การทำความสะอาดข้อมูล (Cleaning Data).....	18
3.1.3 การกำกับชื่อรูป (Image Anotation).....	18
3.1.4 การสร้างไฟล์ Comma Separated Value.....	20
3.1.5 การสร้างไฟล์ Tensorflow Record.....	20
3.1.6 การสร้าง Label Map.....	20
3.2 การสร้างระบบการตรวจจับน้ำแข็ง .....	21
3.2.1 การเลือกใช้ Pre-Trained Model.....	21
3.2.2 การปรับค่าพารามิเตอร์.....	21
3.2.3 การฝึกฝนแบบจำลอง.....	22
3.2.4 การทดลองการใช้งานของแบบจำลองจากวิดีโอตัวอย่าง.....	22
3.3 การสร้างระบบการตรวจจับน้ำแข็งแบบเรียลไทม์.....	23
3.3.1 การกำหนดจุดอ้างอิงของน้ำแข็ง .....	23
3.3.2 การกำหนดบริเวณอ้างอิงสำหรับการนับ .....	24
3.3.3 การเขียนภาษา Python เพื่อตรวจนับแบบเรียลไทม์ .....	25
<b>บทที่ 4 ผลการดำเนินงาน</b>	
4.1 ผลการดำเนินการจากการนับ.....	26
4.2 การแสดงผลแบบเรียลไทม์โดยใช้ Firebase.....	30
<b>บทที่ 5 สรุปและอภิปรายผล</b>	
5.1 การสรุปผลการวิจัย.....	31
5.2 ข้อเสนอแนะ .....	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่าการเอกสารอ้างอิง กิ่งหน้กนเรให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่ใช้

This material is reserved for educational use only, not allowed for commercial use.

จ

Forbidden to modify the content, and cite the document when use.

## สารบัญตาราง

	หน้า
ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงานวิจัย .....	3
ตารางที่ 3.1 การปรับค่าพารามิเตอร์ .....	21
ตารางที่ 4.1 อัตราส่วนจำนวนน้ำแข็งที่ระบบนับได้ต่อจำนวนน้ำแข็งจริง.....	28
ตารางที่ 4.2 เปอร์เซ็นต์ความถูกต้องของการวัดผลการนับ โดยแบ่งตามช่วงเวลา.....	28
ตารางที่ 4.3 เปอร์เซ็นต์ความถูกต้องของการวัดผลการนับ โดยแบ่งตามมุมกล้อง .....	28



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

ฉ

Forbidden to modify the content, and cite the document when use.

## สารบัญรูป

หน้า

รูปที่ 1.1 การลำเลียงก้อนน้ำแข็งผ่านรางลำเลียงลงสู่กระบะท้ายรถบรรทุก.....	2
รูปที่ 2.1 Neural Network.....	5
รูปที่ 2.2 แสดงบริเวณพื้นที่สนใจ (Receptive Field) ของเลขแปด.....	5
รูปที่ 2.3 ภาพรวมการทำงานของโครงข่ายประสาทแบบคอนโวลูชัน.....	6
รูปที่ 2.4 การแปลงรูปภาพให้อยู่ในรูปเมทริกซ์.....	6
รูปที่ 2.5 การ Max-Pooling ด้วยขนาด Kernel เท่ากับ 2.....	7
รูปที่ 2.6 การสร้าง Bounding Box ครอบคลุมเต็มภาพ.....	9
รูปที่ 2.7 การสกัดคุณลักษณะของแต่ละ Bounding Box.....	9
รูปที่ 2.8 การระบุชนิดของวัตถุและตำแหน่งของวัตถุในรูปภาพ.....	10
รูปที่ 2.9 แนวคิดสำหรับการทำ Transfer Learning.....	11
รูปที่ 2.10 ตัวอย่างการทำ Transfer Learning ด้วย Pre-Trained Model.....	12
รูปที่ 2.11 การเปรียบเทียบความเร็วในการฝึกฝนระหว่าง EfficientDet และแบบจำลองตัวอื่นบน GPU และ CPU.....	13
รูปที่ 2.12 ความแม่นยำของ EfficientDet ที่ทำการทดสอบกับ COCO เทียบกับแบบจำลองอื่น.....	13
รูปที่ 2.13 สถาปัตยกรรมของแบบจำลอง EfficientDet.....	14
รูปที่ 2.14 เอาท์พุทจากการทำนายคลาสและตำแหน่งของวัตถุในรูปภาพ.....	14
รูปที่ 2.15 การสร้างกรอบรอบวัตถุที่สนใจในการการตรวจจับวัตถุในภาพ.....	15
รูปที่ 3.1 เฟรมตัวอย่างของแต่ละมุกกล้องในช่วงเวลากลางวันและกลางคืน.....	17
รูปที่ 3.2 รูปภาพตัวอย่างรูปที่เลือกใช้และรูปที่คัดทิ้ง.....	18
รูปที่ 3.3 การ Annotate Image โดยใช้ LabelImg.....	19
รูปที่ 3.4 โครงสร้างของ XML ไฟล์.....	19
รูปที่ 3.5 ตัวอย่างไฟล์ CSV.....	20
รูปที่ 3.6 ตัวอย่างไฟล์ Label Map.....	20
รูปที่ 3.7 หลักการทำงานของระบบการตรวจจับวัตถุ.....	22
รูปที่ 3.8 ตัวอย่างการตรวจจับน้ำแข็งจากเฟรมต้นฉบับ.....	23
รูปที่ 3.9 ตัวอย่างพิกัดอ้างอิงของน้ำแข็ง.....	24

เอกสารนี้เป็นเอกสารที่ออกจากรายงานที่ออกการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

ช

Forbidden to modify the content, and cite the document when use.

## สารบัญรูป

	หน้า
รูปที่ 3.10 เฟรมที่มีการกำหนดบริเวณอ้างอิงทั้ง 4 มุมกล้อง.....	24
รูปที่ 3.11 การทำงานของระบบการตรวจนับแบบเรียลไทม์.....	25
รูปที่ 4.1 การตรวจจับก่อนน้ำแข็งขณะลำเลียงขึ้นรถ .....	26
รูปที่ 4.2 ภาพตัวอย่างการแสดงผลการทำงานของระบบการนับ.....	27
รูปที่ 4.3 การแสดงจำนวนน้ำแข็งที่นับสำเร็จบริเวณซ้ายบนของจอแสดงผล.....	28
รูปที่ 4.4 Firebase แสดงข้อมูลของทั้ง 4 มุมกล้อง.....	30
รูปที่ 4.5 ตัวอย่างการเก็บข้อมูลของก่อนน้ำแข็งที่ลำเลียงสำเร็จใน Firebase .....	30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

๗

Forbidden to modify the content, and cite the document when use.

# บทที่ 1

## บทนำ

ปริญญาานิพนธ์ระบบการนับน้ำแข็งของแบบเรียลไทม์โดยใช้โครงข่ายประสาทเทียม โดยจะกล่าวถึงรายละเอียดความเป็นมาและความสำคัญ วัตถุประสงค์ และขอบเขตของการจัดทำปริญญาานิพนธ์ โดยมีหัวข้อดังต่อไปนี้

- 1.1 ความเป็นมาและความสำคัญ
- 1.2 วัตถุประสงค์
- 1.3 ขอบเขตของปริญญาานิพนธ์
- 1.4 ประโยชน์ที่คาดว่าจะได้รับ
- 1.5 ขั้นตอนการดำเนินงาน

### 1.1 ความเป็นมาและความสำคัญ

โรงงานน้ำแข็งชองเป็นธุรกิจที่มีการผลิตและบริการขนส่งสินค้าไปยังผู้บริโภค ซึ่งกระบวนการจะเริ่มจากแผนกบริการจะรับคำสั่งซื้อจากลูกค้า หลังจากนั้นจะส่งข้อมูลไปที่ฝ่ายจัดส่งเพื่อส่งสินค้าให้ลูกค้า ซึ่งจะต้องมีการลำเลียงสินค้าขึ้นรถบรรทุก ดังรูปที่ 1.1 ตามจำนวนข้อมูลที่ถูกส่งมา ในระหว่างการลำเลียงต้องมีการนับจำนวนสินค้าที่ขนส่งขึ้นรถให้ตรงกับข้อมูลที่ได้รับคำสั่งซื้อ มา พร้อมจดบันทึกจำนวนของสินค้าที่บรรจุขึ้นรถในแต่ละคันก่อนจะขนส่งให้ลูกค้า โดยในแต่ละวันจะมีการตรวจบัญชีทำให้พบว่าจำนวนสินค้าที่ส่งออกและข้อมูลที่ทำกรบันทึกโดยพนักงานไม่ตรงกัน นั่นหมายความว่าปัญหาดังกล่าวเกิดขึ้นในระหว่างการทำงานและการบริหารงานภายในองค์กร โดยผู้ประกอบการตั้งข้อสันนิษฐานว่าสาเหตุเป็นเพราะความผิดพลาดของพนักงาน เช่น ลืมบันทึกข้อมูล นับจำนวนผิด และอาจจะเกิดการขอลงจากพนักงานในโรงงาน ที่ลักลอบนำก้อนน้ำแข็งที่ขาดหายไปในแต่ละวันไปขายให้กับลูกค้ารายอื่น และรับรายได้เข้าตนเองซึ่งเป็นปัญหาที่เกิดขึ้นทุกวัน ทำให้ทางบริษัทต้องจ้างพนักงานฝ่ายตรวจสอบจำนวน 2 คน เพื่อคอยตรวจสอบการทำงานผ่านกล้องวงจรปิดในทุกวัน ต้องคอยนับจำนวนสินค้าที่ถูกขนส่งขึ้นรถผ่านคลิปวิดีโอย้อนหลังเพื่อหาสาเหตุและทำข้อมูลให้ถูกต้อง

คณะผู้จัดทำได้เล็งเห็นถึงปัญหาดังกล่าวและคิดวิธีพัฒนาระบบการตรวจนับ โดยใช้องค์ความรู้ใน

เรื่องคอมพิวเตอร์วิทัศน์ (Computer Vision) ประยุกต์ใช้กับกล้องวงจรปิดที่มีการติดตั้งอยู่ในโรงงาน เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า บริเวณฐานลำเลียงสินค้าขึ้นทำรถบรรทุก ซึ่งระบบนี้จะช่วยประมวลผลภาพการทำงานจากกล้องวงจรปิด ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

ตรวจจับและนับจำนวนก้อนน้ำแข็งในแต่ละชานชลาที่มีการลำเลียง และบันทึกข้อมูลเชิงตัวเลขขึ้น Firebase จัดเก็บเป็นฐานข้อมูลที่สามารถตรวจสอบได้ตลอดเวลา



รูปที่ 1.1 การลำเลียงก้อนน้ำแข็งผ่านรางลำเลียงลงสู่กระบะท้ายรถบรรทุก

วิธีการดังกล่าวสามารถแก้ปัญหาที่ผู้ประกอบการไม่ทราบจำนวนของน้ำแข็งระหว่างการลำเลียงขึ้นรถบรรทุกว่าสูงขึ้นไปเป็นจำนวนเท่าใด ช่วยให้ฝ่ายตรวจสอบสามารถตรวจสอบจำนวนน้ำแข็งที่กำลังลำเลียงได้ทันที (Real Time) และตรวจสอบข้อมูลการทำงานได้ตลอดเวลาจากการจัดเก็บข้อมูลในเชิงตัวเลขและข้อมูลภาพการทำงานจากกล้องวงจรปิด

## 1.2 วัตถุประสงค์

สร้างระบบการตรวจนับจำนวนน้ำแข็งที่บรรจุขึ้นรถและบันทึกข้อมูลแบบเรียลไทม์

## 1.3 ขอบเขตของปริญญานิพนธ์

1. ดำเนินการเฉพาะกระบวนการลำเลียงและนับจำนวนน้ำแข็งของขึ้นรถเท่านั้น
2. ข้อมูลจากกล้อง CCTV 4 ตัวที่สถานีลำเลียงก้อนน้ำแข็งขึ้นรถ
3. ระบบจะทำการนับแค่น้ำแข็งของเท่านั้น
4. ระบบที่ใช้ถูกปรับแต่งจากแบบจำลองที่มีการฝึกสอนมาแล้ว (Pre-Trained Model)

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ประกอบการและฝ่ายตรวจสอบสามารถตรวจสอบจำนวนน้ำแข็งที่ถูกบรรจุขึ้นรถได้ตลอดเวลา
2. ช่วยให้ฝ่ายตรวจสอบปฏิบัติงานง่ายขึ้น
3. ลดการทำงานที่ไม่จำเป็น เช่น การตรวจสอบข้อมูลสินค้าย้อนหลังเนื่องจากข้อมูลไม่ตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

## 1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาสภาพปัจจุบันของโรงงาน วิเคราะห์ปัญหาที่เกิดขึ้นในโรงงาน และกำหนดแผนการทำงานในปีการศึกษา 2563
2. ติดตั้งเครื่องมือสำหรับการทำการตรวจจับวัตถุ (Object Detection) และดำเนินการขั้นตอนจัดเตรียมข้อมูล (Data Preparation)
3. สร้างแบบจำลอง (Model) และฝึกสอนแบบจำลอง (Training) เพื่อตรวจจับวัตถุด้วยชุดข้อมูลที่จัดเตรียมไว้
4. สร้างระบบตรวจนับจำนวนน้ำแข็งแบบเรียลไทม์ (Realtime Counting) โดยแบบจำลองที่ผ่านการฝึกสอนและเชื่อมต่อผ่าน Firebase
5. รวบรวมผลลัพธ์จากการประมวลผลเป็นวิดีโอและทดสอบความถูกต้องของการนับของระบบ
6. สรุปผลงานวิจัย เรียบเรียงข้อมูลและนำเสนอ

ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงานวิจัย

รายการ	พ.ศ. 2563					พ.ศ. 2564				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. วิเคราะห์ที่มาของปัญหาและกำหนดแผนดำเนินงาน										
2. ติดตั้งเครื่องมือที่ใช้ และดำเนินการขั้นจัดเตรียมข้อมูล										
3. สร้างแบบจำลองการตรวจนับน้ำแข็ง										
4. สร้างระบบตรวจนับจำนวนน้ำแข็งแบบเรียลไทม์										
5. รวบรวมผลลัพธ์และทดสอบการทำงานของระบบ										
6. สรุปและอภิปรายผล										
7. เรียบเรียงและนำเสนอ										

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

แนวคิดการศึกษาวิทยานิพนธ์ระบบการนับน้ำแข็งของแบบเรียลไทม์โดยประมวลผลจากโครงข่ายประสาทเทียม มีการศึกษาทฤษฎีและองค์ความรู้ที่เกี่ยวข้อง โดยแบ่งเป็นหัวข้อต่างๆ ดังนี้

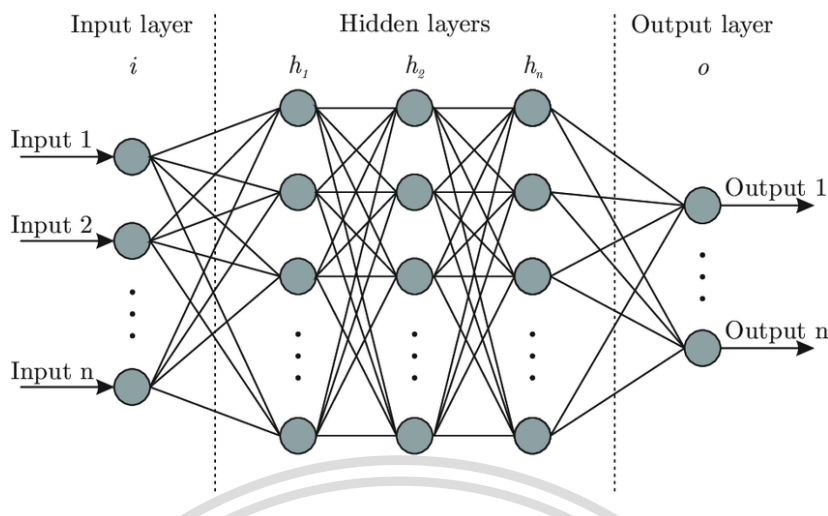
- 2.1 โครงข่ายประสาทเทียม
- 2.2 การตรวจจับวัตถุ
- 2.3 ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนา

#### 2.1 โครงข่ายประสาทเทียม (Artificial Neural Network)

โครงข่ายประสาทเทียม (Artificial Neural Network) เป็นแบบจำลองทางคณิตศาสตร์ที่จำลองกระบวนการคิดจากสมองมนุษย์ โดยสมองนั้นจะมีหน่วยประมวลผลขนาดเล็กอยู่จำนวนมาก และเชื่อมโยงกันด้วยโครงข่ายประสาทมากมายทำให้มนุษย์เกิดการเรียนรู้และคิดวิเคราะห์ได้อย่างรวดเร็ว เกิดการจำลองแล้วทางการเรียนรู้ของมนุษย์ไปในระบบคอมพิวเตอร์เพื่อให้เกิดการเรียนรู้ทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) ที่สามารถนำไปประยุกต์ใช้กับงานหลายด้านได้อย่างมีประสิทธิภาพ เช่น การจำแนกรูปแบบ การทำนาย การควบคุม การหาความเหมาะสม และการจัดกลุ่ม เป็นต้น มีหลักการในการพยายามลอกเลียนแบบการทำงานของเซลล์ประสาทในสมองมนุษย์ เพื่อทำงานได้อย่างมีประสิทธิภาพ ประกอบด้วยหน่วยประมวลผลย่อย (Node) จำนวนมากซึ่งรวมกันเป็นชั้น (Layer) มีการนำเข้าข้อมูลในชั้นแรก (Input Layer) และส่งออกข้อมูลที่ผ่านการประมวลผลในชั้นสุดท้าย (Output Layer) และชั้นที่อยู่ระหว่างการประมวลผลของชั้นแรกและชั้นสุดท้าย คือ ชั้นแอบแฝง (Hidden Layer) ซึ่งจำนวนของชั้นแอบแฝงจะขึ้นอยู่กับความซับซ้อนของการใช้งาน [1] ดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

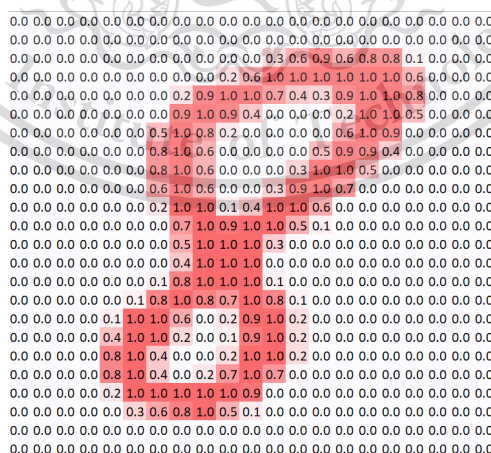
This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 2.1 Neural Network

### 2.1.1 โครงข่ายประสาทแบบคอนโวลูชัน (Convolutional Neural Network)

การวิเคราะห์รูปภาพที่มนุษย์มองเห็น โดยจะแบ่งรูปภาพออกเป็นพื้นที่ย่อยระดับพิกเซล (Pixel) เพื่อทำการวิเคราะห์เมทริกซ์ของรูปภาพ โดยถ้าเป็นรูปภาพสีขาวดำจะเป็นเมทริกซ์มิติ 2 และรูปภาพสีจะเป็นเมทริกซ์มิติ 3 เปรียบเทียบกับการทำงานของโครงข่ายประสาทเทียมทั่วไปที่เชื่อมทุกหน่วยประมวลผลย่อย (Node) ของชั้น (Layer) ก่อนหน้าเข้าด้วยกัน แต่โครงข่ายประสาทแบบคอนโวลูชันจะเลือกเชื่อมต่อแค่บริเวณพื้นที่สนใจ เรียกว่า Receptive Field [2] ในรูปที่ 2.2 เป็นการยกตัวอย่างการนำรูปภาพของเลขแปดมาแบ่งออกเป็นพื้นที่ย่อยระดับพิกเซล และเลือกข้อมูลเฉพาะพื้นที่สีแดงหรือพื้นที่สนใจ (Receptive Field) ก่อนจะดำเนินการไปยังชั้นประมวลผลถัดไป

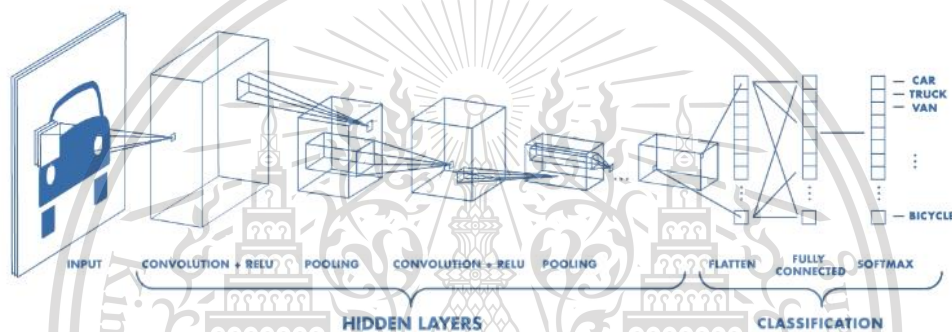


รูปที่ 2.2 แสดงบริเวณพื้นที่สนใจ (Receptive Field) ของเลขแปด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

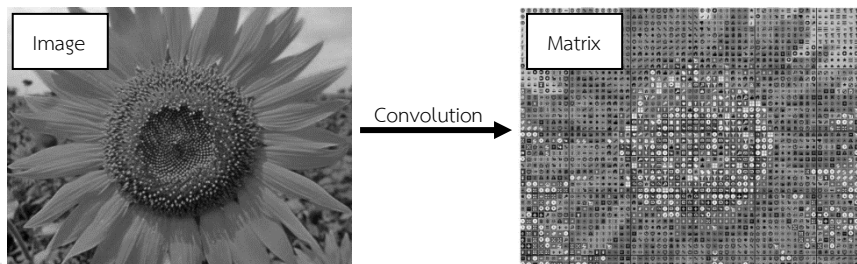
การใช้โครงข่ายประสาทแบบคอนโวลูชันประมวลผลรูปด้วย Filter ที่แตกต่างกัน จะให้ความหมายที่แตกต่างกันไป เช่น หาขอบรูป หาความเบลอ หาความคม และสามารถหาลักษณะจำเพาะของสิ่งที่สนใจได้ตั้งแต่ชั้น (Layer) ส่วนแรกและประมวลผลไปจนถึงชั้นส่วนปลาย ซึ่งจะมีความเป็นนามธรรม (Abstract) มากขึ้นตามลำดับ รูปที่ 2.3 ที่แสดงภาพรวมการทำงานทั้งหมด 4 ขั้นตอนของโครงข่ายประสาทแบบคอนโวลูชัน ได้แก่ Convolution or Filter, RELU, Pooling และ Flatten ใน 3 ขั้นตอนแรกเป็นการกรองข้อมูลทีสนใจจัดอยู่ในชั้นแอบแฝง (Hidden Layer) การกรองข้อมูลทีสนใจ และขั้นตอน Flatten จะเป็นการประมวลผลเพื่อจำแนกประเภทของวัตถุทีสนใจ การอธิบายขั้นตอนการทำงานของโครงข่ายประสาทแบบคอนโวลูชันทั้ง 4 ขั้นตอนมีดังนี้



รูปที่ 2.3 ภาพรวมการทำงานของโครงข่ายประสาทแบบคอนโวลูชัน

### 2.1.1.1 Convolution or Filter

การนำรูปภาพเข้ามาในกระบวนการโดยแบ่งรูปภาพดังกล่าวเป็นพิกเซล (Pixel) เช่น รูปภาพขาวดำจะได้เมทริกซ์ 2 มิติขนาด 50x50 พิกเซล โดยทั้งหมด 2500 พิกเซล จะมีการกำหนดขนาดตัวเลขหรือน้ำหนักของพิกเซลนั้นๆ ในรูปที่ 2.4 เป็นการยกตัวอย่างการนำรูปภาพของดอกทานตะวันจากรูปภาพเดิมมาแบ่งออกเป็นพื้นที่ย่อยระดับพิกเซลและค่าน้ำหนักของรูปเป็นเมทริกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 2.4 การแปลงรูปภาพให้อยู่ในรูปเมทริกซ์  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

### 2.1.1.2 RELU

ใช้ฟังก์ชัน  $RELU(x) = \max(0, x)$  ปรับค่าตัวเลขทั้งหมดในเมทริกซ์ แต่ละพิกเซลให้เป็นค่าจำนวนจริงบวกหรือ  $\max(0, x)$  จะมีผลกับเมทริกซ์ที่มีค่าติดลบ ซึ่งจะถูกรับค่าลบเป็นศูนย์ทั้งหมด เพื่อเป็นการกรองค่าติดลบ และเพิ่มความเร็วของการประมวลผลในขั้นถัดไป

### 2.1.1.3 Pooling

Pooling เป็นกระบวนการที่มีจุดประสงค์ในการลดขอบเขตเชิงพื้นที่ของเมทริกซ์ซึ่งจะช่วยลดพารามิเตอร์และการคำนวณโดยรวมของระบบ การดำเนินการ Max-Pooling ต้องการกำหนดขนาด Kernel size และความยาวก้าวระหว่างการออกแบบสถาปัตยกรรม และดำเนินการเลื่อนเคอร์เนลโดยมีขั้นตอนที่เลือกเฉพาะค่าที่มากที่สุดในแต่ละชั้นส่วน Kernel จากอินพุตที่กำหนดเพื่อให้ได้ค่าสำหรับเอาต์พุตที่มีขนาดเล็กลง กระบวนการนี้สามารถอธิบายในรูปแบบของเมทริกซ์ ดังแสดงในรูปที่ 2.5 การ Max-Pooling โดยมี Kernel Size เท่ากับ  $2 \times 2$  พิกเซลและเลือกค่าที่มากที่สุดของแต่ละเมทริกซ์ Kernel เพื่อลดขนาดรูปภาพขนาดภาพจาก  $4 \times 4$  เป็น  $2 \times 2$  พิกเซล



รูปที่ 2.5 การ Max-Pooling ด้วยขนาด Kernel เท่ากับ 2

### 2.1.1.4 Flatten

การแปลงค่าเมทริกซ์ 3 มิติที่ผ่านการ Pooling เป็นค่าเวกเตอร์มิติเดียว เช่น เมทริกซ์ขนาด  $5 \times 5 \times 2$  จะถูกแปลงเป็นเมทริกซ์ขนาด  $1 \times 1$  จำนวน 50 ค่า เพื่อใช้ในการจำแนกคุณสมบัติด้วยฟังก์ชัน Softmax ซึ่งต้องใช้อินพุต 1 มิติ ที่ได้จากการทำ Flatten Layer สามารถดูผลลัพธ์การจำแนกได้ที่ชั้น (Layer) นี้ได้ในขั้นตอนสุดท้ายของ รูปที่ 2.3

## 2.1.2 คอมพิวเตอร์วิทัศน์ (Computer Vision)

คอมพิวเตอร์วิทัศน์ (Computer Vision) เป็นสาขาหนึ่งของปัญญาประดิษฐ์ จุดประสงค์หลักของคอมพิวเตอร์วิทัศน์ คือ การทำให้คอมพิวเตอร์สามารถเข้าใจทิวทัศน์หรือคุณลักษณะต่างๆ ในภาพได้ การสอนคอมพิวเตอร์ให้มองเห็น แยกแยะวัตถุ และบอกความสัมพันธ์ของวัตถุ สามารถเปรียบเทียบกับ การเรียนรู้ในวัยเด็กของมนุษย์ที่สามารถแยกแยะสิ่งเหล่านี้ได้จากรู้จากการมองเห็นสิ่งต่างๆ ซึ่งไม่จำเป็นต้องสะสมประสบการณ์ นักวิทยาศาสตร์พบว่าช่วงชีวิตเด็กเกิดการเรียนรู้จากการมองเห็นซึ่ง

เปรียบเทียบได้กับรูปภาพถึง 200 ล้านรูป หากต้องการสอนให้คอมพิวเตอร์มองเห็นได้อย่างฉลาดและแม่นยำ จำเป็นต้องมอบประสบการณ์ให้คอมพิวเตอร์ด้วยจำนวนรูปภาพ หรือข้อมูลมหาศาล (Big Data) เช่นเดียวกับมนุษย์

การพัฒนาของคอมพิวเตอร์วิทัศน์นั้น ก็เปรียบกับการมอบดวงตาให้กับคอมพิวเตอร์ เพิ่มความสามารถของคอมพิวเตอร์ เช่น การพัฒนายานพาหนะไร้คนขับหรือหุ่นยนต์อัจฉริยะ หากถูกพัฒนาไปไกล มนุษย์อาจได้ใช้ประโยชน์จากคอมพิวเตอร์วิทัศน์ในวงการแพทย์ เช่น การแยกเซลล์มะเร็ง หรือแยกแยะเซลล์เม็ดเลือดร้ายแทนการมองเห็นของมนุษย์ ซึ่งเป็นการพัฒนาของเทคโนโลยีที่จะช่วยเพิ่มขีดความสามารถของมนุษย์ และอาจจะมีคอมพิวเตอร์ที่มีดวงตาดีกว่ามนุษย์หลายเท่า [3]

## 2.2 การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุเป็นความรู้สาขาหนึ่งในคอมพิวเตอร์วิทัศน์ (Computer Vision) ที่ระบบซอฟต์แวร์ช่วยให้เราสามารถระบุ ค้นหา และติดตามวัตถุจากภาพหรือวิดีโอที่กำหนด คุณลักษณะพิเศษเกี่ยวกับการตรวจจับวัตถุ คือ การระบุคลาสของวัตถุ เช่น บุคคล โต๊ะ รถยนต์ เป็นต้น และการกำหนดพิกัดของตำแหน่งเฉพาะในภาพ ตำแหน่งดังกล่าวถูกแสดงให้เห็นโดยการวาดกรอบล้อมรอบวัตถุ (Bounding Box) แสดงขอบเขตสามารถระบุตำแหน่งของวัตถุได้อย่างแม่นยำ ความสามารถในการค้นหาวัตถุภายในภาพขึ้นอยู่กับกำหนดประสิทธิภาพของอัลกอริทึมที่ใช้ในการตรวจจับ เมื่อมนุษย์ดูภาพหรือวิดีโอในปริมาณมากจะสามารถรับรู้และค้นหาวัตถุที่สนใจได้ภายในเวลาไม่นาน เป้าหมายของการตรวจจับวัตถุ คือ การจำลองการประมวลผลข้อมูลนี้โดยใช้คอมพิวเตอร์ [4]

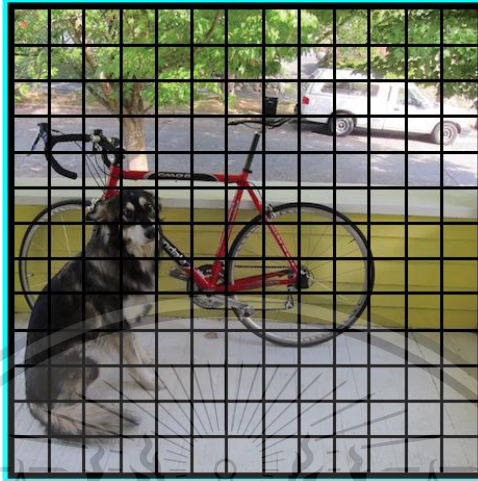
การตรวจจับวัตถุจะมีความคล้ายคลึงกับเทคนิคการมองเห็นด้วยคอมพิวเตอร์อื่นๆ กันเช่นการจดจำภาพ (Image Recognition) และการแบ่งส่วนภาพ (Image Segmentation) ซึ่งจะช่วยให้เราเข้าใจและสามารถวิเคราะห์ฉากในภาพหรือวิดีโอได้ แต่มีข้อแตกต่างที่สำคัญ คือ การจดจำภาพจะแสดงเฉพาะฉลากกำกับคลาสสำหรับวัตถุที่ระบุเท่านั้น และการแบ่งส่วนภาพจะสร้างความเข้าใจในระดับพิกเซลขององค์ประกอบของฉาก สิ่งที่แยกการตรวจจับวัตถุออกจากเทคนิคอื่น ๆ คือ ความสามารถพิเศษในการค้นหาวัตถุที่เราสนใจภายในภาพหรือวิดีโอ ซึ่งจะช่วยให้เราสามารถนับและติดตามวัตถุเหล่านั้นได้ ด้วยความแตกต่างที่สำคัญเหล่านี้และความสามารถเฉพาะของการตรวจจับวัตถุทำให้ปัจจุบันมีการนำไปประยุกต์ใช้มากมาย เช่น การตรวจจับผู้คนตามท้องถนน การแยกคุณภาพของสินค้า และการตรวจจับใบหน้า เป็นต้น ซึ่งทางผู้จัดทำก็ได้นำเทคนิคการตรวจจับวัตถุประยุกต์ใช้กับงานวิจัยนี้ในการนับก้อนน้ำแข็งของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

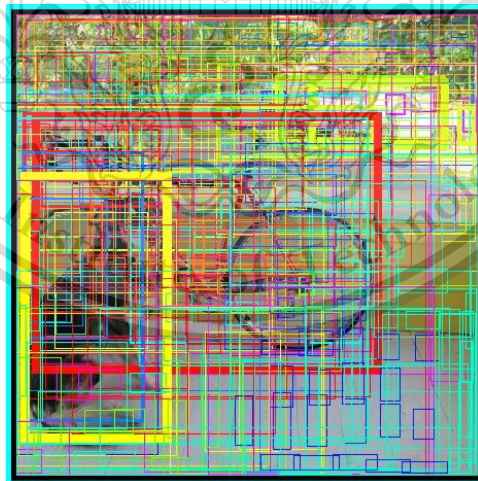
## 2.2.1 ขอบข่ายการทำงานทั่วไปของการตรวจจับวัตถุ

1. ใช้อัลกอริทึมเพื่อสร้าง Bounding Box ที่ครอบคลุมเต็มภาพ ซึ่งเป็นองค์ประกอบของการระบุตำแหน่งของวัตถุ ดังรูปที่ 2.6



รูปที่ 2.6 การสร้าง Bounding Box ครอบคลุมเต็มภาพ [5]

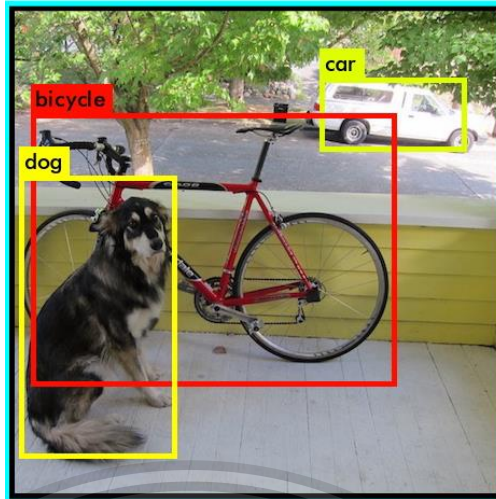
2. คุณลักษณะของรูปภาพ (Visual Features) จะถูกสกัดออกมาสำหรับแต่ละ Bounding Box จำนวนมากในภาพเดียว ดำเนินการประเมินและจะพิจารณาว่ามีวัตถุใดบ้างที่มีอยู่ในกล่องตามคุณลักษณะของรูปภาพ ดังรูปที่ 2.7



รูปที่ 2.7 การสกัดคุณลักษณะของแต่ละ Bounding Box [5]

3. กล่องที่ทับซ้อนกันจะรวมกันเป็น Bounding Box เดียวกันเพื่อแสดงชนิดของวัตถุและเอกสารนี้เต็มตำแหน่งของวัตถุในภาพ ดังรูปที่ 2.8 งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 2.8 การระบุชนิดของวัตถุและตำแหน่งของวัตถุในรูปภาพ [5]

## 2.2.2 การตรวจจับวัตถุโดยใช้ Tensorflow Object Detection API

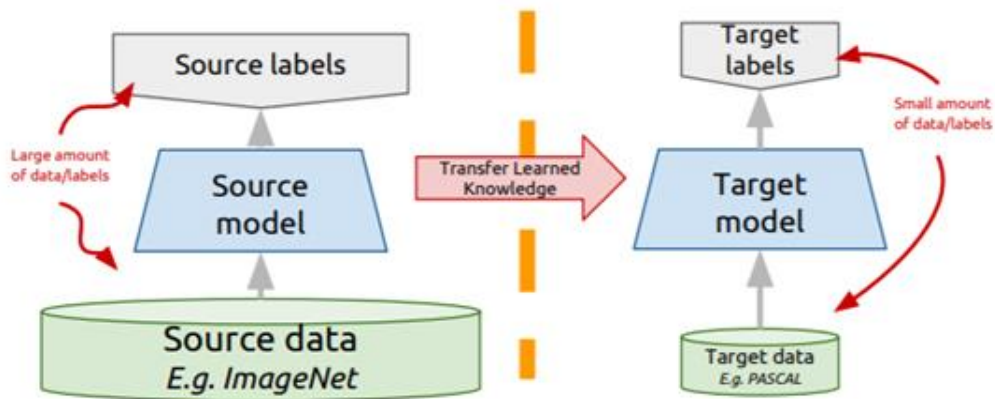
Tensorflow Object Detection API คือ Framework Open Source ที่สร้างขึ้นในไลบรารี Tensorflow ที่มีจุดมุ่งหมายเพื่อให้ง่ายต่อการสร้างฝึกฝนและปรับใช้การตรวจจับวัตถุแบบจำลอง [6] ส่วนมากการฝึกฝนโครงข่ายประสาทแบบคอนโวลูชันทั้งหมดตั้งแต่เริ่มต้นนั้น ต้องใช้เวลานานและชุดข้อมูลจำนวนมาก ปัญหานี้สามารถแก้ไขได้โดยใช้ข้อได้เปรียบของการทำ Transfer Learning ด้วยรูปแบบของ Pre-Trained Model โดยใช้ Tensorflow API

## 2.2.3 Transfer Learning

Transfer Learning [7] เป็นหลักการพื้นฐานที่ว่าแบบจำลองที่ได้จากการเรียนรู้ข้อมูลชุดหนึ่งสามารถนำไปใช้เป็นแบบจำลองตั้งต้นในการเรียนรู้ข้อมูลอีกชุดหนึ่งได้ ซึ่งเป็นเทคนิคที่ช่วยลดเวลาในการฝึกฝนแบบจำลองการเรียนรู้เชิงลึกด้วยการนำบางส่วนของแบบจำลองที่ทำการฝึกฝนเรียบร้อยแล้ว นำมาปรับใช้เป็นส่วนหนึ่งของแบบจำลองใหม่ในการฝึกฝนกับชุดข้อมูลชุดใหม่เพื่อนำไปใช้แก้ปัญหาอื่นที่มีความสอดคล้องกัน ในทางปฏิบัติแล้วมีคนจำนวนน้อยมากที่ฝึกฝน โครงข่ายประสาทแบบคอนโวลูชัน (Convulational Neural Network) ตั้งแต่นั้น เนื่องจากไม่มีชุดข้อมูลที่ใหญ่เพียงพอ ดังนั้นคนส่วนใหญ่ใช้วิธีนำแบบจำลองที่ผ่านการฝึกฝนกับชุดข้อมูลขนาดใหญ่ เช่น ImageNet ที่มีข้อมูลตัวอย่างจำนวน 1.5 ล้านรูป โดยเราจะนำแบบจำลองนั้นมาเป็นแบบจำลองตั้งต้นเพื่อฝึกฝนต่อกับชุดข้อมูลขนาดเล็กในงานเฉพาะทางหรือใช้การสกัดคุณลักษณะ (Feature Extractor) สำหรับงานที่ต้องการขนถ่ายข้อมูลด้วยแนวคิดของการทำ Transfer Learning เป็นดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 2.9 แนวคิดสำหรับการทำ Transfer Learning [7]

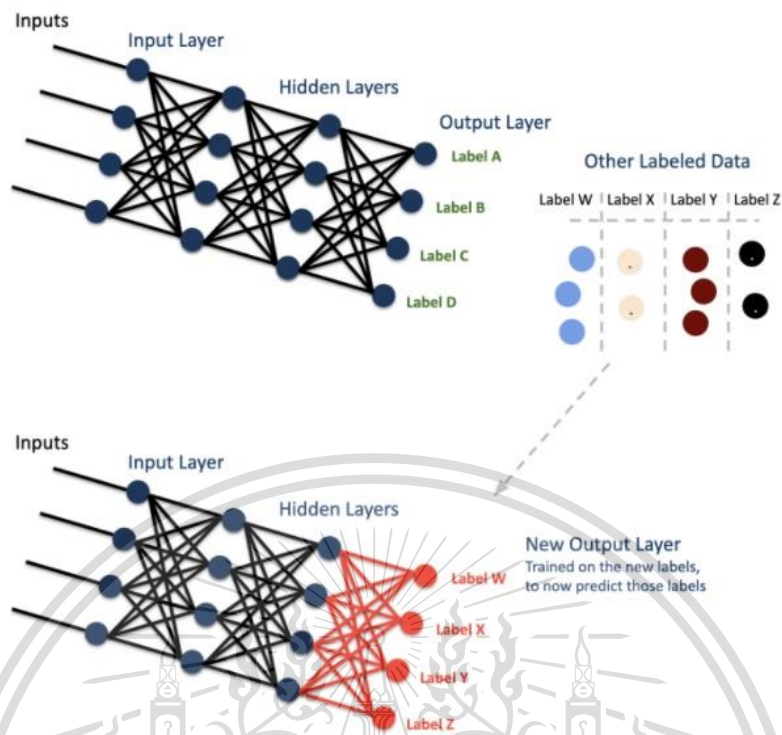
การทำ Transfer Learning มี 3 วิธีหลัก ได้แก่

1. ใช้โครงข่ายประสาทแบบคอนโวลูชันเป็น Fixed Feature Extractor วิธีนี้จะนำโครงข่ายประสาทแบบคอนโวลูชันมาลบชั้นสุดท้ายออกไปจากนั้นจะส่วนที่เหลือจะถือเป็น Fixed Feature Extractor สำหรับชุดข้อมูลชุดใหม่
2. Fine-Tuning แบบจำลองโครงข่ายประสาทแบบคอนโวลูชันทั้งหมด วิธีจะต่างจากวิธีแรกเล็กน้อยโดยแทนที่จะฝึกฝนเฉพาะส่วนหัว แต่สามารถ Fine-Tuning ทั้งแบบจำลองโครงข่ายประสาทแบบคอนโวลูชันในทุกชั้น เพื่อให้ได้ประสิทธิภาพที่ดีขึ้นทั้งการทำงานและชุดข้อมูลใหม่
3. ใช้ Pre-Trained Model เนื่องจากโครงข่ายประสาทแบบคอนโวลูชันสมัยใหม่ใช้เวลาหลายสัปดาห์บนเครื่องเซิร์ฟเวอร์ที่มีพลังในการประมวลผลสูงและมีหลาย GPU จึงมีผู้นำ Pre-Trained Model ซึ่งเป็นแบบจำลองที่ผ่านการฝึกฝนมาเรียบร้อยแล้วมาแบ่งปันกันผ่าน GitHub โดยให้บุคคลทั่วไปสามารถนำไปใช้ประโยชน์และปรับแต่งเครือข่ายได้ เช่น Tensorflow 2 Object Detection Model Zoo โดยเราจะนำ Pre-Trained Model ที่ผ่านการฝึกฝนเพื่อให้สามารถทำนายวัตถุที่มีฉลากกำกับชื่อจากตัวอักษร "A", "B", "C" และ "D" และชุดข้อมูลใหม่ที่มีฉลากกำกับชื่อ "W", "X", "Y" และ "Z" จากนั้นทำการฝึกฝนแบบจำลองด้วยชุดข้อมูลใหม่ของเราเพียงเลเยอร์สุดท้ายของแบบจำลอง หลักจากนั้นฝึกฝนเสร็จจะได้เลเยอร์ชั้นสุดท้ายอันใหม่ที่สามารถทำนายวัตถุที่มีฉลากกำกับ "W", "X", "Y" และ "Z" ได้

ดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 2.10 ตัวอย่างการทำ Transfer Learning ด้วย Pre-Trained Model [8]

## 2.2.4 Pre-Trained model ที่ใช้ในงานวิจัย

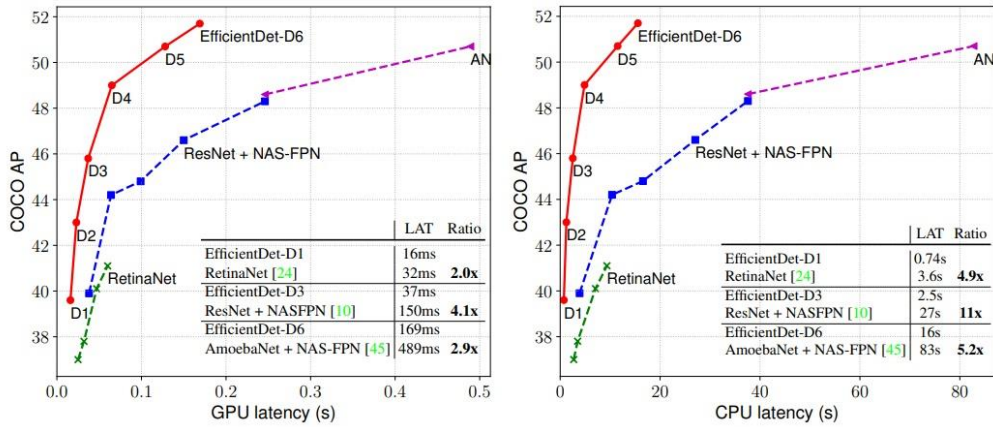
EfficientDet เป็นสถาปัตยกรรมเครือข่ายประสาทสำหรับการตรวจจับวัตถุ เป็นหนึ่งใน Tensorflow Object Detection API จาก Tensorflow 2 Object Detection Model Zoo [9] ในแบบจำลองรุ่นต่างๆ เช่น CenterNet, MobileNet, ResNet และ Fast R-CNN

EfficientDets [10] เป็นตระกูลของแบบจำลองการตรวจจับวัตถุที่ล้ำสมัยที่ผ่านค่า 55.1 Mean Average Precision (mAP) ในการทดสอบของ COCO ในขณะที่มีขนาดเล็กกว่า 4 ถึง 9 เท่า และใช้ FLOP น้อยกว่าเครื่องตรวจจับรุ่นอื่นๆ 13 ถึง 42 เท่า รุ่นนี้ยังทำงานเร็วขึ้น 2 ถึง 4 เท่าบน GPU และ 5 ถึง 11 เท่าบน CPU เร็วกว่าเครื่องตรวจจับอื่นๆ ซึ่งหมายความว่าแบบจำลอง EfficientDet จะมีค่า Latency ที่ต่ำกว่าแบบจำลองตัวอื่น ซึ่งทำให้สามารถฝึกฝนได้รวดเร็วกว่าตัวอื่นแต่ยังคงความแม่นยำที่สูง

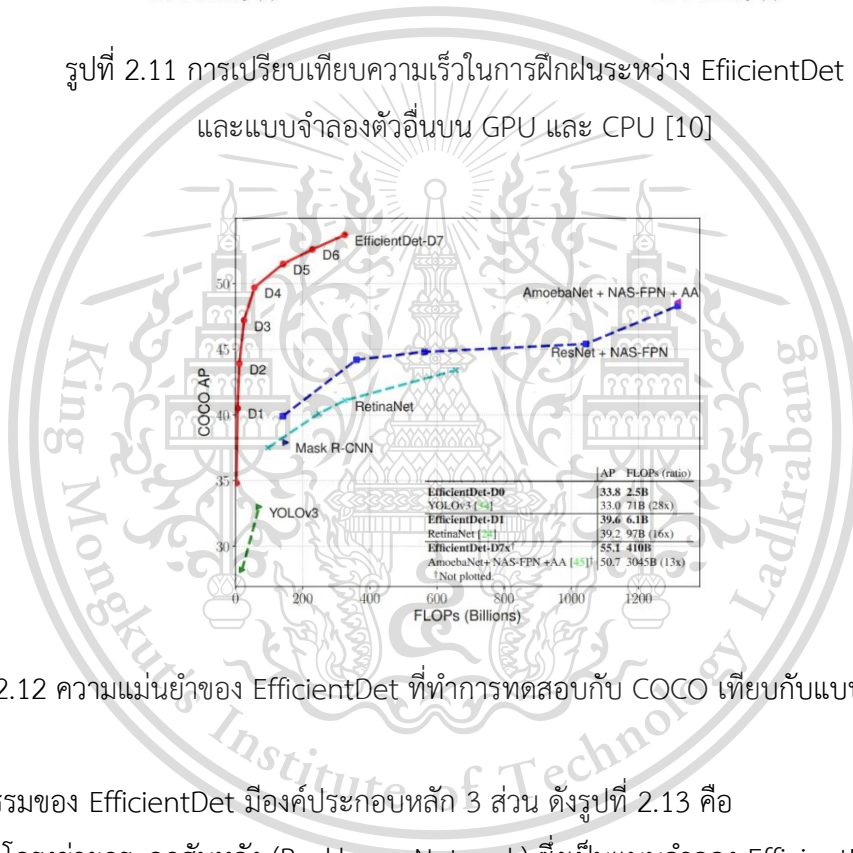
ดังรูปที่ 2.11 ดังนั้นจึงเป็นสถาปัตยกรรมที่เหมาะสมสำหรับทางผู้จัดทำเนื่องจากข้อกีดในเรื่องของทรัพยากร ในการประมวลผลที่ต่ำ ตระกูลแบบจำลองของ EfficientDet จะเริ่มต้นจาก EfficientDet-D0 ซึ่งมีความแม่นยำเทียบเท่ากับ YOLOv3 จากนั้นเราจะขยายแบบจำลองพื้นฐานนี้โดยใช้วิธีการสเกลแบบผสมของเราเพื่อรับรายการแบบจำลองการตรวจจับ EfficientDet-D1 ที่มีน้ำหนักเบาไปจนถึง EfficientDet-D6 ที่

มีน้ำหนักมาก โดยมีการแลกเปลี่ยนระหว่างความแม่นยำและความซับซ้อนของแบบจำลองที่แตกต่างกัน ดังรูปที่ 2.12 ซึ่งยังมีน้ำหนักมากก็ยังคงใช้ทรัพยากรในการประมวลผลมากขึ้นเนื่องจากมีความละเอียดที่มากขึ้น

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 2.11 การเปรียบเทียบความเร็วในการฝึกฝนระหว่าง EfficientDet และแบบจำลองตัวอื่นบน GPU และ CPU [10]



รูปที่ 2.12 ความแม่นยำของ EfficientDet ที่ทำการทดสอบกับ COCO เทียบกับแบบจำลองอื่น

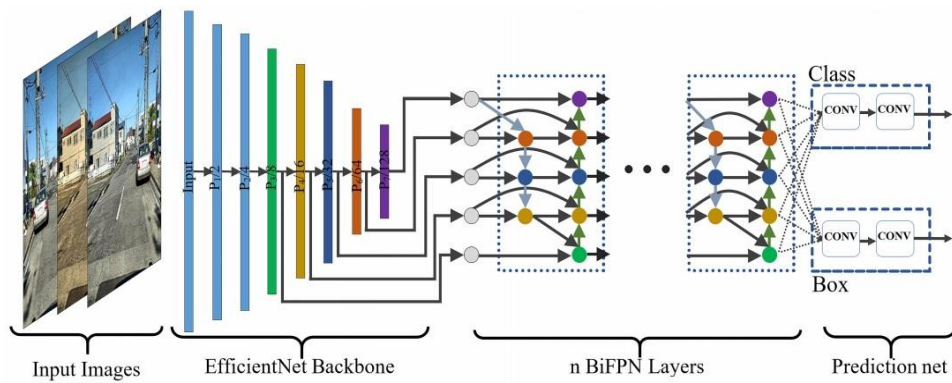
สถาปัตยกรรมของ EfficientDet มีองค์ประกอบหลัก 3 ส่วน ดังรูปที่ 2.13 คือ

1. โครงข่ายกระดูกสันหลัง (Backbone Network) ซึ่งเป็นแบบจำลอง EfficientNet สำหรับสกัดคุณลักษณะจากภาพที่เราได้มา
2. โครงข่ายคุณลักษณะ (Feature Network) ใช้เป็น BiFPN ซึ่งเป็นโครงข่ายแบบ 2 ทิศทางที่รับคุณลักษณะหลายระดับจากกระดูกสันหลังเป็นอินพุตและส่งออกรายการคุณลักษณะที่หลอมรวมซึ่งแสดงถึงลักษณะเด่นของภาพเป็นเอาต์พุต
3. โครงข่ายคลาสหรือกล่อง (Class/Box Network) จะใช้คุณลักษณะที่ได้มาทำนายคลาสและ

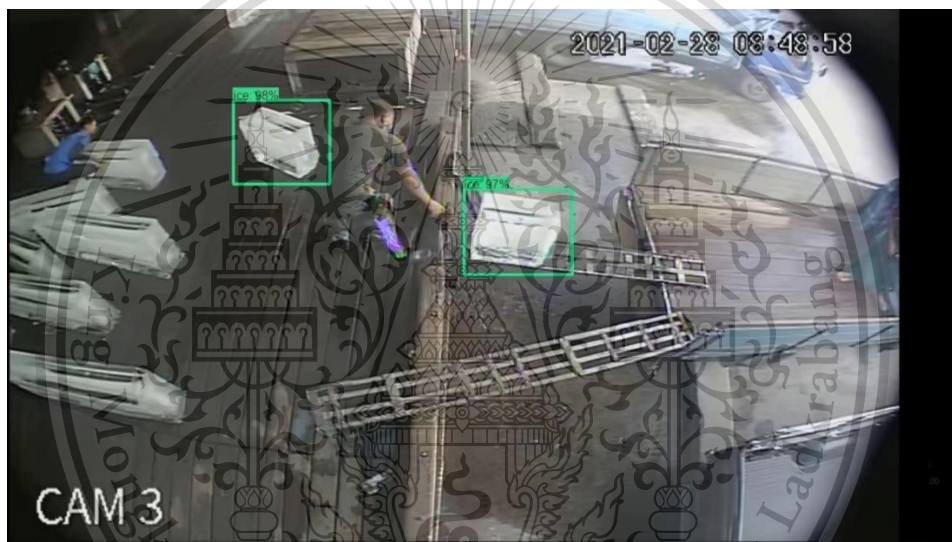
ตำแหน่งของแต่ละวัตถุในรูปภาพ ดังรูปที่ 2.14

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 2.13 สถาปัตยกรรมของแบบจำลอง EfficientDet [10]



รูปที่ 2.14 เอ้าท์พุทจากการทำนายคลาสและตำแหน่งของวัตถุในรูปภาพ

## 2.3 ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนา

### 2.3.1 ภาษา Python

ภาษา Python เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยากรณ์ของภาษาทางคอมพิวเตอร์ออกไป ในส่วนของ การแปลงชุดคำสั่งที่เขียนให้เป็นภาษาคอมพิวเตอร์ Python มีการทำงานแบบ Interpreter เป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่ต้องการ นอกจากนี้ ภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัด

อยู่ทำงานเฉพาะทางใดทางหนึ่ง (General-Purpose Language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรทั่วโลก [11]

This material is reserved for educational use only, not allowed for commercial use.

## 2.3.2 ไลบรารีที่ใช้ในการพัฒนา

### 2.3.2.1 OpenCV

OpenCV ย่อมาจาก Open Source Computer Vision คือ ไลบรารีฟังก์ชันการเขียนโปรแกรม โดยมีเป้าหมายไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์โดย OpenCV ถูกพัฒนาขึ้นจากบริษัท Intel ลักษณะสำคัญของ OpenCV คือ ไลบรารีแบบข้ามแพลตฟอร์มสามารถใช้งานได้ฟรีแบบ Open Source โดยยังรองรับ Deep Learning Frameworks เช่น Tensorflow, Torch และ PyTorch เป็นต้น

ขอบเขตการใช้งานของ OpenCV สามารถใช้งานได้อย่างกว้างขวาง มีความสามารถหลากหลาย รองรับการใช้งานทั้งภาพนิ่งและภาพวิดีโอเคลื่อนไหว เครื่องมือที่ใช้มีตั้งแต่ระดับพื้นฐานไปจนถึงระดับสูง ซึ่งรวมถึงการใช้เทคนิคการเรียนรู้ของเครื่องด้วย OpenCV นั้นถูกเขียนขึ้นมาโดยภาษา C++ อีกทั้งยังมีการรองรับภาษา Python, Java และ Matlab ซึ่งได้มีการรวบรวมไว้หลากหลายภาษา เช่น C#, Perl, Ch, Haskell และ Ruby ได้รับการพัฒนาเพื่อส่งเสริมการนำมาใช้งานโดยผู้ใช้ที่มีเพิ่มขึ้น โมดูลในภาษา Python ที่ใช้งานอย่างแพร่หลายอย่าง OpenCV-python ซึ่งใช้สำหรับเขียนโปรแกรมใช้งาน OpenCV ผ่านภาษา Python ได้ สามารถใช้ปรับแต่งรูปภาพหรือวิดีโอที่นำเข้า เช่น การหมุน การย่อขยาย การใช้ฟังก์ชันเบลอ การทำโมเสก การหาขอบภาพ และการตรวจจับวัตถุในภาพ เป็นต้น [12] การตรวจจับวัตถุที่สนใจในรูปที่ 2.15 ไลบรารี OpenCV สามารถกำหนดตำแหน่งของกรอบรอบวัตถุ ความหนาของกรอบ สีของกรอบ และจำแนกวัตถุในแต่ละประเภทที่แตกต่างกัน



รูปที่ 2.15 การสร้างกรอบรอบวัตถุที่สนใจในการการตรวจจับวัตถุในภาพ [13]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

### 2.3.2.2 Tensorflow

Tensorflow คือ ไบเบรารีสำหรับการเรียนรู้เชิงลึก ถูกพัฒนาโดยบริษัท Google โดยทางผู้พัฒนาได้ใช้ Machine Learning เพิ่มประสิทธิภาพกับผลิตภัณฑ์มากมาย ไม่ว่าจะเป็นเครื่องมือค้นหา (Search Engine) การแปลภาษา (Translation) คำบรรยายภาพ (Image Captioning) และเครื่องมือช่วยการเสนอแนะ (Recommendations)

โดย Tensorflow สามารถรับข้อมูลเป็นอาร์เรย์หลายมิติที่เรียกกันว่า Tensors และผู้ใช้งานมีหน้าที่จัดเรียงลำดับการประมวลผลเป็น Flowchart ข้อมูลที่ถูกป้อนไปก็จะไหล (Flow) ผ่านลำดับของกระบวนการจนออกมาเป็นผลลัพธ์ ด้านสถาปัตยกรรมแบ่งเป็น 4 ส่วน คือ การเตรียมประมวลผลข้อมูล การสร้างแบบจำลอง การฝึกแบบจำลอง และประเมินแบบจำลอง

ด้านการประมวลผลสามารถใช้การประมวลผลได้ทั้ง CPUs และ GPUs ซึ่งการใช้งาน GPUs จะทำให้ Tensorflow สามารถทำงานได้ดีและรวดเร็วกว่าในด้านการคำนวณทางคณิตศาสตร์ อย่างการคำนวณเมทริกซ์และพีชคณิตที่ต้องการหาผลคูณเมทริกซ์จำนวนมาก Tensorflow ถูกเขียนขึ้นจากภาษาทางคอมพิวเตอร์ C++ และ Python [14]

### 2.3.2.3 Firebase

Firebase คือ Platform ที่รวบรวมเครื่องมือต่างๆ สำหรับการจัดการในส่วนของ Backend หรือ Server Side ของบริษัท Google สามารถทำให้ Build Mobile Application ได้อย่างมีประสิทธิภาพและลดเวลาและค่าใช้จ่ายของการทำ Server Side หรือการวิเคราะห์ข้อมูล โดยมีทั้งเครื่องมือสาธารณะและเครื่องมือที่มีค่าใช้จ่ายตามระดับการใช้งาน

จัดเก็บและเชื่อมต่อข้อมูลระหว่างผู้ใช้และอุปกรณ์ในระดับโลก โดยใช้ฐานข้อมูลแบบ NoSQL ที่โฮสต์บนคลาวด์ Cloud Firestore ให้เกิดการซิงโครไนซ์แบบสดและการสนับสนุนแบบออฟไลน์ไปพร้อมกับการสืบค้นข้อมูลที่มีประสิทธิภาพ การบูรณาการ Firebase ร่วมกับผลิตภัณฑ์อื่นๆ จะช่วยให้สร้าง Application แบบไร้เซิร์ฟเวอร์ได้ จัดการผู้ใช้ด้วยวิธีที่ง่ายและปลอดภัยด้วย Firebase Auth ที่มีหลายวิธีในการตรวจสอบสิทธิ์รวมถึงอีเมลและรหัสผ่านผู้ให้บริการบุคคลที่สาม เช่น Google และ Facebook ที่ใช้ระบบบัญชีโดยตรง

Realtime Database คือ ฐานข้อมูลดั้งเดิมของ Firebase เป็นการดำเนินการที่มีประสิทธิภาพและมีเวลาแฝงต่ำสำหรับ Application บนอุปกรณ์เคลื่อนที่ที่ต้องการสถานะการเชื่อมต่อแบบเรียลไทม์ นิยมใช้ Cloud Firestore แทน Realtime Database สำหรับนักพัฒนาส่วนใหญ่ที่เริ่มโครงการใหม่ ส่งข้อความและการแจ้งเตือนไปยังผู้ใช้ข้ามแพลตฟอร์มทั้ง Android iOS และเว็บไซต์ได้ฟรี สามารถส่งข้อความไปยังอุปกรณ์เดียวกลุ่มอุปกรณ์หรือหัวข้อเฉพาะหรือกลุ่มผู้ใช้ Firebase Cloud Messaging (FCM) ปรับขนาดเป็น Application ที่ใหญ่ที่สุดโดยส่งข้อความหลายแสนล้านข้อความต่อวัน [15]

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

## บทที่ 3

### วิธีการดำเนินงาน

วิธีการดำเนินงานจะกล่าวถึงการสร้างระบบการตรวจนับน้ำแข็งแบบเรียลไทม์ (Realtime Ice Counting System) โดยอาศัยหลักการพื้นฐานจากการตรวจจับวัตถุ (Object Detection) ในการตรวจนับน้ำแข็งซึ่งใช้เป็นจุดอ้างอิงในการนับเมื่อผ่านพื้นที่ที่สนใจในเฟรม โดยอาศัยการเขียนโปรแกรมจากภาษา Python จากนั้นจะจัดเก็บจำนวนข้อมูลในการตรวจนับที่ได้แบบเรียลไทม์ผ่าน Firebase เพื่อให้สามารถตรวจสอบข้อมูลได้ตลอดเวลา โดยสำหรับวิธีการดำเนินงานประกอบด้วย 3 ขั้นตอนหลักๆ ดังนี้

- 3.1 การเตรียมชุดข้อมูล
- 3.2 การสร้างระบบตรวจนับน้ำแข็ง
- 3.3 การสร้างระบบการตรวจนับน้ำแข็งแบบเรียลไทม์

#### 3.1 การเตรียมชุดข้อมูล

เป็นกระบวนการที่จัดการกับข้อมูลเพื่อปรับเปลี่ยนข้อมูลให้อยู่ในรูปแบบที่เหมาะสม ก่อนจะนำเข้าระบบประมวลผลและนำไปฝึกฝนกับแบบจำลองต่อไป

##### 3.1.1 การรวบรวมข้อมูล (Data Collecting)

ทำการเก็บรวบรวมข้อมูลดิบในรูปแบบของวิดีโอซึ่งเป็นวิดีโอตัวอย่างการลำเลียงน้ำแข็งของขึ้นรถจากห้องวงจรปิดที่ทางบริษัทได้ทำการบันทึกไว้ โดยชุดข้อมูลเหล่านี้ได้มาจากกล้องวงจรปิด 4 ตัวที่ติดตั้งอยู่ในสถานีขนส่งลำเลียงน้ำแข็งของขึ้นรถ มุมกล้องที่ติดตั้งจะแตกต่างกันตามสายการขนส่งซึ่งมีทั้งหมด 4 สาย ในแต่ละมุมกล้องมีการบันทึกตั้งแต่ช่วงเวลาเช้าถึงกลางคืน ทางผู้จัดทำได้เลือกวิดีโอจากแต่ละมุมกล้องมาอย่างละ 2 ไฟล์ ประกอบด้วยช่วงเวลาเช้าและกลางคืน รวมทั้งหมด 8 ไฟล์ หลังจากนั้นนำไฟล์วิดีโอตัวอย่างเหล่านี้มาทำแปลงเป็นภาพนิ่งหลายเฟรมให้อยู่ในรูปแบบไฟล์ jpg ดังรูปที่ 3.1 โดยแต่ละเฟรมจะถูกนำไปเปลี่ยนให้เหมาะสมเป็นอินพุตสำหรับการฝึกฝนแบบจำลองต่อไป



เอกสารนี้เป็นเอกสารที่สง

ลิขสิทธิ์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

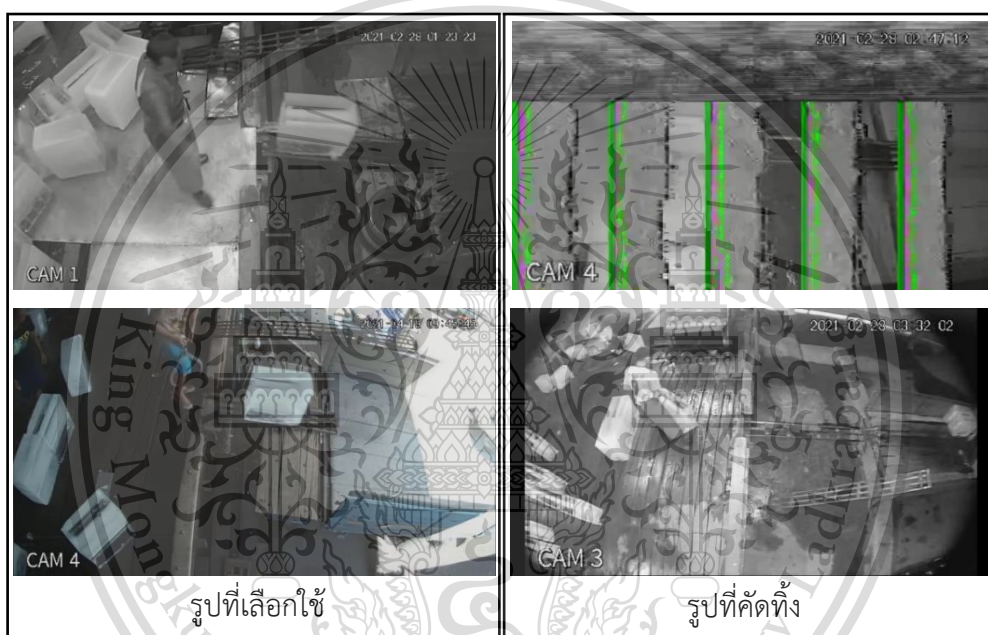
รูปที่ 3.1 เฟรมตัวอย่างของแต่ละมุมกล้องในช่วงเวลากลางวันและกลางคืน

This material is reserved for educational use only, not allowed for commercial use.

### 3.1.2 การทำความสะอาดข้อมูล (Cleaning Data)

เป็นขั้นตอนการคัดกรองข้อมูลรูปภาพ กำจัดรูปภาพที่มีความไม่สมบูรณ์หรือไม่สำคัญในการนำไปฝึกฝนแบบจำลองเพื่อให้แบบจำลองมีความแปรปรวนน้อยที่สุด โดยทางผู้จัดทำจะทำการคัดกรองรูปภาพจากทั้งหมด 8 ไฟล์ จาก 4 มุมกล้อง และ 2 ช่วงเวลา มาอย่างละ 50 รูปภาพ รวมทั้งหมด 400 ภาพ ดังตัวอย่างในรูปที่ 3.2 โดยเงื่อนไขในการเลือกและกำจัดรูปภาพได้แก่

1. เลือกรูปภาพที่เห็นน้ำแข็งของชุดเจนบนรางลำเลียง
2. กำจัดรูปภาพที่มีความเบลหรือคุณภาพต่ำ
3. กำจัดรูปภาพที่มีการบดบังน้ำแข็ง



รูปที่ 3.2 รูปภาพตัวอย่างรูปที่เลือกใช้และรูปที่คัดทิ้ง

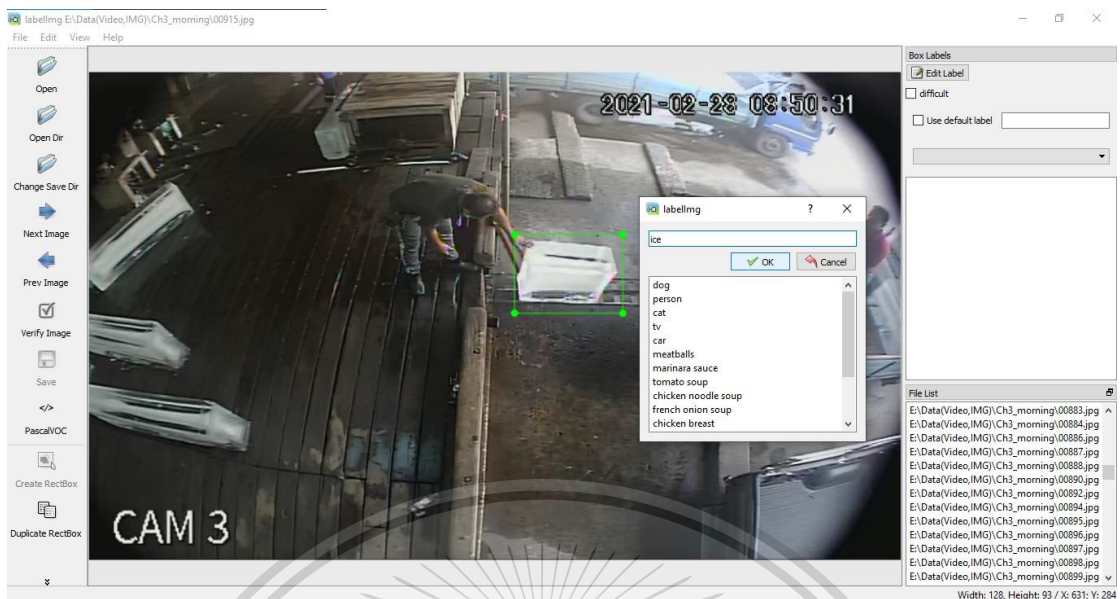
### 3.1.3 การกำกับชื่อรูป (Image Annotation)

นำเฟรมทั้งหมดมาทำการระบุตำแหน่งและระบุป้ายฉลากกำกับชื่อของน้ำแข็งของให้สามารถฝึกฝนแบบจำลอง เพื่อค้นหาน้ำแข็งและสามารถรับรู้ได้ว่าในพิกัดตำแหน่งนี้เป็นน้ำแข็งอย่างที่เราต้องการ โดยประเภทของการทำ Image Annotate ที่ทางผู้จัดทำใช้ดำเนินการ คือ กล่องระบุตำแหน่ง (Bounding Box) ซึ่งเป็นกล่องสี่เหลี่ยมที่ใช้กำหนดตำแหน่งของวัตถุเป้าหมายที่สนใจในรูปภาพ

ผู้จัดทำใช้ซอฟต์แวร์ชื่อ “Labelimg” ในการสร้างกล่องระบุตำแหน่ง และระบุป้ายฉลากกำกับชื่อของวัตถุในภาพ โดยเริ่มจาก

1. กำหนดชื่อประเภทของวัตถุในซอฟต์แวร์ซึ่งสำหรับงานวิจัยนี้มีเพียงประเภทเดียว คือ “ice” การดำเนินการนี้ไม่ว่ากรณีใดๆทั้งสิ้น
2. ทำการวาดกล่องระบุตำแหน่งล้อมรอบน้ำแข็งช่องในแต่ละเฟรม ดังรูปที่ 3.3 ครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 3.3 การ Annotate image โดยใช้ Labelimg

เมื่อทำเสร็จครบทุกเฟรมแล้วจะได้ไฟล์ Extensible Markup Language (XML) สำหรับในแต่ละเฟรมภาพ ซึ่งในไฟล์ XML ประกอบไปด้วยตำแหน่งของกล่อง ชื่อไฟล์ และชื่อประเภทของวัตถุ สำหรับแต่ละวัตถุในรูปภาพ จะเห็นว่าน้ำหนักแข็งของของเราถูกติดป้ายฉลากกำกับชื่อว่า “ice” และตัวเลขระบุพิกัดทั้งจุดตามแนวแกนตั้งและแนวแกนนอนของ Bounding box ได้แก่  $x_{min}$ ,  $y_{min}$ ,  $x_{max}$  และ  $y_{max}$  ดังแสดงในรูปที่ 3.4

```

00915.xml 00890.xml
1 <annotation>
2   <folder>Ch3_morning</folder>
3   <filename>00915.jpg</filename>
4   <path>E:\Data (Video,IMG)\Ch3_morning\00915.jpg</path>
5   <source>
6     <database>Unknown</database>
7   </source>
8   <size>
9     <width>960</width>
10    <height>576</height>
11    <depth>3</depth>
12  </size>
13  <segmented>0</segmented>
14  <object>
15    <name>ice</name>
16    <pose>Unspecified</pose>
17    <truncated>0</truncated>
18    <difficult>0</difficult>
19    <bndbox>
20      <xmin>503</xmin>
21      <ymin>191</ymin>
22      <xmax>631</xmax>
23      <ymax>284</ymax>
24    </bndbox>
25  </object>
26 </annotation>
27

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 3.4 โครงสร้างของ XML ไฟล์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

### 3.1.4 การสร้างไฟล์ Comma Separated Value (CSV)

การเขียนโปรแกรม Python แปลงไฟล์ข้อมูลของความละเอียดของเฟรม ชื่อประเภทของวัตถุ และตำแหน่งของ Bounding Box ที่ล้อมรอบน้ำแข็งในแต่ละเฟรมจากไฟล์ XML ที่ได้มาจากการขั้นตอนก่อน มาเก็บเป็นข้อมูลในรูปแบบตาราง ดังรูปที่ 3.5

	A	B	C	D	E	F	G	H
1	filename	width	height	name	xmin	ymin	xmax	ymax
2	00035.jpg	960	576	ice	550	201	667	298
3	00035.jpg	960	576	ice	372	259	452	368
4	00035.jpg	960	576	ice	298	291	409	407
5	00036.jpg	960	576	ice	612	212	732	295
6	00036.jpg	960	576	ice	375	253	447	370
7	00036.jpg	960	576	ice	299	291	409	404
8	00037.jpg	960	576	ice	669	214	783	300
9	00037.jpg	960	576	ice	377	255	445	372
10	00037.jpg	960	576	ice	299	286	406	405
11	00048.jpg	960	576	ice	230	94	290	170
12	00048.jpg	960	576	ice	276	140	362	254
13	00048.jpg	960	576	ice	376	258	452	371
14	00048.jpg	960	576	ice	299	288	412	408
15	00053.jpg	960	576	ice	457	199	584	289
16	00053.jpg	960	576	ice	373	257	451	375
17	00053.jpg	960	576	ice	299	288	407	406
18	00054.jpg	960	576	ice	508	197	631	296

รูปที่ 3.5 ตัวอย่างไฟล์ CSV

### 3.1.5 การสร้างไฟล์ Tensorflow Record

เนื่องจากต้องมีการฝึกฝนแบบจำลองผ่าน Tensorflow ซึ่งจะยอมรับอินพุตในรูปแบบมาตรฐานที่เรียกว่าไฟล์ TFRecord โดยขั้นตอนนี้ใช้การเขียนโปรแกรม Python เพื่อแปลงไฟล์ข้อมูลในรูปแบบตารางของไฟล์ CSV จากขั้นตอนที่แล้ว ให้อยู่ในรูปแบบไฟล์ TFRecord

### 3.1.6 การสร้าง Label Map

ชุดข้อมูลแต่ละชุดต้องการ Label Map ที่สอดคล้องตรงกับชุดข้อมูลนั้นๆ เพื่อให้แสดงตัวแปร id และ ฉลากกำกับชื่อของวัตถุเราในรูปแบบ Dictionary ซึ่งจะใช้ในการบ่งบอกประเภทของวัตถุตอนเอาที่พุดจากการระบบการตรวจจับวัตถุ ซึ่งประเภทของเราในที่มีชนิดเดียว คือ “ice” กำหนดให้เป็น id : 1 ดังรูปที่ 3.6

```
label_map.pbtxt  X
1  item {
2    name: "ice"
3    id: 1
4  }
5
```

รูปที่ 3.6 ตัวอย่างไฟล์ Label Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะบุคคลเท่านั้น ไม่ควรเผยแพร่ให้ผู้อื่นได้โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

## 3.2 การสร้างระบบการตรวจจับน้ำแข็ง

วัตถุประสงค์หลักของขั้นตอนนี้ คือ การทำให้คอมพิวเตอร์รับรู้ถึงพิกัดของน้ำแข็งโดยการตรวจจับน้ำแข็งและระบุกรอบล้อมรอบวัตถุที่ระบุตำแหน่งของน้ำแข็งสำหรับทุกๆ เฟรมและใช้เป็นอินพุตด้วยหลักการ Transfer Learning ด้วย Pre-Trained Model ซึ่งใช้เป็นตัวตั้งต้นในการเริ่มต้นหลังจากนั้นจะนำไปฝึกฝนกับชุดข้อมูลที่เราได้จัดเตรียมมา

### 3.2.1 การเลือกใช้ Pre-Trained Model

ในปัจจุบันมีอัลกอริทึมมากมายให้เลือกใช้สำหรับการตรวจจับวัตถุแต่การเลือกแบบจำลองให้เหมาะสมนั้นใช้เวลาค่อนข้างมากเนื่องจากมีหลายปัจจัยที่เกี่ยวข้องรวมถึงข้อกำหนดของทรัพยากรที่ใช้ในการประมวลผล โดยทางผู้จัดทำได้เลือกใช้แบบจำลอง EfficientDet D0 ความละเอียด 512x512 ซึ่งเป็น Pre-Trained Model จาก Tensorflow 2 Object Detection Model Zoo เนื่องจากผู้จัดทำมีข้อจำกัดในเรื่องของทรัพยากรในการประมวลผลที่ต่ำ โดยตัวแบบจำลองในการตรวจจับวัตถุนี้ใช้เวลาในการฝึกฝนเร็วกว่าแบบจำลองตัวอื่นบน CPU แต่ยังคงความแม่นยำที่สูง

### 3.2.2 การปรับค่าพารามิเตอร์

การพารามิเตอร์ของแบบจำลองก่อนนำไปฝึกฝนของงานวิจัยมี 3 ค่าดังแสดง ตารางที่ 3.1

1. Number of Classes คือ จำนวนประเภทของวัตถุที่สนใจ โดยในงานวิจัยนี้มีเพียงชนิดเดียว
2. Batch Size คือ จำนวนตัวอย่างของชุดข้อมูลที่ส่งเข้าไปฝึกฝนกับแบบจำลองแต่ละรอบ ยิ่งมีจำนวนมากก็จะยิ่งใช้หน่วยความจำมากขึ้น
3. Score Threshold เป็นพารามิเตอร์ที่กำหนดคะแนนความเชื่อมั่นขั้นต่ำสำหรับการตรวจจับ โดยถ้าหากคะแนนความเชื่อมั่นจากการทำนายไม่ถึงเกณฑ์ที่กำหนดไว้ผลลัพธ์จะถูกคัดออก

ตารางที่ 3.1 การปรับค่าพารามิเตอร์

ชื่อพารามิเตอร์	ค่า
Number of Classes	1
Batch Size	4
Score_Threshold	0.5 (50%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

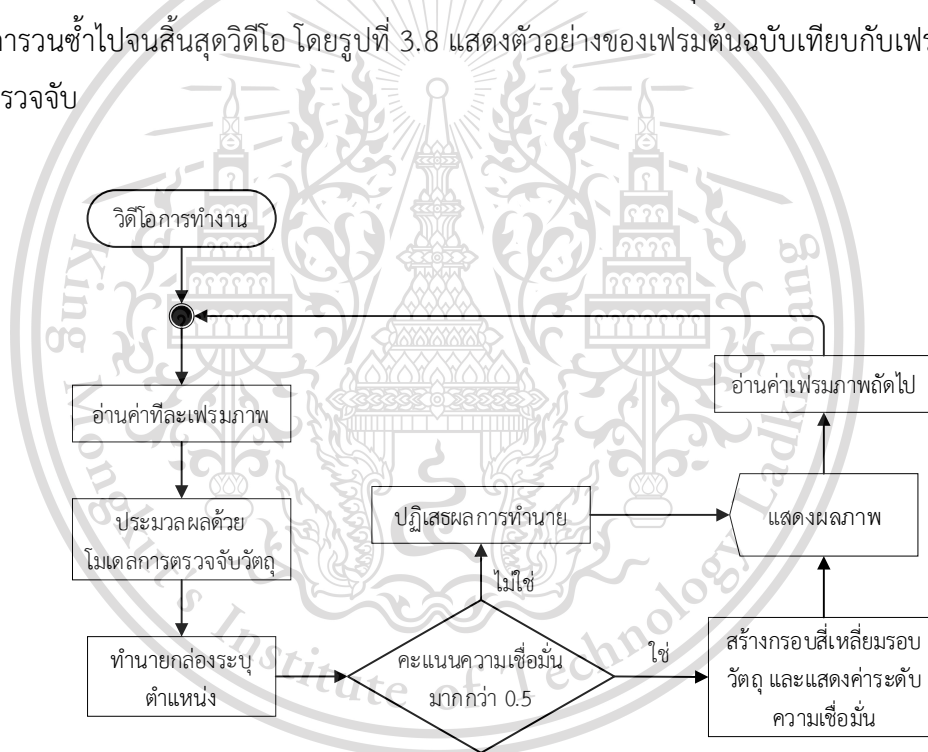
This material is reserved for educational use only, not allowed for commercial use.

### 3.2.3 การฝึกฝนแบบจำลอง

การนำชุดข้อมูลฝึกฝนไฟล์ TFRecord ที่ได้จัดเตรียมไว้จะถูกนำไป ฝึกฝนกับ Pre-Trained Model บน Intel (R) Core (TM) i7-6700 HQ CPU @ 2.60 GHz

### 3.2.4 การทดลองการใช้งานของแบบจำลองกับวิดีโอตัวอย่าง

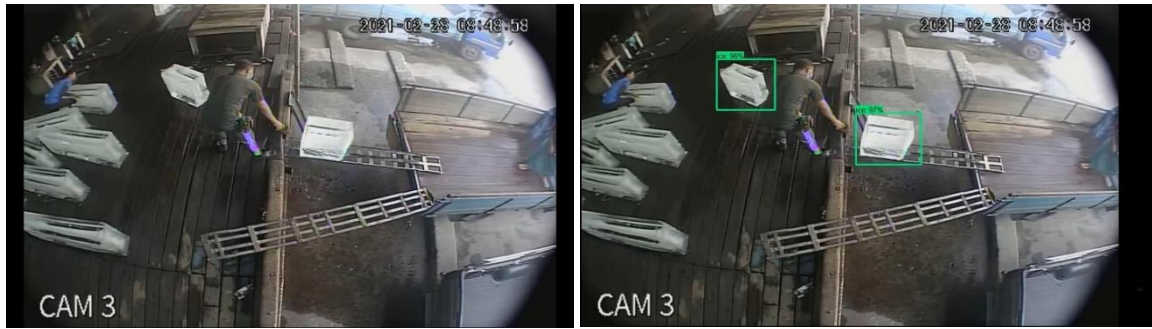
การนำแบบจำลองที่ผ่านการฝึกฝนแล้วจะถูกนำมาทดลองกับคลิปวิดีโอตัวอย่างโดยใช้ภาษา Python ในการเขียนระบบมีการทำงาน ดังรูปที่ 3.7 (Flowchart) เริ่มตั้งแต่การนำวิดีโอตัวอย่างเข้าระบบ และอ่านทีละเฟรมอาศัยไลบรารี OpenCV โดยที่แต่ละเฟรมจะถูกนำมาประมวลผลเพื่อตรวจจับวัตถุผ่าน Tensorflow เพื่อทำนายผลการตรวจจับออกมาเป็นในลักษณะของกล่องระบุตำแหน่งและฉลากกำกับชื่อของวัตถุ โดยมีเงื่อนไขว่าคะแนนความเชื่อมั่น (Score Threshold) ต้องมากกว่า 0.5 ถึงจะยอมรับการทำนายนั้น หลังจากนั้นจะประมวลผลภาพเพื่อแสดงผลการตรวจจับวัตถุผ่านหน้าจอ และจะอ่านเฟรมถัดไปเป็นการวนซ้ำไปจนสิ้นสุดวิดีโอ โดยรูปที่ 3.8 แสดงตัวอย่างของเฟรมต้นฉบับเทียบกับเฟรมที่มีการแสดงผลการตรวจจับ



รูปที่ 3.7 หลักการทำงานของระบบการตรวจจับวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



เฟรมต้นฉบับ

เฟรมที่ผ่านการตรวจจับ

รูปที่ 3.8 ตัวอย่างการตรวจจับน้ำแข็งจากเฟรมต้นฉบับ

### 3.3 การสร้างระบบการตรวจนับน้ำแข็งแบบเรียลไทม์

วัตถุประสงค์ของขั้นตอนนี้ คือ การสร้างระบบการตรวจนับน้ำแข็งเมื่อผ่านเข้ามาในบริเวณที่กำหนด โดยวิธีการนับจะขึ้นอยู่กับจุดอ้างอิงและพื้นที่อ้างอิงที่ใช้ในการนับวัตถุ โดยมีขั้นตอนดังนี้

1. กำหนดจุดอ้างอิงของน้ำแข็ง
2. กำหนดบริเวณอ้างอิงสำหรับการนับ
3. เขียนโปรแกรม Python เพื่อการตรวจนับ

#### 3.3.1 การกำหนดจุดอ้างอิงของน้ำแข็ง

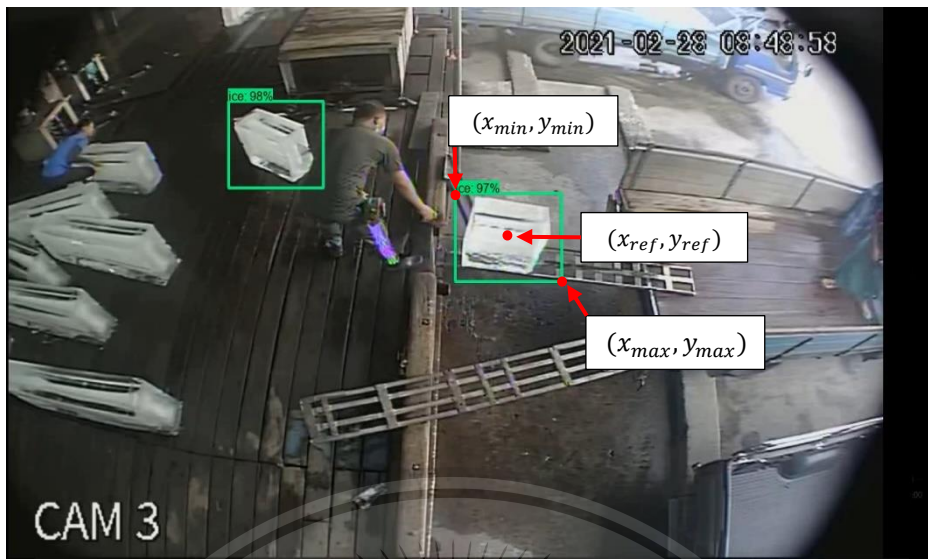
เอาท์พุทในแต่ละเฟรมที่ได้จากระบบการตรวจจับวัตถุจะได้กล่องระบุตำแหน่งที่มาพร้อมกับพิกัดและฉลากกำกับประเภทวัตถุ โดยผู้วิจัยนำค่าพิกัดจากกล่องมาหาค่ากึ่งกลางของกล่องของน้ำแข็งแต่ละก้อนเพื่อเป็นจุดอ้างอิงที่ใช้ในการนับ

สมมติให้ในรูปที่ 3.9 ระบบการตรวจจับวัตถุประมวลผลเฟรมและวาดกล่องระบุตำแหน่งขึ้นมา โดยมีพิกัดดังนี้  $(x_{min}, y_{min})$  และ  $(x_{max}, y_{max})$  โดยที่  $x_{min}$  และ  $y_{min}$  คือ พิกัด x และ y ของมุมบนซ้ายของกล่องตามลำดับ ในขณะที่  $x_{max}$  และ  $y_{max}$  คือ พิกัด x และ y ของมุมล่างขวาของกล่องตามลำดับ ดังนั้นพิกัดจุดกึ่งกลางของกล่องที่ใช้เป็นจุดอ้างอิงหาได้จากสมการที่ 3.1 โดยรูปที่ 3.9 แสดงตัวอย่างจุดอ้างอิงของวัตถุที่ใช้ในการนับ

$$(x_{ref}, y_{ref}) = (x_{min} + \frac{x_{max} - x_{min}}{2}, y_{min} + \frac{y_{max} - y_{min}}{2}) \quad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 3.9 ตัวอย่างพิกัดอ้างอิงของน้ำแข็ง

### 3.3.2 การกำหนดบริเวณอ้างอิงสำหรับการนับ

เมื่อคอมพิวเตอร์สามารถระบุจุดอ้างอิงของน้ำแข็งได้สำเร็จจะมีขั้นตอนที่จำเป็นต่อการนับ คือ พื้นที่ที่อ้างอิงในเฟรมจากวิดีโอเพื่อใช้เป็นบริเวณอ้างอิงในการนับน้ำแข็งเมื่อน้ำแข็งผ่านเข้ามาในบริเวณอ้างอิงที่เรากำหนดผ่าน Python ซึ่งจะกำหนดให้อยู่ในรูปสี่เหลี่ยมตั้งอยู่บนรางลำเลียงน้ำแข็ง โดยกำหนดขนาดให้มีความมั่นใจได้ว่าจุดอ้างอิงของน้ำแข็งจะผ่านเข้ามาและอยู่ในพื้นที่ที่กำหนดและการกำหนดบริเวณจะแตกต่างกันออกไปขึ้นอยู่กับมุมมองเนื่องจากทั้ง 4 มุมกล้องมีระยะและตำแหน่งการวางที่ต่างกันและจะวาดสี่เหลี่ยมเพื่อแสดงผลผ่านหน้าจอโดยใช้ไลบรารี OpenCV ดังรูปที่ 3.7



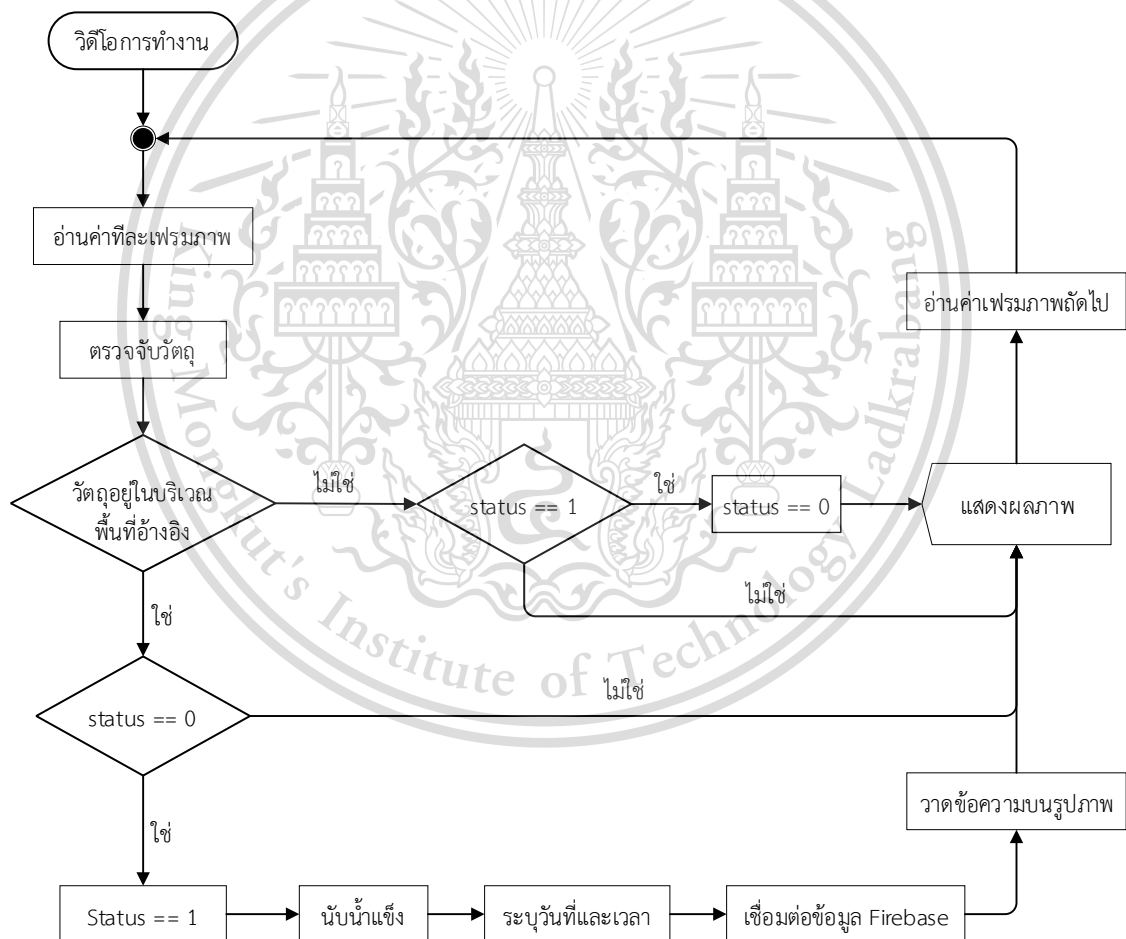
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น การนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

### 3.3.3 การเขียนภาษา Python เพื่อตรวจนับแบบเรียลไทม์

หลักการการทำงานของระบบการตรวจนับแบบเรียลไทม์จะมีการทำงานต่อจากระบบการตรวจจับโดยจะมีการตรวจสอบว่าจุดอ้างอิงของน้ำแข็งเข้ามาในบริเวณอ้างอิงหรือไม่ ถ้าหากว่ามีน้ำแข็งเข้ามาในระบบจะทำการนับพร้อมกับระบุวันที่และเวลาของน้ำแข็งแต่ละก้อนที่ถูกนับโดยใช้ไลบรารี Time โดยในการนับมีเงื่อนไขว่าในบริเวณที่อ้างอิงต้องมีน้ำแข็งอยู่หรือต้องรอให้น้ำแข็งก้อนเก่าออกจากบริเวณไปก่อนถึงจะพร้อมจะนับก้อนใหม่ กำหนดตัวแปรชื่อ Status ไว้บ่งบอกสถานะกรณี Status = 0 (Available) แสดงว่าพร้อมนับ ส่วนกรณี Status = 1 (Not Available) แสดงว่าไม่พร้อมนับ โดยหลังจากนับเสร็จระบบจะดึงค่าจำนวนน้ำแข็งที่นับได้จัดเก็บเป็นฐานข้อมูลผ่าน Firebase หลังจากนั้นระบบจะวาดข้อความของจำนวนน้ำแข็งที่นับได้โดยใช้ไลบรารี OpenCV และประมวลผลภาพขึ้นหน้าจอใน รูปที่ 3.11



รูปที่ 3.11 การทำงานของระบบการตรวจนับแบบเรียลไทม์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

## บทที่ 4

### ผลการดำเนินงาน

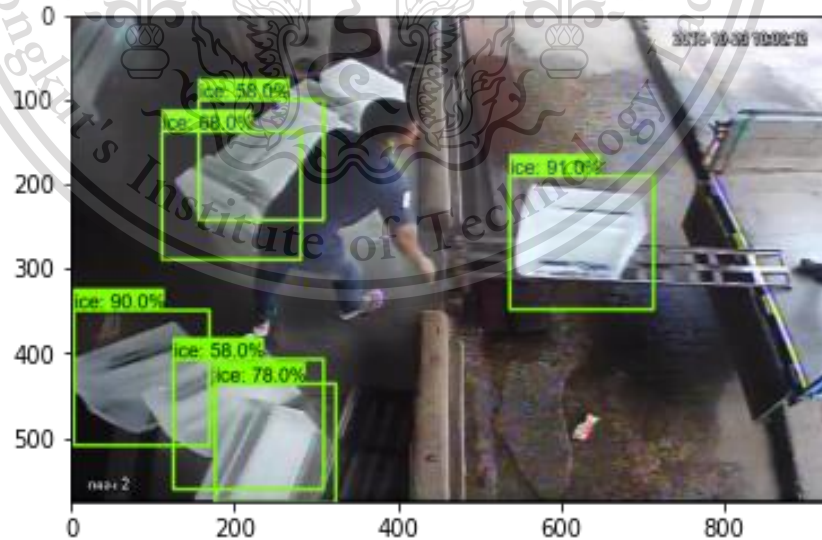
จากการศึกษาและดำเนินการนับน้ำแข็งของจากวิดีโอตัวอย่างการลำเลียงก้อนน้ำแข็ง ได้ผลลัพธ์ของการดำเนินการและการประมวลผลจากแบบจำลอง โดยแบ่งเป็นหัวข้อดังต่อไปนี้

4.1 ผลการดำเนินการจากการนับ

4.2 การแสดงผลแบบเรียลไทม์โดยใช้ Firebase

#### 4.1 ผลการดำเนินการจากการนับ

เมื่อนำแบบจำลองที่ปรับแต่งเสร็จออกมาใช้กับคลิปวิดีโอจากกล้อง CCTV ของโรงงานน้ำแข็ง ได้ผลลัพธ์ว่าก้อนน้ำแข็งของที่อยู่บนรางถูกตรวจจับและแสดงตัวเลขความแม่นยำในแต่ละภาพ โดยน้ำแข็งส่วนที่อยู่ฝั่งซ้ายของภาพที่มีการตรวจจับที่ความแม่นยำต่ำกว่าหรือไม่ถูกตรวจจับนั้นอาจเกิดจากการทับซ้อนกันของน้ำแข็งและมีพนักงานยืนบดบังแต่เป้าหมายที่สนใจคือ น้ำแข็งที่กำลังผ่านรางลำเลียงอย่างเดียว เนื่องจากระบบจะนับเฉพาะน้ำแข็งที่ผ่านรางลำเลียงขึ้นรถเท่านั้น ดังรูปที่ 4.1



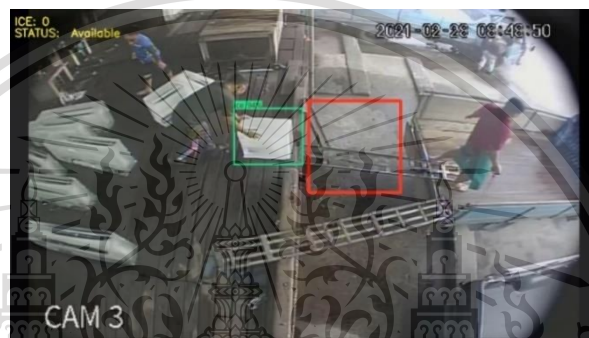
รูปที่ 4.1 การตรวจจับก้อนน้ำแข็งขณะลำเลียงขึ้นรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

รูปที่ 4.2 แสดงภาพตัวอย่างการแสดงผลการทำงานของระบบโดยแบ่งเป็น 3 ช่วง ประกอบด้วย

- (1) ช่วงก่อนน้ำแข็งเข้ามาในบริเวณอ้างอิง คือ สถานะพร้อมนับ และขึ้นกรอบบริเวณอ้างอิงสีแดง
- (2) ช่วงที่น้ำแข็งเข้ามาในบริเวณอ้างอิง คือ การนับน้ำแข็ง สถานะไม่พร้อมนับ และขึ้นกรอบบริเวณอ้างอิงสีเขียว
- (3) ช่วงที่น้ำแข็งออกจากบริเวณอ้างอิง คือ สถานะพร้อมนับ และขึ้นกรอบบริเวณอ้างอิงสีแดง ผลการดำเนินการจากการนับจากมุมมองทั้งหมดในช่วงเวลากลางวันและกลางคืน โดยมีตัวอักษรสีเหลืองบริเวณมุมซ้ายบนของหน้าจอแสดงจำนวนน้ำแข็งที่นับได้ ดังรูปที่ 4.3



(1) ช่วงก่อนเข้าพื้นที่อ้างอิง



(2) ช่วงที่น้ำแข็งเข้ามาในบริเวณอ้างอิง



(3) ช่วงที่น้ำแข็งออกจากบริเวณอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามรูปที่ 4.2 ภาพตัวอย่างการแสดงผลการทำงานของระบบการนับ ครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.



รูปที่ 4.3 การแสดงจำนวนน้ำแข็งที่นับสำเร็จบริเวณช้ายบนของจอแสดงผล

ผู้จัดทำได้ทำการทดสอบประสิทธิภาพของระบบการตรวจจับเป็นเปอร์เซ็นต์ความแม่นยำเฉลี่ยในการนับโดยแบ่งเป็น 2 แบบ ได้แก่ ความแม่นยำแบบแบ่งตามช่วงเวลากลางวันและกลางคืนและความแม่นยำแบบแบ่งตามกล้องทั้ง 4 ตัว

ผู้จัดทำได้ทำการทดสอบโดยเริ่มจากการสุ่มตัวอย่างวิดีโอในช่วงที่มีการทำงานจากทุกมุมกล้อง มุมกล้องละ 2 ช่วงเวลา คือ ช่วงเวลากลางวันและกลางคืนรวมทั้งหมด 8 ชุดวิดีโอ ในแต่ละชุดวิดีโอถูกแบ่งออกเป็น 3 วิดีโอที่แตกต่างกันโดยในแต่ละวิดีโอมีความยาว 5 นาที รวมทั้งหมด 24 การทดลอง โดยผลการนับในแต่ละการทดลองแสดงดังตารางที่ 4.1 โดยตัวเลขที่อยู่ด้านซ้ายในตารางคือ จำนวนที่ระบบนับได้ ส่วนตัวเลขที่อยู่ด้านขวาคือ จำนวนก้อนน้ำแข็งที่ลำเลียงจริง แสดงผลเป็นเปอร์เซ็นต์ความแม่นยำ เอกสารนี้เ้แบบแบ่งตามช่วงเวลาและแบ่งตามกล้องทั้ง 4 ตัว ดังตารางที่ 4.2 และ 4.3 ตามลำดับใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

ตารางที่ 4.1 อัตราส่วนจำนวนน้ำแข็งที่ระบบนับได้ต่อจำนวนน้ำแข็งที่ลำเลียงจริง

การวัดผล	เวลากลางวัน				เวลากลางคืน			
	กล่องที่ 1	กล่องที่ 2	กล่องที่ 3	กล่องที่ 4	กล่องที่ 1	กล่องที่ 2	กล่องที่ 3	กล่องที่ 4
วิดีโอที่ 1	35   35	30   36	32   32	15   15	37   37	24   24	29   29	28   28
วิดีโอที่ 2	29   29	31   31	39   39	23   23	41   42	32   32	26   26	10   26
วิดีโอที่ 3	24   26	26   27	27   27	20   22	34   34	19   21	33   33	26   26

ตารางที่ 4.2 เปอร์เซนต์ความถูกต้องของการวัดผลการนับ โดยแบ่งตามช่วงเวลา

การวัดผล	เวลากลางวัน				เวลากลางคืน			
	กล่องที่ 1 (%)	กล่องที่ 2 (%)	กล่องที่ 3 (%)	กล่องที่ 4 (%)	กล่องที่ 1 (%)	กล่องที่ 2 (%)	กล่องที่ 3 (%)	กล่องที่ 4 (%)
วิดีโอที่ 1	100.00	83.33	100.00	100.00	100.00	100.00	100.00	100.00
วิดีโอที่ 2	100.00	100.00	100.00	100.00	97.56	100.00	100.00	38.46
วิดีโอที่ 3	92.30	96.30	100.00	90.91	100.00	90.48	100.00	100.00
ค่าเฉลี่ย	97.43	93.21	100.00	96.97	99.19	96.83	100.00	79.49
% ค่าเฉลี่ยที่นับได้ในเวลากลางวัน = 96.90					% ค่าเฉลี่ยที่นับได้ในเวลากลางคืน = 93.88			

ตารางที่ 4.3 เปอร์เซนต์ความถูกต้องของการวัดผลการนับ โดยแบ่งตามมุมกล้อง

การวัดผล	กล่องที่ 1		กล่องที่ 2		กล่องที่ 3		กล่องที่ 4	
	กลางวัน (%)	กลางคืน (%)	กลางวัน (%)	กลางคืน (%)	กลางวัน (%)	กลางคืน (%)	กลางวัน (%)	กลางคืน (%)
วิดีโอที่ 1	100.00	100.00	83.33	100.00	100.00	100.00	100.00	100.00
วิดีโอที่ 2	100.00	97.56	100.00	100.00	100.00	100.00	100.00	38.46
วิดีโอที่ 3	92.30	100.00	96.30	90.48	100.00	100.00	90.91	100.00
ค่าเฉลี่ย	97.43	99.19	93.21	96.83	100.00	100.00	96.97	79.49
ค่าเฉลี่ยต่อกล้อง (%)	98.31		95.02		100.00		88.23	

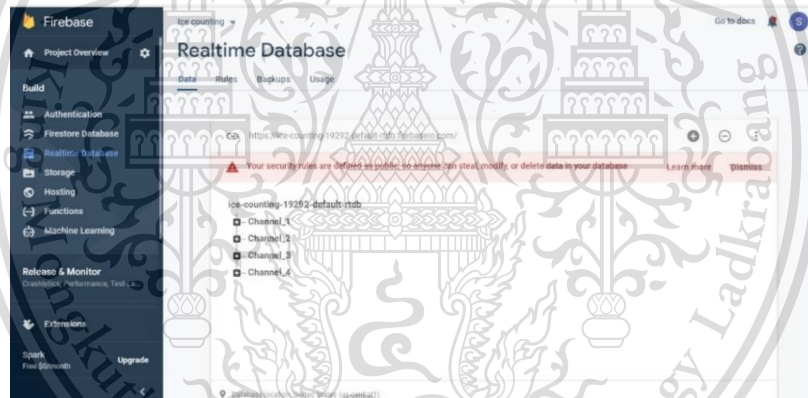
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

จากตารางที่ 4.2 และ 4.3 จะได้ค่าเฉลี่ยความแม่นยำในเวลากลางวันและกลางคืน 96.90% และ 93.88% ตามลำดับ แสดงให้เห็นว่าระบบมีการทำงานที่ดีกว่าในตอนกลางวัน ค่าเฉลี่ยความแม่นยำของ กล้องที่ 1 ถึง 4 เท่ากับ 98.31% 95.02% 100.00% และ 88.23% ตามลำดับ แสดงให้เห็นว่าระบบมีการ ทำงานที่ดีที่สุดในกล้องที่ 3 และแย่ที่สุดในกล้องที่ 4 เนื่องจากความคมชัดของภาพ รวมทั้งมุมกล้องที่ ต่างกันจากสภาพแสง และตำแหน่งในการติดตั้งกล้อง

#### 4.2 การแสดงผลแบบเรียลไทม์โดยใช้ Firebase

ข้อมูลและจำนวนตัวเลขของน้ำแข็งที่ละลายสำเร็จ จะสามารถตรวจสอบได้จากเว็บไซต์ Firebase โดยมีข้อมูลที่เชื่อมต่อได้แก่ จำนวนน้ำแข็ง เวลา วันที่ และจำนวนน้ำแข็งสะสม ซึ่งถูกบันทึกไว้ ขณะที่น้ำแข็งถูกละลายและออกจากพื้นที่อ้างอิงที่กำหนดไว้ โดยข้อมูลจะแบ่งเป็น 4 Channel ตาม จำนวนของกล้อง CCTV เช่น Channel\_1, Number=4, Time=Thursday 08 April 2021 14:37:20, Total=4 เป็นต้น



รูปที่ 4.4 Firebase แสดงข้อมูลของทั้ง 4 มุมกล้อง



รูปที่ 4.5 ตัวอย่างการเก็บข้อมูลของก้อนน้ำแข็งที่ละลายสำเร็จใน Firebase

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรืออาจใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น ถือว่าเป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารหรือข้อมูลใดๆที่มี การนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

## บทที่ 5

### สรุปและอภิปรายผล

ระบบการตรวจนับน้ำแข็งแบบเรียลไทม์ มีวัตถุประสงค์เพื่อสร้างระบบการตรวจนับน้ำแข็งของ ขณะลำเลียงผ่านรางลำเลียงขึ้นรถด้วยกล้องวงจรปิดซึ่งจะจัดเก็บเป็นข้อมูลตัวเลขในฐานข้อมูลแบบ เรียลไทม์ผ่าน Firebase โดยเริ่มจากการเตรียมชุดข้อมูล ได้แก่ การรวบรวมข้อมูลจากคลิปวิดีโอตัวอย่าง การทำงาน การทำ Data Cleaning, Data Annotation และการแปลงข้อมูลให้อยู่ในรูปแบบ TFRecord หลังจากนั้นนำชุดข้อมูลที่ได้เป็นอินพุตสำหรับการฝึกฝนแบบจำลอง EfficientDet-D0 ซึ่งเป็นแบบจำลอง ที่ผ่านการฝึกฝนมาแล้วสำหรับการตรวจจับวัตถุผ่าน Tensorflow หลังจากนั้นนำแบบจำลองการตรวจจับ น้ำแข็งที่ได้มาสร้างระบบการตรวจจับน้ำแข็งแบบเรียลไทม์ โดยเริ่มจากการกำหนดจุดอ้างอิงของน้ำแข็ง การกำหนดบริเวณอ้างอิงสำหรับการนับและเขียนโปรแกรม Python เพื่อการตรวจนับ โดยในบทนี้จะ กล่าวถึงการสรุปผลการวิจัย ปัญหาและข้อเสนอแนะจากการดำเนินงานที่ผ่านมาของการพัฒนาระบบการ ตรวจนับน้ำแข็งของแบบเรียลไทม์ โดยแบ่งเป็นหัวข้อดังต่อไปนี้

#### 5.1 สรุปผลการวิจัย

#### 5.2 ปัญหาและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

ผลการวิจัยจากการดำเนินการวัดผลการนับจำนวนน้ำแข็งของจะแบ่งได้เป็น 2 ส่วน ได้แก่ ความ แม่นยำในการนับโดยแบ่งตามช่วงเวลาและมุมกล้องที่ต่างกัน

##### 5.1.1 ความแม่นยำในการนับแบ่งตามช่วงเวลา

จากการทดสอบพบว่าในช่วงเวลากลางวัน ผลของการนับจากมุมกล้องที่ 1 ถึง 4 มีค่าเฉลี่ยความ แม่นยำอยู่ที่ 97.43%, 96.30%, 100.00% และ 96.97% ตามลำดับ มีค่าเฉลี่ยเท่ากับ 96.90% และ ในช่วงเวลากลางคืนมีค่าเฉลี่ยความแม่นยำอยู่ที่ 99.19%, 96.83%, 100% และ 79.49% ตามลำดับ มีค่าเฉลี่ยเท่ากับ 93.88%

จากการวัดผลที่คำนวณออกมาแสดงให้เห็นว่าในช่วงเวลาในช่วงกลางวันระบบการตรวจนับจะมี ความแม่นยำมากกว่าในช่วงเวลากลางคืน เนื่องจากแสงมีผลกระทบต่อความแม่นยำในการตรวจจับน้ำแข็ง โดยในสภาพแวดล้อมที่มีแสงน้อยทำให้เห็นส่งผลให้ความแม่นยำในการตรวจจับวัตถุลดลงเนื่องจากระบุ ความเป็นน้ำแข็งได้ยากขึ้นทำให้ระบบนับผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

### 5.1.2 ความแม่นยำในการนับแบ่งตามมุกกล้อง

จากการทดสอบพบว่าในมุกกล้องที่ 1 ในช่วงเวลากลางวันและกลางคืนซึ่งมีค่าเฉลี่ยความแม่นยำอยู่ที่ 97.43% และ 99.19% ตามลำดับ เฉลี่ยเท่ากับ 98.31% มุกกล้องที่ 2 ในช่วงเวลากลางวันและกลางคืนซึ่งมีค่าเฉลี่ยความแม่นยำอยู่ที่ 93.21% และ 96.83% ตามลำดับ เฉลี่ยเท่ากับ 95.02% มุกกล้องที่ 3 ในช่วงเวลากลางวันและกลางคืนซึ่งมีค่าเฉลี่ยความแม่นยำอยู่ที่ 100%, 100% เฉลี่ยเท่ากับ 100% และมุกกล้องที่ 4 ในช่วงเวลากลางวันและกลางคืนซึ่งมีค่าเฉลี่ยความแม่นยำอยู่ที่ 96.97% และ 79.49% เฉลี่ยเท่ากับ 88.23%

จากการวัดผลที่คำนวณออกมาจากทั้ง 4 มุกกล้อง แสดงให้เห็นค่าความแม่นยำของระบบการตรวจนับที่ขึ้นอยู่กับมุกกล้อง โดยมุกกล้องที่ 3 ให้ค่าความแม่นยำสูงสุด เนื่องจากกล้องที่ 3 มีคุณภาพความชัดของวิดีโอที่สูงและมีการติดตั้งอยู่ในมุมที่ดีและมีระยะห่างที่เหมาะสมจากรางลำเลียง ส่งผลให้ระบบสามารถตรวจจับน้ำแข็งได้แม่นยำ และระบบมีการทำงานที่แย่ที่สุดในกล้องที่ 4 เนื่องจากมีคุณภาพความชัดวิดีโอที่ต่ำและติดตั้งอยู่ในมุมที่ไม่ดีและมีระยะที่ไกลจากรางลำเลียง ทำให้มองเห็นน้ำแข็งไม่ชัดส่งผลให้ระบบมีความแม่นยำในการตรวจจับที่ต่ำทำให้ระบบนับผิดพลาด

### 5.2 ปัญหาและข้อเสนอแนะ

1. การเก็บรวบรวมข้อมูลในการฝึกฝนแบบจำลองทางผู้วิจัยได้เก็บรวบรวมมาเพียง 4 มุกกล้องที่ถูกตั้งไว้คงที่ ทำให้ข้อมูลไม่หลากหลายมากพอ ถ้าหากมีการนำระบบตรวจนับน้ำแข็งไปประยุกต์ใช้กับมุกกล้องที่แตกต่างออกไปอาจส่งผลให้ความแม่นยำมีค่าลดลง โดยทางผู้วิจัยมีข้อเสนอแนะโดยการเก็บรวบรวมข้อมูลที่หลากหลายมากกว่านี้ หรือใช้เทคนิคการทำ Data Augmentation มาประยุกต์ใช้
2. การใช้อัลกอริทึมตรวจจับวัตถุเพียงอย่างเดียวในระบบการตรวจนับอาจส่งผลให้การนับผิดพลาดได้ เนื่องจากความไม่ต่อเนื่องของเฟรมภาพ โดยเฉพาะกล้องที่มีอัตราเฟรม (Fps) ที่ต่ำ ดังนั้นผู้วิจัยนำเสนอใช้อัลกอริทึมการติดตามวัตถุ (Object Tracking) เข้ามาประยุกต์ใช้ร่วมกับการตรวจจับวัตถุ ซึ่งจะช่วยให้การติดตามน้ำแข็งมีความต่อเนื่องมากขึ้นจะส่งผลให้การนับมีความแม่นยำมากขึ้น
3. ในการทดสอบความแม่นยำของระบบการตรวจนับวัตถุควรทดสอบกับจำนวนวิดีโอที่หลากหลายและมากขึ้นกว่านี้เพื่อให้ได้ผลการทดสอบที่น่าเชื่อถือมากกว่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

## เอกสารอ้างอิง

[1] ไกรศักดิ์ เกษร. 2564. The Art of Data Science. Computer Science and Information Technology Department, Faculty of Science, NU

[2] Zijie Jay Wang. “CNN Explainer” [Online]. Available : <https://poloclub.github.io/cnn-explainer>. (เมษายน, 2564)

[3] Natthawat Phongchit. “Convolutional Neural Network (CNN)” [Online]. Available : <https://medium.com/@natthawatphongchit>. (เมษายน, 2564)

[4] The MathWorks Authors. “What is Object Detection.” [Online]. Available : <https://www.mathworks.com/discovery/object-detection.html>. (เมษายน, 2564).

[5] Alakh Sethi. “Build your Own Object Detection Using Tensorflow API.” [Online]. Available : [www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/](http://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/). (เมษายน, 2563)

[6] The Tensorflow Authors. “Tensorflow Object Detection API (readme).” [Online]. Available : [https://github.com/TensorFlow/models/tree/master/research/object\\_detection](https://github.com/TensorFlow/models/tree/master/research/object_detection) (เมษายน, 2564)

[7] Dipanjan (DJ) Sarkar. “A comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning.” [Online]. Available : <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>. (พฤศจิกายน, 2561)

[8] Robert (Munro) Monarch. “Active Transfer Learning with Pytorch.” [Online]. Available : <https://medium.com/pytorch/active-transfer-learning-with-pytorch71ed889f08c1>. (มกราคม, 2563).

[9] Rathod V & Wu N. “Tensorflow detection model zoo (documentation).” [Online]. Available : [https://github.com/tensorflow/models/blob/master/research/object\\_detection/g3doc/tf2\\_detection\\_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md). 2021.

[10] Mingxing Tan Ruoming Pang Quoc V. Le Google Research, Brain Team. “EfficientDet: Scalable and Efficient Object Detection.” arXiv:1911.09070v7, 2020. pp. 1-7.

[11] Guido van Rossum. “General Python FAQ” [Online]. Available : <https://docs.python.org/3/faq/general.html>. (เมษายน, 2564)

[12] Dimitri van Heesch. “OpenCV modules.” [Online]. Available : <https://docs.opencv.org/master>. (เมษายน, 2564)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และห้ามเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์  
ไม่ว่ากรณีใดๆก็ตาม กรุณาอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

[13] Banyapon Poolsawas. “Measuring Size” [Online]. Available : [www.daydev.com/machine-learning/machine-learning-opencv-python-measuring-size-object.html](http://www.daydev.com/machine-learning/machine-learning-opencv-python-measuring-size-object.html). (มิถุนายน, 2563)

[14] Pete Warden. “Classify Images with Tensorflow” [Online]. Available : [https://ai.googleblog.com/2015/12/how-to-classify-images-with-tensor flow.html](https://ai.googleblog.com/2015/12/how-to-classify-images-with-tensor-flow.html). (ธันวาคม, 2558)

[15] Sirawit Sek. “Firebase Early 2019” [Online]. Available : <https://medium.com/@sirawit>. (มกราคม, 2562)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.