



การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุ
ประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก

Development of an Automatic Approach to Create a High-Performance
Object Detection Model from Small Datasets

ธีรธร รักษาเมือง

สุปวีณ์ สัจจกร

ชญาดา เอียดทองใส

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์

สำนักวิชาวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยวลัยลักษณ์

ปีการศึกษา 2567

การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุ
ประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก

Development of an Automatic Approach to Create a High-Performance Object
Detection Model from Small Datasets

ธีรธร รักษาเมือง
สุปวีณ์ สัจจร
ชญาดา เอียดทองใส

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์
สำนักวิชาวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยวลัยลักษณ์
ปีการศึกษา 2567

ชื่อเรื่อง	การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุ ประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก
ชื่อผู้พัฒนา	ธีรธร รักษาเมือง สุวิทย์ สัจจกร ชญาดา เอียดทองใส
อาจารย์ที่ปรึกษา	รศ.ดร. วัฒนพงศ์ เกิดทองมี
ปริญญาและสาขาวิชา	หลักสูตรวิศวกรรมศาสตรและเทคโนโลยี สาขาวิชาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์
ปีการศึกษา	2567

บทคัดย่อ

รายงานวิจัยนี้มีวัตถุประสงค์เพื่อการพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุ
ประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก มีเป้าหมายเพื่อแก้ไขข้อจำกัดของการใช้ชุดข้อมูลขนาดใหญ่ โดย
โครงการนี้มุ่งเน้นการออกแบบและพัฒนาเฟรมเวิร์คที่สามารถสร้างชุดข้อมูลจำลอง (Synthetic Dataset)
โดยอัตโนมัติ การประยุกต์ใช้เทคนิคการเพิ่มข้อมูล (Data Augmentation) และการฝึกโมเดลโดยใช้ YOLOv8
ที่สามารถตรวจจับวัตถุได้อย่างแม่นยำและรวดเร็วในสถานการณ์ที่มีข้อมูลจำกัด นอกจากนี้ ยังได้พัฒนา
เครื่องมือที่ช่วยให้กระบวนการสร้างข้อมูลและฝึกโมเดลเป็นไปอย่างอัตโนมัติ โดยวัดผลสำเร็จจากค่าความ
แม่นยำเฉลี่ย (mAP), Precision, และ Recall ผลลัพธ์ที่ได้ช่วยลดต้นทุนและเพิ่มความสะดวกในการพัฒนา
โมเดล ซึ่งเป็นประโยชน์ต่อการนำไปใช้ในงานที่ต้องการความแม่นยำสูงและมีข้อจำกัดด้านทรัพยากร

คำสำคัญ: การตรวจจับวัตถุ, YOLOv8, ชุดข้อมูลขนาดเล็ก, ชุดข้อมูลจำลอง, การเพิ่มข้อมูล

Title	Development of an Automatic Approach to Create a High-Performance Object Detection Model from Small Datasets
Student's Name	Teeratorn Raksamuang Supavee Sonjohn Chayada Iadthongsai
Advisor	Assoc. Prof. Dr. Watthanapong Kerdthongmee
Degree and Program	Bachelor of Engineering Program in Computer Engineering and Artificial Intelligence
Academic Year	2024

Abstract

This research aims to develop an automated approach for creating high-performance object detection models from small datasets. The objective is to address the limitations of using large datasets. The project focuses on designing and developing a framework capable of automatically generating synthetic datasets, applying data augmentation techniques, and training models using YOLOv8 to achieve accurate and efficient object detection in scenarios with limited data. Additionally, tools were developed to automate data generation and model training processes. The success of the model was evaluated using metrics such as mean Average Precision (mAP), Precision, and Recall. The results demonstrate the model's ability to perform effectively in terms of accuracy and speed, reducing development costs and improving convenience. This work is beneficial for applications requiring high precision while operating under resource constraints.

Keywords: Object detection, YOLOv8, small datasets, synthetic datasets, data augmentation

กิตติกรรมประกาศ

รายงานวิจัยฉบับนี้สำเร็จลุล่วงได้ด้วยความรู้และความเอาใจใส่อย่างยิ่งจาก รศ.ดร. วัฒนพงศ์ เกิดทองมี อาจารย์ที่ปรึกษา ผู้ซึ่งให้คำแนะนำปรึกษาตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่างๆ รวมถึงการตรวจสอบและปรับปรุงแก้ไขในทุกขั้นตอน ด้วยความมุ่งมั่นและใส่ใจในทุกขั้นตอนของการดำเนินงาน คณะผู้พัฒนาขอแสดงความขอบคุณอย่างสูงไว้ ณ โอกาสนี้

คณะผู้พัฒนายังขอแสดงความขอบคุณไปยังทีมพัฒนาแบบจำลอง YOLOv8 และ Google Inc. ที่ได้จัดเตรียมแหล่งข้อมูลและเครื่องมือที่มีประโยชน์อย่างยิ่งต่อการศึกษาวิจัยครั้งนี้ ทั้งในด้านการค้นคว้า การพัฒนา และการประยุกต์ใช้งาน ทำให้การดำเนินงานบรรลุเป้าหมายได้อย่างมีประสิทธิภาพ

สุดท้ายนี้ คณะผู้พัฒนาหวังเป็นอย่างยิ่งว่ารายงานวิจัยฉบับนี้จะเป็นประโยชน์แก่ผู้ที่สนใจในหัวข้อดังกล่าว และหากมีข้อบกพร่องใด ๆ คณะผู้พัฒนาขอน้อมรับคำแนะนำเพื่อปรับปรุงและพัฒนางานในอนาคตให้ดียิ่งขึ้นต่อไป

คณะผู้พัฒนา
ธีรธร รักษาเมือง
สุปวีณ์ สัจจกร
ชญาดา เอียดทองใส

สารบัญ

หน้า

บทคัดย่อ.....	ผิดพลาด! ไม่ได้กำหนดบุ๊กมาร์ก
บทคัดย่อ อังกฤษ.....	ผิดพลาด! ไม่ได้กำหนดบุ๊กมาร์ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	ผิดพลาด! ไม่ได้กำหนดบุ๊กมาร์ก
สารบัญภาพ	จ
บทที่ 1	1
1.1 ความสำคัญและที่มาของปัญหา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของงาน.....	1
1.3.1 การเก็บข้อมูลและจัดการข้อมูล	1
1.3.2 การพัฒนาและฝึกอบรมโมเดล	2
1.3.3 การพัฒนาระบบอัตโนมัติ.....	2
1.3.4 การประเมินผลและปรับปรุงระบบ	2
1.3.5 ข้อจำกัดของโครงการ	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 ผลที่คาดว่าจะได้รับเมื่อเสร็จสิ้นโครงการ	3
1.6 ขั้นตอนการดำเนินงาน	4
1.7 อุปกรณ์ที่ใช้ในการพัฒนางาน.....	4
1.7.1 ด้านซอฟต์แวร์ (Software)	4
1.7.2 ด้านฮาร์ดแวร์ (Hardware)	5
บทที่ 2	7
2.1 ทฤษฎีที่เกี่ยวข้อง	7
2.1.1 การตรวจจับวัตถุ (Object Detection).....	7
2.1.2 การเพิ่มข้อมูล (Data Augmentation).....	8
2.1.3 Mean Average Precision (mAP)	10

2.1.4.	การประมวลผลภาพ (Image Processing).....	12
2.2	เทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ.....	16
2.2.1.	RoboFlow	16
2.2.2.	opencv-python.....	17
2.2.3.	TensorFlow	18
2.3	งานวิจัยหรือระบบงานใกล้เคียง	20
2.3.1	YOLO (You Only Look Once).....	20
2.3.2	การพัฒนาการตรวจจับรอยเปื้อนบนเสื้อผ้า (Development of Stain Detection on Clothes).....	21
2.3.3	พฤติกรรมการกินถุงพลาสติกของเต่าทะเลและผลกระทบต่อสุขภาพ: กรณีศึกษามลภาวะพลาสติกในทะเลไทย (The Feeding Behavior of Sea Turtles on Plastic Bags and Its Health Impacts: A Case Study of Marine Plastic Pollution in Thailand)	22
2.4	เปรียบเทียบระบบงานใกล้เคียงกับระบบที่พัฒนา	22
บทที่ 3	24
3.1	การวิเคราะห์ระบบ (System Analysis).....	24
3.1.1.	ภาพรวมสถาปัตยกรรมระบบ (System Architecture Diagram)	24
3.1.2.	การศึกษาความต้องการของผู้ใช้งาน (User Requirement Analysis).....	25
3.2	การวิเคราะห์ระบบปัจจุบัน (Existing System Analysis)	30
3.3	การระบุปัญหาและโอกาสพัฒนา (Problem and Opportunity Identification)	31
บทที่ 4	33
4.1.	สถาปัตยกรรมของระบบ (System architecture).....	33
4.2.	การเก็บรวบรวมข้อมูล	34
4.2.1.	การถ่ายภาพฟีเจอร์ (ขวดพลาสติก).....	34
4.2.2.	การจัดการพื้นหลังของฟีเจอร์ (Feature Extraction).....	35
4.2.3.	การเก็บรวบรวมภาพพื้นหลัง	36
4.2.4.	การจัดเก็บข้อมูล	37
4.2.5.	การตรวจสอบคุณภาพข้อมูล	38
4.3.	การสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset).....	38

4.3.1.	การนำพีเจอร์และภาพพื้นหลังมาใช้งาน	38
4.3.2.	การตรวจจับพื้นที่น้ำ (Water Mask Generation).....	39
4.3.3.	การวางพีเจอร์บนพื้นที่น้ำ.....	39
4.3.4.	การสร้างภาพจำลอง.....	39
4.3.5.	การสร้าง Annotation.....	40
4.3.6.	การจัดเก็บข้อมูล	40
4.3.7.	การตรวจสอบและประเมินผล	41
4.4.	การอัปโหลดชุดข้อมูลเข้า Roboflow (สำหรับ YOLOv8).....	41
4.4.1.	การจัดเตรียมชุดข้อมูล	41
4.4.2.	การสร้างโปรเจกต์ใน Roboflow	42
4.4.3.	การอัปโหลดชุดข้อมูล.....	42
4.4.4.	การปรับแต่งชุดข้อมูล.....	42
4.4.5.	การส่งออกชุดข้อมูล	42
4.5.	การฝึกฝนแบบจำลอง YOLOv8	43
4.5.1.	การเตรียมสภาพแวดล้อม.....	43
4.5.2.	การตั้งค่าและการโหลดชุดข้อมูล	43
4.5.3.	กระบวนการฝึกโมเดล	43
4.5.4.	การประเมินผลลัพธ์ของโมเดล.....	43
4.5.5.	การปรับปรุงโมเดล (Fine-tuning)	44
4.5.6.	การบันทึกและส่งออกโมเดล.....	44
4.6.	การพัฒนาเฟรมเวิร์ค AI	44
4.6.1.	การ Automate การสร้าง Synthetic Dataset	44
4.6.2.	การ Automate Data Augmentation บน Roboflow	44
4.6.3.	การเชื่อมโยงกระบวนการแบบ Automate.....	45
4.6.4.	การจัดเก็บและจัดการข้อมูล.....	45
4.6.5.	การปรับปรุงเฟรมเวิร์คในอนาคต	45

สารบัญตาราง

ตาราง	หน้า
ตาราง 1 ตารางแสดงความต้องการของผู้ใช้	25
ตาราง 2 ตารางแสดงคุณลักษณะของ UC01.....	27
ตาราง 3 ตารางแสดงคุณลักษณะของ UC02.....	28
ตาราง 4 ตารางแสดงคุณลักษณะของ UC03.....	28
ตาราง 5 ตารางแสดงคุณลักษณะของ UC04:.....	29
ตาราง 6 ตารางแสดงคุณลักษณะของ UC05.....	29

สารบัญภาพ

ภาพ	หน้า
รูป 1 การทำ AUGMENTED IMAGES	10
รูป 2 SYSTEM ARCHITECTURE DIAGRAM ของทั้งระบบ.....	24
รูป 3 แผนผังสถาปัตยกรรมระบบ	33
รูป 4 ตัวอย่างภาพถ่ายขวดพลาสติกบนพื้นหลังสีเขียว	35
รูป 5 ตัวอย่างภาพฟีเจอร์ในรูปแบบไฟล์ PNG ที่มีพื้นหลังโปร่งใส	36
รูป 6 ตัวอย่างภาพพื้นหลังจากแหล่งข้อมูลออนไลน์	36
รูป 7 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ FEATURES/	37
รูป 8 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ BACKGROUNDS/.....	38
รูป 9 ตัวอย่างการค่าช่วงสี HSV ที่กำหนดเพื่อสร้าง MASK ของพื้นพื้นน้ำ	39
รูป 10 ตัวอย่างการวางฟีเจอร์บนพื้นพื้นน้ำโดยสุ่มขนาด พิกัด และมุมหมุน	40
รูป 11 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ SYNTHETIC_DATASET/.....	41

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในปัจจุบัน เทคโนโลยีการตรวจจับวัตถุ (Object Detection) ได้กลายมาเป็นเครื่องมือสำคัญในหลายอุตสาหกรรม เช่น ด้านความปลอดภัย ระบบยานพาหนะไร้คนขับ การแพทย์ การเฝ้าระวังพื้นที่ หรือแม้กระทั่งการพัฒนาหุ่นยนต์อัตโนมัติ การตรวจจับวัตถุช่วยให้ระบบสามารถจดจำและแยกแยะวัตถุในสภาพแวดล้อมต่างๆ ได้อย่างมีประสิทธิภาพ ซึ่งถือเป็นพื้นฐานสำคัญในการพัฒนาเทคโนโลยีและนวัตกรรมใหม่ๆ ที่สามารถสร้างผลกระทบอย่างมากต่อทั้งภาคธุรกิจและสังคม

การพัฒนาโมเดลตรวจจับวัตถุที่มีประสิทธิภาพสูงในปัจจุบันมักจะอาศัยชุดข้อมูลขนาดใหญ่ในการฝึกฝนโมเดล เพื่อให้ระบบสามารถเรียนรู้ลักษณะและคุณสมบัติต่างๆ ของวัตถุที่ต้องการตรวจจับได้อย่างแม่นยำ อย่างไรก็ตาม การรวบรวมและสร้างชุดข้อมูลขนาดใหญ่เหล่านี้กลับเป็นสิ่งที่ท้าทายและมีค่าใช้จ่ายสูง เนื่องจากต้องการทั้งทรัพยากรในด้านเวลา บุคลากร และการติดป้ายกำกับข้อมูลอย่างละเอียดถี่ถ้วน โดยเฉพาะในโดเมนที่ข้อมูลมีจำนวนจำกัดหรือการจัดเก็บข้อมูลทำได้ยาก เช่น ในด้านการแพทย์ การสำรวจอวกาศ หรือการวิจัยเฉพาะทาง การขาดแคลนข้อมูลส่งผลให้การพัฒนาโมเดลที่แม่นยำและเสถียรนั้นทำได้ยากลำบาก

จากปัญหาดังกล่าว แนวทางการพัฒนาโมเดลตรวจจับวัตถุที่สามารถทำงานได้ดีบนชุดข้อมูลขนาดเล็กจึงมีความสำคัญเพิ่มมากขึ้น การพัฒนาแนวทางอัตโนมัติ เพื่อสร้างโมเดลที่มีประสิทธิภาพสูงแม้ว่าจะมีชุดข้อมูลจำกัด จึงเป็นหัวข้อที่ได้รับความสนใจอย่างมากในวงการวิจัยและอุตสาหกรรม การพัฒนาแนวทางนี้จะช่วยแก้ไขข้อจำกัดของการใช้ชุดข้อมูลขนาดใหญ่ รวมถึงช่วยลดต้นทุนในการพัฒนาโมเดล ทำให้การนำเทคโนโลยีการตรวจจับวัตถุไปใช้งานในสาขาต่างๆ เป็นไปได้อย่างกว้างขวางและคุ้มค่ามากขึ้น

1.2 วัตถุประสงค์

- 1) พัฒนาแนวทางประยุกต์ที่เหมาะสมในการขยายจำนวนของชุดข้อมูลภาพ
- 2) สร้างโมเดลที่สามารถตรวจจับวัตถุได้อย่างแม่นยำอย่างน้อย 80% และมีความเร็วในการประมวลผล โดยใช้ข้อมูลที่มีจำนวนจำกัดในการฝึกโมเดล
- 3) เปรียบเทียบวิธีการเพิ่มปริมาณชุดข้อมูลภาพที่นำมาใช้ในการฝึกเพื่อให้ได้โมเดลที่มีประสิทธิภาพสูงสุด

1.3 ขอบเขตของงาน

1.3.1 การเก็บข้อมูลและการจัดการข้อมูล

- เก็บข้อมูลภาพขดพลาสดิกที่ถ่ายในบริบทที่ควบคุม เช่น พื้นหลังสีเขียว

- ใช้กระบวนการ Data Augmentation เช่น การหมุน การปรับขนาด และการย้ายตำแหน่ง เพื่อเพิ่มความหลากหลายของข้อมูล
- จัดเก็บข้อมูลที่แยกเป็น Training Set และ Test Set สำหรับการพัฒนาและทดสอบโมเดล
- ตรวจสอบและจัดเตรียมข้อมูล เช่น การสร้าง Mask ของพื้นทีน้ำ เพื่อให้เหมาะสมกับการสร้างชุดข้อมูลจำลอง (Synthetic Dataset)

1.3.2. การพัฒนาและฝึกอบรมโมเดล

- พัฒนาโมเดลตรวจจับวัตถุโดยใช้เทคนิค Deep Learning เช่น YOLO (You Only Look Once)
- สร้างชุดข้อมูลจำลอง (Synthetic Dataset) ที่มี Annotation พร้อมใช้งาน เพื่อฝึกโมเดลตรวจจับขวดพลาสติกบนพื้นทีน้ำ
- ใช้กระบวนการ Fine-tuning เพื่อปรับปรุงโมเดลให้เหมาะสมกับการตรวจจับวัตถุในชุดข้อมูลขนาดเล็ก
- ทดสอบและวิเคราะห์โมเดลด้วยเมตริก เช่น mAP (Mean Average Precision), Precision, และ Recall

1.3.3. การพัฒนาระบบอัตโนมัติ

- พัฒนาระบบการอัตโนมัติสำหรับการสร้างชุดข้อมูลจำลอง ที่สามารถวางพีเจอร์ขวดพลาสติกบนพื้นทีน้ำในภาพพื้นหลัง
- เตรียมระบบสำหรับการนำข้อมูลเข้าแพลตฟอร์ม RoboFlow เพื่อจัดการชุดข้อมูลและการฝึกโมเดล
- ออกแบบอินเทอร์เฟซหรือเครื่องมือสำหรับผู้ใช้งาน เพื่อช่วยให้สามารถดำเนินการสร้างชุดข้อมูลและใช้งานระบบได้สะดวก

1.3.4. การประเมินผลและปรับปรุงระบบ

- ประเมินความแม่นยำของโมเดลด้วยเมตริก เช่น mAP, Precision, Recall บนชุดข้อมูล Test Set
- เก็บข้อเสนอแนะและข้อคิดเห็นจากผู้ใช้งาน เพื่อนำมาปรับปรุงโมเดลและกระบวนการอัตโนมัติ
- อัปเดตและปรับปรุงโมเดลเพิ่มเติมเมื่อมีข้อมูลใหม่ เช่น การขยายขอบเขตของวัตถุที่ตรวจจับ หรือบริบทของภาพพื้นหลัง

1.3.5. ข้อจำกัดของโครงการ

- การพัฒนาเน้นการทำงานกับ ชุดข้อมูลขนาดเล็ก และ Synthetic Dataset อาจมีข้อจำกัดด้านความแม่นยำเมื่อเทียบกับชุดข้อมูลขนาดใหญ่
- ระบบต้องพึ่งพาคุณภาพของ Mask น้ำ และกระบวนการ Augmentation ซึ่งอาจส่งผลต่อความสมจริงของชุดข้อมูล
- ระบบต้องสามารถปรับตัวรองรับการอัปเดตในอนาคต เพื่อให้เพิ่มฟีเจอร์หรือปรับปรุงโมเดลได้ง่าย

1.4 ประโยชน์ที่คาดว่าจะได้รับ

การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุจากชุดข้อมูลขนาดเล็กนี้จะช่วยลดเวลาและความซับซ้อนในการสร้างโมเดล ทำให้ผู้พัฒนาสามารถมุ่งเน้นไปที่การปรับปรุงประสิทธิภาพของโมเดลได้อย่างเต็มที่ โดยไม่ต้องเสียเวลาทำซ้ำขั้นตอนเดิม

นอกจากนี้ โมเดลที่พัฒนาขึ้นจากชุดข้อมูลขนาดเล็กจะมีความสามารถในการใช้งานได้จริง แม้ว่าจะมีข้อมูลจำกัด เหมาะสำหรับองค์กรหรือบุคคลที่มีทรัพยากรน้อย โดยสามารถนำโมเดลไปประยุกต์ใช้ในงานที่ต้องการการตรวจจับวัตถุในเวลาที่รวดเร็วและมีต้นทุนต่ำ

การใช้ชุดข้อมูลขนาดเล็กยังทำให้การเก็บรวบรวมและการเตรียมข้อมูลมีความง่ายและรวดเร็วมากขึ้น ทำให้สามารถสร้างและปรับปรุงโมเดลได้อย่างต่อเนื่องและมีประสิทธิภาพ

ดังนั้น การพัฒนานี้จึงมีความสำคัญต่อการส่งเสริมการเติบโตในด้านเทคโนโลยีการตรวจจับวัตถุ และเปิดโอกาสใหม่ในการสร้างผลิตภัณฑ์ที่มีประสิทธิภาพสูงขึ้น

1.5 ผลที่คาดว่าจะได้รับเมื่อเสร็จสิ้นโครงการ

เมื่อโครงการการพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็กเสร็จสิ้นลง คาดว่าจะได้รับผลลัพธ์ที่มีความสำคัญหลายประการ ดังนี้:

1. แนวทางประยุกต์ที่เหมาะสมในการขยายจำนวนของชุดข้อมูลภาพ
2. โมเดลที่สามารถตรวจจับวัตถุได้อย่างแม่นยำอย่างน้อย 80% และมีความเร็วในการประมวลผล โดยใช้ข้อมูลที่มีจำนวนจำกัดในการฝึกโมเดล
3. วิธีการเพิ่มปริมาณชุดข้อมูลภาพที่นำมาใช้ในการฝึกโมเดล เพื่อให้ได้โมเดลที่มีประสิทธิภาพสูงสุด

ผลที่คาดว่าจะได้รับเหล่านี้จะทำให้โครงการมีผลกระทบเชิงบวกต่อการพัฒนาเทคโนโลยีและการดำเนินงานในภาคธุรกิจต่าง ๆ โดยเฉพาะในด้านการตรวจจับวัตถุที่ต้องการความแม่นยำและรวดเร็ว

1.6 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

ลำดับที่	แผนการดำเนินงาน	ตุลาคม				พฤศจิกายน				ธันวาคม				มกราคม				กุมภาพันธ์				มีนาคม				เมษายน				พฤษภาคม			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	เขียนข้อเสนอโครงการ																																
2	สอบหัวข้อเสนอโครงการ					^	^																										
3	วิเคราะห์และออกแบบระบบ																																
4	สร้าง/train model																																
4.1	data preprocessing																																
4.2	train และ ปรับปรุง hyperparameter																																
4.3	ทดสอบประสิทธิภาพของ model																																
4.4	ปรับปรุงประสิทธิภาพของ model																																
5	สอบกลางภาคโครงการ										^1																						
6	เตรียมความพร้อมก่อนทดสอบ ประสิทธิภาพของโครงการ																																
7	สอบความก้าวหน้าโครงการ																																
8	ทดสอบประสิทธิภาพของ โครงการ																																
9	ประเมินและสรุปผลการวิจัย																																

หมายเหตุ

หมายถึง ระยะเวลาแผนการดำเนินงาน

^ หมายถึง สอบข้อเสนองาน

^1 หมายถึง สอบกลางภาคโครงการ

^2 หมายถึง สอบความก้าวหน้าโครงการ

1.7 อุปกรณ์ที่ใช้ในการพัฒนางาน

การพัฒนา ระบบ ผู้พัฒนาได้เลือกใช้เครื่องมือที่หลากหลาย เพื่อการใช้งานที่หลากหลาย ครบถ้วนตามความต้องการซึ่งมีทั้งซอฟต์แวร์และฮาร์ดแวร์ดังนี้

1.7.1 ด้านซอฟต์แวร์ (Software)

ระบบปฏิบัติการ

- Windows 11

ซอฟต์แวร์สำหรับการพัฒนาระบบ

- Jupyter Notebook: ใช้พัฒนาและทดลองโมเดล Machine Learning รวมถึงจัดการข้อมูล (Data Preprocessing)

- **VS Code:** ใช้เขียนและแก้ไขโค้ด โดยเฉพาะโค้ด Python สำหรับการสร้างและจัดการ Synthetic Dataset
- **Google Colab:** ใช้สำหรับพัฒนาและทดสอบโมเดลโดยใช้ทรัพยากรประมวลผลบนคลาวด์ (GPU/TPU)
- **Roboflow:** แพลตฟอร์มที่ใช้จัดการชุดข้อมูล เช่น การทำ Augmentation, การจัดการ Annotation และการเชื่อมต่อกับโมเดล Machine Learning

ซอฟต์แวร์สำหรับการจัดการเอกสาร

- **Microsoft Word:** ใช้จัดทำเอกสารรายงาน ผลการศึกษา และบันทึกข้อมูลสำคัญของโครงการ เพื่อให้ได้เอกสารที่มีมาตรฐานและสามารถแก้ไขได้สะดวก
- **Microsoft Excel:** ใช้จัดการและวิเคราะห์ข้อมูลเบื้องต้น รวมถึงการเก็บผลการทดสอบโมเดลต่าง ๆ ในรูปแบบตาราง เพื่อความสะดวกในการเปรียบเทียบและสรุปผล

ไลบรารีและเฟรมเวิร์กสำหรับการพัฒนา

- **OpenCV:** ใช้จัดการภาพ เช่น การตรวจจับพื้นที่น้ำ (Mask Detection) และการสร้าง Synthetic Dataset
- **NumPy:** ใช้ในการประมวลผลข้อมูลเชิงตัวเลข
- **TensorFlow/Keras:** ใช้สำหรับพัฒนาและฝึกอบรมโมเดล Deep Learning
- **rembg:** ไลบรารีสำหรับลบพื้นหลังของภาพเพื่อตัดพีเจอร์ออกจากแบกกราวด์
- **Matplotlib/Seaborn:** ใช้สำหรับการแสดงผลข้อมูลและผลการวิเคราะห์ในรูปแบบกราฟ

1.7.2 ด้านฮาร์ดแวร์ (Hardware)

การพัฒนาระบบ ผู้พัฒนาได้ใช้เครื่องคอมพิวเตอร์สำหรับการพัฒนา ซึ่งคอมพิวเตอร์ผู้พัฒนาสามารถระบุคุณสมบัติทางฮาร์ดแวร์ได้ดังนี้

- MSI Bravo 17 A4DDK-077TH

- CPU: AMD Ryzen 7 4800H (2.90 GHz up to 4.20 GHz)
- GPU: AMD Radeon RX 5500M
- RAM: 8 GB DDR4 3200 MHz
- OS: Windows 10 Home (64 Bit)

- **Acer Aspire 7 A715-42G-R4BX**

- CPU: AMD Ryzen 5 5500U (2.10 GHz up to 4.00 GHz)
- GPU: NVIDIA GeForce GTX 1650 4 GB GDDR6
- RAM: 24 GB DDR4 3200 MHz
- OS: Windows 11 Home (64 Bit)

- **HP Victus 16-d1213TX**

- CPU: Intel Core i5-12500H (2.50 GHz up to 4.50 GHz)
- GPU: Nvidia GeForce RTX 3050 4GB GDDR6
- RAM: 8 GB DDR5 4800 MHz
- OS: Windows 11 Home 64bit

บทที่ 2

ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1. การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุ (Object Detection) เป็นเทคนิคในวิทยาการคอมพิวเตอร์ที่ผสมผสานการจำแนกภาพ (Image Classification) และการระบุตำแหน่งวัตถุ (Object Localization) เข้าด้วยกัน เพื่อระบุและหาตำแหน่งของวัตถุภายในภาพหรือวิดีโอ เทคนิคนี้ช่วยให้ระบบสามารถตรวจจับวัตถุหลายชนิดในภาพเดียวกัน กำหนดป้ายกำกับให้กับวัตถุเหล่านั้น และให้ข้อมูลตำแหน่งที่แน่นอนของวัตถุในภาพ

การตรวจจับวัตถุมีบทบาทสำคัญในงานด้านวิทยาการคอมพิวเตอร์หลายด้าน เช่น การใส่คำอธิบายภาพ (Image Annotation) การนับจำนวนยานพาหนะ การรู้จำกิจกรรม (Activity Recognition) การตรวจจับใบหน้า (Face Detection) และการรู้จำใบหน้า (Face Recognition)

วิธีการตรวจจับวัตถุสามารถแบ่งออกเป็นสองประเภทหลัก ได้แก่ วิธีที่ใช้เครือข่ายประสาทเทียม (Neural Network-Based) และวิธีที่ไม่ใช้เครือข่ายประสาทเทียม (Non-Neural Approaches)

สำหรับวิธีที่ไม่ใช้เครือข่ายประสาทเทียม จำเป็นต้องกำหนดคุณลักษณะของวัตถุล่วงหน้า เช่น การใช้คุณลักษณะ Haar ในกรอบการทำงานของ Viola-Jones หรือการใช้คุณลักษณะ Histogram of Oriented Gradients (HOG) ในทางกลับกัน วิธีที่ใช้เครือข่ายประสาทเทียมสามารถทำการตรวจจับวัตถุแบบ end-to-end โดยไม่ต้องกำหนดคุณลักษณะเฉพาะ และมักอิงกับเครือข่ายประสาทคอนโวลูชัน (Convolutional Neural Networks - CNN)

โมเดลการตรวจจับวัตถุที่ใช้เครือข่ายประสาทเทียมที่ได้รับความนิยม ได้แก่ R-CNN, Fast R-CNN, Faster R-CNN, YOLO (You Only Look Once), และ SSD (Single Shot MultiBox Detector) และการตรวจจับวัตถุยังมีการประยุกต์ใช้ในงานต่าง ๆ เช่น การติดตามวัตถุในวิดีโอ การวิเคราะห์พฤติกรรม และการนำไปใช้ในระบบยานยนต์อัตโนมัติ

การทำงานทั่วไปของการตรวจจับวัตถุมีดังนี้:

- ภาพอินพุต: กระบวนการตรวจจับวัตถุเริ่มต้นด้วยการวิเคราะห์ภาพหรือวิดีโอ

- การประมวลผลล่วงหน้า: ภาพจะได้รับการประมวลผลล่วงหน้าเพื่อให้แน่ใจว่ารูปแบบเหมาะสมกับแบบจำลองที่ใช้
- การสกัดคุณลักษณะ: แบบจำลอง CNN ถูกใช้เป็นตัวสกัดคุณลักษณะ โดยแบบจำลองจะรับผิดชอบในการแยกภาพออกเป็นส่วนๆ และดึงคุณลักษณะออกจากแต่ละส่วนเพื่อตรวจจ็บบรูปแบบของวัตถุต่างๆ
- การจำแนกประเภท: แต่ละส่วนของภาพจะถูกจำแนกประเภทตามคุณลักษณะที่สกัดออกมา งานการจำแนกประเภทจะดำเนินการโดยใช้ SVM หรือเครือข่ายประสาทเทียมอื่นๆ ที่คำนวณความน่าจะเป็นของแต่ละหมวดหมู่ที่ปรากฏในบริเวณนั้น
- การระบุตำแหน่ง: พร้อมกันกับกระบวนการจำแนกประเภท แบบจำลองจะกำหนดกรอบขอบเขตสำหรับวัตถุที่ตรวจพบแต่ละชิ้น ซึ่งเกี่ยวข้องกับการคำนวณพิกัดสำหรับกรอบที่ล้อมรอบวัตถุแต่ละชิ้น จึงสามารถระบุตำแหน่งภายในภาพได้อย่างแม่นยำ
- การระบุค่าที่ไม่ใช่ค่าสูงสุด: เมื่อแบบจำลองระบุกรอบขอบเขตหลายกรอบสำหรับวัตถุเดียวกัน การระบุค่าที่ไม่ใช่ค่าสูงสุดจะถูกใช้เพื่อจัดการกับการทับซ้อนเหล่านี้ เทคนิคนี้จะเก็บเฉพาะกรอบขอบเขตที่มีคะแนนความเชื่อมั่นสูงสุดเท่านั้น และจะลบกรอบขอบเขตอื่นๆ ที่ทับซ้อนกันออกไป
- ผลลัพธ์: กระบวนการสิ้นสุดลงด้วยภาพต้นฉบับที่ถูกทำเครื่องหมายด้วยกรอบขอบเขตและป้ายกำกับที่แสดงวัตถุที่ตรวจพบและหมวดหมู่ที่เกี่ยวข้อง

วิธีการเรียนรู้เชิงลึกสำหรับการตรวจจ็บบวัตถุ มีวิธีการตรวจจ็บบวัตถุอยู่ 2 ประเภทหลัก:

Two-stage Object Detection: จะแบ่งการทำงานเป็นสองขั้นตอน: ขั้นตอนแรกจะเสนอภูมิภาคที่เป็นตัวเลือก จากนั้นจึงจัดหมวดหมู่ภูมิภาคนั้น เครื่องตรวจจ็บบสองขั้นตอนบางประเภท ได้แก่ R-CNN, Fast R-CNN และ Fastw R-CNN

Single-Stage Detectors for Object Detection จะคาดการณ์กล่องขอบเขตและความน่าจะเป็นของคลาสสำหรับทุกพื้นที่ของภาพได้อย่างแม่นยำ เน้นที่การรวมงานการระบุตำแหน่งวัตถุและการจำแนกประเภทเข้าในเครือข่ายประสาทเทียมแบบผ่านครั้งเดียว

2.1.2. การเพิ่มข้อมูล (Data Augmentation)

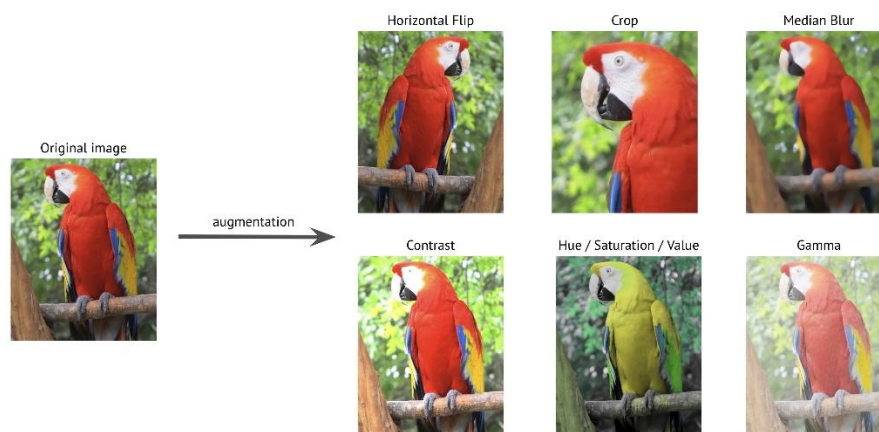
การทำ Augmented Images เพื่อที่จะปรับปรุงประสิทธิภาพของโมเดล ซึ่งเป็นขั้นตอนที่การ augmentations จะถูกนำไปใช้กับรูปภาพที่มีอยู่ในชุดข้อมูล กระบวนการนี้สามารถช่วยปรับปรุงความแม่นยำของโมเดล เมื่อโมเดลชุดข้อมูลรูปภาพใหม่ที่ไม่เคยเห็นโดยการ augmentations ภาพมีดังต่อไปนี้: พลิกกลับ หมุน 90 องศา หมุนแบบสุ่ม ครอบ

ตัดแบบสุ่ม การเฉือนแบบสุ่ม การเบลอ การเปิดรับแสง สัญญาณรบกวนแบบสุ่ม การตัดออก และโมเสก

รูปแบบการปรับแต่งข้อมูล (Augmentation Options)

- การพลิกภาพ (Flip): การพลิกภาพคือการสะท้อนภาพไปตามแกนเฉพาะเพื่อสร้างความหลากหลายในชุดข้อมูลและจำลองสถานการณ์ในโลกจริงที่วัตถุอาจสะท้อนกลับได้ มีสองประเภทหลัก ได้แก่ การพลิกแนวนอน ซึ่งจะพลิกภาพในทิศทางซ้าย/ขวา และการพลิกแนวตั้ง ซึ่งจะพลิกภาพในทิศทางขึ้น/ลง
- การหมุนแบบสุ่ม (Random Rotation): หมุนภาพรอบจุดศูนย์กลางด้วยมุมที่สุ่มเลือก โดยอาจหมุนตามหรือทวนเข็มนาฬิกา วิธีนี้ช่วยให้โมเดลมีความยืดหยุ่นต่อการเปลี่ยนแปลงทิศทางของวัตถุในภาพ
- การครอบแบบสุ่ม (Random Crop): ตัดหรือเลือกส่วนหนึ่งของภาพในขณะที่ละทิ้งส่วนที่เหลือ การครอบแบบสุ่มช่วยสร้างความหลากหลายในชุดข้อมูลโดยเน้นไปที่พื้นที่ต่างๆ ของภาพ และช่วยให้โมเดลเรียนรู้คุณลักษณะที่สำคัญได้ดียิ่งขึ้น
- การบิดแบบสุ่ม (Random Shear): บิดภาพในแนวนอนหรือแนวตั้งแบบสุ่ม เพื่อเพิ่มความหลากหลายในมุมมองของวัตถุ
- การปรับแสง (Exposure): เปลี่ยนค่าความสว่างหรือความมืดของภาพโดยการปรับค่าการรับแสง (gamma exposure)
- การเบลอ (Blur): เพิ่มเอฟเฟกต์เบลอให้กับภาพ เช่น การเบลอแบบ Gaussian เพื่อสร้างความสมจริงหรือจำลองการถ่ายภาพที่มีความไม่ชัดเจน
- การเพิ่มสัญญาณรบกวนแบบสุ่ม (Random Noise): เพิ่มสัญญาณรบกวนในภาพ เช่น สัญญาณรบกวนแบบเกลือและพริกไทย (salt and pepper noise) เพื่อจำลองภาพที่มีคุณภาพต่ำหรือมีความซับซ้อน
- การปรับแต่งกรอบคำอธิบายภาพ (Bounding Box Augmentation): ปรับเปลี่ยนกรอบคำอธิบายภาพ (bounding box) เพื่อให้เน้นเฉพาะเนื้อหาภายในกรอบข้อมูล วิธีนี้ช่วยสร้างชุดข้อมูลที่เหมาะสมสำหรับการเรียนรู้เชิงลึก

การใช้เทคนิคทั้งหมดนี้ช่วยเพิ่มความหลากหลายในชุดข้อมูล ทำให้โมเดลสามารถรับมือกับความแตกต่างของข้อมูลในโลกจริงได้ดียิ่งขึ้น



รูป 1 การทำ Augmented Images

https://albumentations.ai/docs/images/introduction/image_augmentation/augmentation.jpg

2.1.3. Mean Average Precision (mAP)

ความแม่นยำเฉลี่ยเฉลี่ย (Mean Average Precision หรือ mAP) คือตัวชี้วัดประสิทธิภาพที่ใช้ในการประเมินโมเดลการเรียนรู้ของเครื่อง โดยเป็นตัวชี้วัดที่ได้รับความนิยมมากที่สุดซึ่งใช้โดยความท้าทายด้านเกณฑ์มาตรฐานต่างๆ เช่น PASCAL VOC, COCO, ImageNET challenge, Google Open Image Challenge เป็นต้น ความแม่นยำเฉลี่ยเฉลี่ยมีความหมายต่างกันไปในแต่ละแพลตฟอร์ม ดังนั้น จึงมักจะทำให้สับสนได้

$$mAP = \frac{1}{N} \sum_i^N AP_i$$

โดยที่ mAP จะใช้เกณฑ์ย่อยดังต่อไปนี้

Confusion Matrix เป็นเครื่องมือที่ใช้ในการประเมินประสิทธิภาพของ Machine Learning โดยเฉพาะในงานจำแนกประเภท (Classification) เป็นวิธีการเปรียบเทียบระหว่างค่าที่ทำนาย (Predicted values) และค่าจริง (True values) โดยตารางที่แสดงจำนวนของการคาดการณ์ที่ถูกต้องและผิดพลาดในแต่ละคลาส (Class) ตารางนี้จะมีขนาด $n \times n$ โดยที่ n คือจำนวนคลาสทั้งหมดในข้อมูล และมี 4 ส่วนหลัก ๆ คือ:

- True Positive (TP) = สิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณี ทำนายว่าจริง และสิ่งที่เกิดขึ้น ก็คือ จริง
- True Negative (TN) = สิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้น ในกรณี ทำนายว่าไม่จริง และสิ่งที่เกิดขึ้น ก็คือ ไม่จริง
- False Positive (FP) = สิ่งที่ทำนายไม่ตรงกับสิ่งที่เกิดขึ้น คือทำนายว่า จริง แต่สิ่งที่เกิดขึ้น คือ ไม่จริง

- False Negative (FN) = สิ่งที่ทำนายไม่ตรงกับที่เกิดขึ้นจริง คือทำนายว่าไม่จริง แต่สิ่งที่เกิดขึ้น คือ จริง

Intersection Over Union (IoU) คือหน่วยวัดที่ใช้ประเมินความแม่นยำการตรวจจับวัตถุ โดยทั่วไป IoU จะใช้ร่วมกับเกณฑ์เพื่อพิจารณาว่าการตรวจจับนั้นถูกต้องหรือไม่ แบบจำลองการตรวจจับวัตถุโดยเปรียบเทียบกล่องขอบเขตที่คาดการณ์ไว้กับกล่องขอบเขตพื้นฐาน หน่วยวัดนี้จะวัดว่ากล่องทั้งสองทับซ้อนกันมากเพียงใดเป็นเศษส่วนของพื้นที่รวมของกล่องทั้งสอง ดังสมการที่ (2)

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2)$$

โดยที่:

Area of Overlap: พื้นที่ที่กรอบขอบเขตที่คาดการณ์ไว้และกรอบขอบเขตของข้อมูลพื้นฐานทับซ้อนกัน

Area of Union: พื้นที่ทั้งหมดที่ครอบคลุมโดยทั้งกรอบขอบเขตที่คาดการณ์ไว้และกรอบขอบเขตของข้อมูลพื้นฐาน ไม่รวมพื้นที่ทับซ้อน โดยจะสามารถแสดงเป็นสูตรทางคณิตศาสตร์ ดังสมการที่ (3)

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

โดยที่

A คือกรอบขอบเขตพื้นฐาน

B คือกรอบขอบเขตที่คาดการณ์ไว้

$|A \cap B|$ คือพื้นที่จุดตัดของทั้งสอง

$|A \cup B|$ คือพื้นที่การรวมตัวของทั้งสอง

Recall หรือ Sensitivity บอกถึง ความครอบคลุม ว่ากฎหรือตัวจำแนกสามารถตรวจพบกรณีบวกจริงได้มากน้อยเพียงใด มันมุ่งเน้นไปที่การลดจำนวน False Negative เพื่อไม่ให้พลาดกรณีสำคัญที่ควรจะถูกระบุ

ในด้าน Information Retrieval (IR) นั้น Recall มักไม่ได้รับความสำคัญมากนัก เนื่องจากสมมติฐานที่ว่าเอกสารที่เกี่ยวข้องมีอยู่จำนวนมาก และไม่สำคัญว่าเราจะค้นพบชุดย่อยใด นอกจากนี้ เรายังไม่สามารถทราบเกี่ยวกับความเกี่ยวข้องของเอกสารที่ไม่ได้ถูกดึงมาได้ ในด้าน Machine Learning และ Computational Linguistics Recall มักถูกละเลย

หรือนำมารวมเฉลี่ยไปกับปัจจัยอื่น ๆ โดยเน้นไปที่ความมั่นใจในกฎหรือตัวจำแนกเป็นหลัก
 ดังสมการที่ (4)

$$\text{Recall} = TP / RP = A / (A + C) \quad (4)$$

โดยที่

TP (True Positive): จำนวนกรณีบวกจริงที่ถูกทำนายถูกต้อง

RP (Real Positive): จำนวนกรณีทั้งหมดที่เป็นบวกจริง

A: กรณีที่เป็นบวกจริงและทำนายถูกต้อง

C: กรณีที่เป็นบวกจริงแต่ทำนายผิด (False Negative)

Precision หรือที่บางครั้งเรียกว่า Confidence (ในบริบทของ Data Mining) หมายถึง สัดส่วนของกรณีที่ทำนายว่าเป็นบวก (Predicted Positive) ที่เป็นบวกจริง (Real Positive)

ความสำคัญของ Precision ในด้าน Machine Learning, Data Mining และ Information Retrieval, Precision เป็นตัววัดที่สำคัญ เพราะแสดงให้เห็นว่าค่าทำนายบวกที่ทำไปนั้นมีความถูกต้องแม่นยำเพียงใด ในการระบุกรณีบวก โดยเป็นสัดส่วนระหว่างความสำเร็จ (True Positive) กับความพยายามทั้งหมด (Predicted Positive) อย่างไรก็ตาม ใน ROC Analysis, Precision มักไม่ได้รับการพิจารณาโดยตรง ดังสมการที่ (5)

$$\text{Precision} = TP / PP = A / (A + B) \quad (5)$$

โดยที่

TP (True Positive): จำนวนกรณีบวกจริงที่ถูกทำนายถูกต้อง

PP (Predicted Positive): จำนวนกรณีทั้งหมดที่ถูกทำนายว่าเป็นบวก

A: กรณีที่เป็นบวกจริงและทำนายถูกต้อง

B: กรณีที่ถูกทำนายว่าเป็นบวก แต่ไม่ใช่บวกจริง (False Positive)

2.1.4. การประมวลผลภาพ (Image Processing)

การประมวลผลภาพประกอบด้วยขั้นตอนหลักๆ ดังต่อไปนี้:

1. การนำเข้าภาพผ่านเครื่องมือรับภาพ
2. การวิเคราะห์และปรับแต่งภาพ
3. เอาต์พุตที่สามารถเปลี่ยนผลลัพธ์ได้ ภาพหรือรายงานที่อิงจากการวิเคราะห์ภาพนั้น

โดยที่ภาพจะถูกกำหนดให้เป็นฟังก์ชันสองมิติ $F(x,y)$ โดยที่ x และ y เป็นพิกัดเชิงพื้นที่ และแอมพลิจูดของ F ที่พิกัดใดๆ (x,y) เรียกว่าความเข้มของภาพนั้น ณ จุดนั้น เมื่อค่า x, y และแอมพลิจูดของ F มีจำกัด เราจะเรียกว่าภาพดิจิทัล กล่าวอีกนัยหนึ่ง ภาพสามารถกำหนดได้โดยอาร์เรย์สองมิติที่จัดเรียงเป็นแถวและคอลัมน์โดยเฉพาะ

ภาพดิจิทัลประกอบด้วยองค์ประกอบจำนวนจำกัด โดยแต่ละองค์ประกอบจะมีค่าเฉพาะที่ตำแหน่งเฉพาะ องค์ประกอบเหล่านี้เรียกว่าองค์ประกอบภาพ องค์ประกอบภาพและพิกเซล พิกเซลถูกใช้กันอย่างแพร่หลายที่สุดเพื่อระบุองค์ประกอบของภาพดิจิทัล

ประเภทของภาพ:

- ภาพไบนารี นั้นประกอบด้วยองค์ประกอบพิกเซลเพียงสององค์ประกอบ คือ 0 และ 1 โดย 0 หมายถึงสีดำ และ 1 หมายถึงสีขาว ภาพนี้เรียกอีกอย่างว่าภาพขาวดำ
- ภาพขาวดำ เป็นภาพที่ประกอบด้วยสีขาวดำเท่านั้นเรียกว่าภาพขาวดำ
- รูปแบบสี 8 บิต เป็นรูปแบบภาพที่มีชื่อเสียงที่สุด มีเฉดสีที่แตกต่างกัน 256 เฉด และเรียกอีกอย่างว่าภาพระดับสีเทา ในรูปแบบนี้ 0 หมายถึงสีดำ 255 หมายถึงสีขาว และ 127 หมายถึงสีเทา
- รูปแบบสี 16 บิต เป็นรูปแบบภาพสี มีสีที่แตกต่างกัน 65,536 สี เรียกอีกอย่างว่ารูปแบบสีสูง ในรูปแบบนี้ การกระจายสีจะไม่เหมือนกับภาพระดับสีเทา และรูปแบบ 16 บิตยังแบ่งออกเป็นรูปแบบเพิ่มเติมอีกสามรูปแบบ ได้แก่ สีแดง สีเขียว และสีน้ำเงิน ซึ่งเป็นรูปแบบ RGB

ขั้นตอนการประมวลผลภาพ:

1. การรับภาพ (Image Acquisition):

ขั้นตอนนี้เกี่ยวข้องกับการได้มาซึ่งภาพและการปรับให้เหมาะสมกับการประมวลผล โดยงานหลักประกอบด้วย:

- การปรับขนาดภาพให้เหมาะสมกับการใช้งาน
- การแปลงสี เช่น การแปลงภาพจาก RGB เป็นสีเทา หรือในทางกลับกัน

2. การปรับปรุงภาพ (Image Enhancement):

เป็นกระบวนการที่ง่ายและน่าสนใจที่สุดในงานประมวลผลภาพ ใช้เพื่อเพิ่มคุณภาพภาพหรือดึงรายละเอียดที่ซ่อนอยู่ในภาพ ซึ่งมักขึ้นอยู่กับความต้องการของผู้ใช้งาน

3. การฟื้นฟูภาพ (Image Restoration):

มุ่งเน้นการปรับปรุงภาพที่เสียหายหรือด้อยคุณภาพ โดยใช้แบบจำลองทางคณิตศาสตร์หรือความน่าจะเป็นเพื่อฟื้นฟูภาพให้ใกล้เคียงกับต้นฉบับมากที่สุด

4. การประมวลผลภาพสี (Color Image Processing):

เกี่ยวข้องกับการจัดการภาพสีและการปรับปรุงภาพที่มีสีเต็มรูปแบบหรือสีเทียม การใช้แบบจำลองสีช่วยในการปรับแต่งสีในภาพดิจิทัลได้อย่างมีประสิทธิภาพ

5. เวฟเล็ตและการประมวลผลความละเอียดหลายระดับ (Wavelets and Multiresolution Processing):

เป็นพื้นฐานในการแสดงภาพในหลายระดับความละเอียด ช่วยให้สามารถประมวลผลภาพในหลายสเกลได้อย่างละเอียด

6. การบีบอัดภาพ (Image Compression):

กระบวนการลดขนาดหรือความละเอียดของภาพ เพื่อประหยัดพื้นที่จัดเก็บและเพิ่มความเร็วในการส่งข้อมูล

7. การประมวลผลทางสัณฐานวิทยา (Morphological Processing):

ใช้สำหรับการแยกแยะและวิเคราะห์ส่วนประกอบในภาพ เช่น รูปร่าง โครงสร้าง หรือเส้นขอบ เพื่อช่วยอธิบายคุณลักษณะต่างๆ

8. การแบ่งส่วนภาพ (Image Segmentation):

เป็นกระบวนการแยกภาพออกเป็นส่วนต่างๆ หรือวัตถุที่สนใจ การแบ่งส่วนอัตโนมัติถือเป็นงานที่ซับซ้อนที่สุดในงานประมวลผลภาพ

9. การแสดงและคำอธิบาย (Representation and Description):

กระบวนการที่ต่อเนื่องจากการแบ่งส่วน ใช้เพื่อแปลงข้อมูลดิบที่ได้จากการแบ่งส่วนให้เป็นข้อมูลที่พร้อมใช้งาน โดยการเลือกวิธีการแสดงผลมีความสำคัญต่อการแปลความหมายของข้อมูล

10. การตรวจจับและการรับรู้วัตถุ (Object Detection and Recognition):

มุ่งเน้นการกำหนดป้ายกำกับให้กับวัตถุในภาพตามลักษณะหรือตัวอธิบายของวัตถุนั้น ช่วยให้ระบบสามารถระบุและจดจำวัตถุได้อย่างแม่นยำ

ข้อดีของการประมวลผลภาพดิจิทัล:

- คุณภาพของภาพที่ดีขึ้น: อัลกอริธึมการประมวลผลภาพดิจิทัลสามารถปรับปรุงคุณภาพของภาพของภาพ ทำให้ภาพชัดเจนขึ้น คมชัดขึ้น และให้ข้อมูลมากขึ้น
- งานที่ใช้ภาพอัตโนมัติ: การประมวลผลภาพดิจิทัลสามารถทำให้การทำงานที่ใช้ภาพหลายอย่างเป็นอัตโนมัติ เช่น การจดจำวัตถุ การตรวจจ็บบรูปแบบ และการวัด
- ประสิทธิภาพที่เพิ่มขึ้น: อัลกอริธึมการประมวลผลภาพดิจิทัลสามารถประมวลผลภาพได้เร็วกว่ามนุษย์มาก ทำให้สามารถวิเคราะห์ข้อมูลจำนวนมากได้ในเวลาอันสั้น
- ความแม่นยำที่เพิ่มขึ้น: อัลกอริธึมการประมวลผลภาพดิจิทัลสามารถให้ผลลัพธ์ที่แม่นยำกว่ามนุษย์ โดยเฉพาะสำหรับงานที่ต้องมีการวัดที่แม่นยำหรือการวิเคราะห์เชิงปริมาณ

ข้อเสียของการประมวลผลภาพดิจิทัล:

- ต้นทุนการคำนวณสูง: อัลกอริธึมการประมวลผลภาพดิจิทัลบางตัวต้องใช้การคำนวณจำนวนมากและต้องใช้ทรัพยากรการคำนวณจำนวนมาก
- ความสามารถในการตีความจำกัด: อัลกอริธึมการประมวลผลภาพดิจิทัลบางตัวอาจให้ผลลัพธ์ที่มนุษย์ตีความได้ยาก โดยเฉพาะอย่างยิ่งสำหรับอัลกอริธึมที่ซับซ้อนหรือล้ำสมัย การพึ่งพาคุณภาพของอินพุต: คุณภาพของเอาต์พุตของอัลกอริธึมการประมวลผลภาพดิจิทัลนั้นขึ้นอยู่กับคุณภาพของภาพอินพุตเป็นอย่างมาก ภาพอินพุตที่มีคุณภาพต่ำอาจส่งผลให้เอาต์พุตมีคุณภาพต่ำ
- ข้อจำกัดของอัลกอริธึม: อัลกอริธึมการประมวลผลภาพดิจิทัลมีข้อจำกัด เช่น ความยากลำบากในการจดจำวัตถุในฉากที่รกหรือมีแสงไม่เพียงพอ หรือไม่สามารถจดจำวัตถุที่มีการบิดเบือนหรือการบดบังอย่างมีนัยสำคัญ
- การพึ่งพาข้อมูลการฝึกอบรมที่ดี: ประสิทธิภาพของอัลกอริธึมการประมวลผลภาพดิจิทัลจำนวนมากขึ้นอยู่กับคุณภาพของข้อมูลการฝึกอบรมที่ใช้ในการพัฒนาอัลกอริธึม ข้อมูลการฝึกอบรมที่มีคุณภาพต่ำอาจส่งผลให้ประสิทธิภาพของอัลกอริธึมลดลง

2.2 เทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ

2.2.1. RoboFlow

RoboFlow เป็นเครื่องมือที่ออกแบบมาสำหรับนักพัฒนา เพื่อช่วยปรับปรุงกระบวนการสร้างและพัฒนาโมเดลคอมพิวเตอร์วิชัน (Computer Vision) ซึ่งกำลังได้รับความนิยมในหลากหลายอุตสาหกรรม ด้วยฟีเจอร์ที่ครอบคลุม RoboFlow ช่วยลดความซับซ้อนและเร่งกระบวนการพัฒนาโมเดลตั้งแต่ต้นจนจบ

จุดเด่นของ RoboFlow

- การจัดการชุดข้อมูลภาพ
- รองรับการอัปโหลด ใส่คำอธิบายประกอบ (Annotation) และประมวลผลชุดข้อมูลภาพล่วงหน้า
- ช่วยให้นักพัฒนาสามารถมุ่งเน้นไปที่การฝึกและปรับแต่งโมเดลโดยไม่เสียเวลาไปกับการเตรียมข้อมูล

คุณสมบัติหลักของ RoboFlow

- การเพิ่มข้อมูล (Data Augmentation): RoboFlow นำเสนอเทคนิคการเพิ่มข้อมูลที่หลากหลาย เช่น การหมุน (Rotation), การพลิก (Flip), และการปรับขนาด (Resizing) เพื่อสร้างความหลากหลายในชุดข้อมูลการฝึกอบรม และช่วยเพิ่มประสิทธิภาพของโมเดล
- การฝึกอบรมโมเดล (Model Training):
 - สามารถใช้โมเดลที่ผ่านการฝึกอบรมล่วงหน้า (Pre-trained Models)
 - ฝึกอบรมโมเดลเฉพาะของตนเองบนแพลตฟอร์ม RoboFlow
 - ปรับแต่งไฮเปอร์พารามิเตอร์และสถาปัตยกรรมโมเดลเพื่อเพิ่มความแม่นยำ

การนำไปใช้งาน (Deployment): RoboFlow รองรับการปรับใช้โมเดลที่ฝึกอบรมแล้วในแพลตฟอร์มต่างๆ เช่น:

- บริการคลาวด์ (Cloud Services): สำหรับแอปพลิเคชันออนไลน์
- อุปกรณ์เอดจ์ (Edge Devices): เช่น กล้อง IoT หรือระบบฝังตัว

ประโยชน์ของการใช้ RoboFlow

- ประหยัดเวลา: ลดเวลาในกระบวนการเตรียมข้อมูลและการตั้งค่า
- ความยืดหยุ่นสูง: รองรับการปรับแต่งโมเดลตามความต้องการ
- เพิ่มประสิทธิภาพของโมเดล: ด้วยเครื่องมือสำหรับการเพิ่มข้อมูลและการทดลองไฮเปอร์พารามิเตอร์

2.2.2. opencv-python

OpenCV (Open-Source Computer Vision Library) เป็นไลบรารีโอเพ่นซอร์สที่ได้รับความนิยมอย่างมากในด้านการประมวลผลภาพและการมองเห็นด้วยคอมพิวเตอร์ (Computer Vision) โดย OpenCV-Python เป็นตัวเชื่อม (binding) ที่ช่วยให้นักพัฒนาสามารถใช้ฟังก์ชันของ OpenCV ได้ง่ายผ่านภาษา Python โดยในปัจจุบัน Opencv มีบทบาทสำคัญในการทำงานแบบเรียลไทม์ ซึ่งมีความสำคัญมากในระบบปัจจุบัน โดยการใช้ Opencv จะทำให้สามารถประมวลผลรูปภาพและวิดีโอเพื่อระบุวัตถุ ใบหน้า หรือแม้แต่ลายมือของมนุษย์ได้

เมื่อรวมเข้ากับไลบรารีต่างๆ เช่น NumPy แล้ว Python จะสามารถประมวลผลโครงสร้างอาร์เรย์ของ Opencv เพื่อวิเคราะห์ได้ เพื่อระบุรูปแบบภาพและคุณลักษณะต่างๆ ของภาพ เราใช้เวกเตอร์สเปซและดำเนินการทางคณิตศาสตร์กับคุณลักษณะเหล่านี้

การใช้งาน OpenCV

คุณสมบัติที่สำคัญของ OpenCV-Python

การประมวลผลภาพพื้นฐาน:

- อ่าน/เขียนรูปภาพและวิดีโอ (cv2.imread, cv2.imwrite, cv2.VideoCapture)
- การแปลงสี เช่น จาก RGB เป็น Grayscale หรือ HSV (cv2.cvtColor)
- การครอบ (Crop), ย่อขยาย (Resize), และการหมุนภาพ (Rotation)

การตรวจจับและการรู้จำวัตถุ:

- การตรวจจับขอบ (Edge Detection) เช่น Canny Edge Detector
- การตรวจจับเส้น (Hough Line Detection)
- การตรวจจับใบหน้า (Face Detection) ด้วย Haar Cascade หรือ DNN models

ฟิลเตอร์และการประมวลผลเชิงพื้นที่:

- การปรับความคมชัด (Sharpening) และเบลอ (Blurring)
- การทำ Morphological Transformations เช่น Erosion, Dilation

การทำงานกับวิดีโอและการติดตามวัตถุ:

- การอ่านและเขียนวิดีโอ
- การติดตามวัตถุด้วยวิธี Optical Flow หรือ Tracking Algorithms

การรู้จำรูปแบบ (Pattern Recognition):

- การจดจำใบหน้า, ป้ายทะเบียนรถ

- การจับคู่รูปภาพ (Template Matching)

การประยุกต์ใช้ Deep Learning:

- รองรับการใช้งานโมเดลที่เทรนจาก TensorFlow, PyTorch, หรือ Caffe ผ่านโมดูล cv2.dnn

2.2.3. TensorFlow

TensorFlow เป็นกรอบงานสำหรับกำหนดและเรียกใช้การคำนวณที่เกี่ยวข้องกับเทนเซอร์ตามชื่อ เทนเซอร์เป็นการสรุปเวกเตอร์และเมทริกซ์ไปยังมิติที่สูงกว่า โดยภายใน TensorFlow แสดงเทนเซอร์เป็นอาร์เรย์ n มิติของประเภทข้อมูลพื้นฐาน แต่ละองค์ประกอบใน Tensor มีประเภทข้อมูลเดียวกัน และประเภทข้อมูลจะเป็นที่ทราบเสมอ รูปร่าง (นั่นคือ จำนวนมิติที่มีและขนาดของแต่ละมิติ) อาจเป็นที่รู้จักเพียงบางส่วนเท่านั้น การดำเนินการส่วนใหญ่สร้างเทนเซอร์ที่มีรูปร่างที่ทราบทั้งหมดหากรูปร่างของอินพุตเป็นที่รู้จักอย่างสมบูรณ์เช่นกัน แต่ในบางกรณี จะสามารถค้นหารูปร่างของเทนเซอร์ได้เฉพาะในเวลาที่ยดำเนินการกราฟเท่านั้น

คุณลักษณะของ TensorFlow

- การแยกความแตกต่างโดยอัตโนมัติ (AutoDifferentiation) TensorFlow มีพีเจอรการแยกความแตกต่างโดยอัตโนมัติ ซึ่งช่วยคำนวณเวกเตอร์การไล่ระดับ (gradient) ของโมเดลได้อย่างมีประสิทธิภาพ โดยเฉพาะในอัลกอริทึมอย่างแบ็กโพรพาเกชัน (Backpropagation) ที่ใช้ในการฝึกโมเดล เฟรมเวิร์กจะติดตามลำดับการดำเนินการของเทนเซอร์อินพุตและคำนวณการไล่ระดับตามพารามิเตอร์ที่เหมาะสม
- การดำเนินการอย่างกระตือรือร้น (Eager Execution) TensorFlow รองรับโหมดการดำเนินการอย่างกระตือรือร้น ซึ่งทำให้การดำเนินการสามารถประเมินผลได้ทันที โดยไม่ต้องสร้างกราฟการคำนวณล่วงหน้า การทำงานในโหมดนี้ช่วยให้โค้ดอ่านง่ายขึ้นและแก้ไขข้อผิดพลาดได้ง่ายขึ้น เนื่องจากสามารถตรวจสอบผลลัพธ์ทีละขั้นตอนผ่านดีบั๊กเกอร์ได้
- การกระจายการคำนวณ (Distribution) TensorFlow สนับสนุนการคำนวณแบบกระจายบนหลายอุปกรณ์ เช่น CPU, GPU หรือ TPU ผ่าน API และกลไกการกระจายที่หลากหลาย ซึ่งช่วยเพิ่มประสิทธิภาพในการฝึกอบรมและประเมินโมเดล โดยเฉพาะสำหรับโมเดลขนาดใหญ่

- ฟังก์ชันการสูญเสีย (Loss Functions) TensorFlow มีฟังก์ชันการสูญเสียหลากหลาย เช่น ข้อผิดพลาดกำลังสองเฉลี่ย (MSE) และเอนโทรปีแบบไขว้ (Cross Entropy) เพื่อใช้ประเมินผลลัพธ์ของโมเดลและช่วยในการปรับปรุงโมเดล
- เมตริก (Metrics) สำหรับการประเมินประสิทธิภาพของโมเดล TensorFlow มีเมตริกที่ใช้กันทั่วไป เช่น ความแม่นยำ (Accuracy), การเรียกคืน (Recall), ความแม่นยำ (Precision) และ Intersection-over-Union (IoU)
- TF.nn TensorFlow.nn เป็นโมดูลสำหรับดำเนินการเครือข่ายประสาทเทียม เช่น การคอนโวลูชัน (Convolution), ฟังก์ชันการเปิดใช้งาน (Activation Functions) เช่น ReLU, Softmax, Sigmoid และการดำเนินการอื่น ๆ เช่น Max-Pooling และ Bias-Addition
- ตัวเพิ่มประสิทธิภาพ (Optimizers) TensorFlow มีตัวเพิ่มประสิทธิภาพที่หลากหลาย เช่น Adam, Adagrad, และ Stochastic Gradient Descent (SGD) ซึ่งช่วยปรับค่าพารามิเตอร์ของโมเดลให้เหมาะสมเพื่อปรับปรุงการเรียนรู้และประสิทธิภาพของโมเดล

ขั้นตอนการใช้งาน TensorFlow

1. นำเข้าหรือสร้างข้อมูล ข้อมูลเป็นปัจจัยสำคัญสำหรับการเรียนรู้ของเครื่อง TensorFlow มีชุดข้อมูลที่พร้อมใช้งาน เช่น MNIST และ CIFAR-10 ซึ่งช่วยให้เริ่มต้นได้ง่ายขึ้น
2. แปลงและปรับมาตรฐานข้อมูล ก่อนใช้งาน ข้อมูลต้องถูกแปลงให้อยู่ในรูปแบบและประเภทที่อัลกอริทึม TensorFlow คาดหวัง และปรับมาตรฐานข้อมูล (Normalization) เพื่อให้เหมาะสมกับการฝึกโมเดล
3. ตั้งค่าพารามิเตอร์ของอัลกอริทึม กำหนดค่าคงที่ เช่น จำนวนการวนซ้ำ (Epochs), อัตราการเรียนรู้ (Learning Rate) และพารามิเตอร์อื่น ๆ ที่จำเป็น
4. กำหนดค่าตัวแปรและตัวแทน (Placeholders) TensorFlow ต้องการให้กำหนดค่าตัวแปร (Variables) และตัวแทน (Placeholders) เพื่อระบุว่าส่วนใดของข้อมูลสามารถปรับเปลี่ยนได้ระหว่างการฝึก
5. กำหนดโครงสร้างของโมเดล สร้างกราฟการคำนวณโดยกำหนดลำดับการดำเนินการและการเปลี่ยนแปลงของข้อมูลในโมเดล
6. ประกาศฟังก์ชันการสูญเสีย กำหนดฟังก์ชันการสูญเสียเพื่อตรวจสอบความคลาดเคลื่อนระหว่างค่าที่คาดการณ์และค่าจริง

7. เริ่มต้นและฝึกโมเดล เริ่มต้นกราฟการคำนวณ และป้อนข้อมูลเข้าสู่โมเดลเพื่อปรับค่าพารามิเตอร์และลดค่าฟังก์ชันการสูญเสีย
8. ประเมินโมเดล (Optional) ตรวจสอบประสิทธิภาพของโมเดลโดยใช้ข้อมูลใหม่และเมตริกที่กำหนดไว้
9. ทำนายผลลัพธ์ใหม่ (Optional) ใช้โมเดลที่ฝึกแล้วเพื่อทำนายผลลัพธ์สำหรับข้อมูลที่เคยเห็นมาก่อน

2.3 งานวิจัยหรือระบบงานใกล้เคียง

ในปัจจุบันมีงานวิจัยและซอฟต์แวร์ที่มุ่งเน้นการสร้างและฝึกอบรมโมเดลตรวจจับวัตถุที่มีประสิทธิภาพสูง แม้ว่างานส่วนใหญ่จะเน้นการใช้ชุดข้อมูลขนาดใหญ่ แต่งานวิจัยบางชิ้นยังมุ่งเน้นไปที่การพัฒนาโมเดลสำหรับชุดข้อมูลขนาดเล็ก เพื่อลดความจำเป็นในการใช้ทรัพยากรที่สูงในขั้นตอนการฝึกอบรม โดยระบบงานใกล้เคียงที่มีแนวคิดใกล้เคียงกับโครงการนี้ ได้แก่ YOLO (You Only Look Once), การพัฒนาการตรวจจับรอยเปื้อนบนเสื้อผ้า (Development of Stain Detection on Clothes), และ พฤติกรรมการกินถุงพลาสติกของเต่าทะเลและผลกระทบต่อสุขภาพ: กรณีศึกษา มลภาวะพลาสติกในทะเลไทย (The Feeding Behavior of Sea Turtles on Plastic Bags and Its Health Impacts: A Case Study of Marine Plastic Pollution in Thailand) ซึ่งเป็นตัวอย่างที่มีความสามารถสูงในการตรวจจับวัตถุ โดยมีรายละเอียดในข้อ 2.1, 2.2, และ 2.3 และการเปรียบเทียบระบบต่างๆ ไว้ในข้อ 2.4

2.3.1 YOLO (You Only Look Once)

YOLO เป็นโมเดลที่สามารถตรวจจับวัตถุได้แบบเรียลไทม์โดยใช้การเรียนรู้เชิงลึก (Deep Learning) การตรวจจับภาพด้วย YOLO ทำได้ในขั้นตอนเดียว โดยแบ่งภาพเป็นกริดแล้วคำนวณกรอบและคลาสของวัตถุ YOLO มีข้อได้เปรียบด้านความเร็วสูง แต่ยังต้องการชุดข้อมูลขนาดใหญ่ในการฝึกอบรมให้ได้ผลลัพธ์ที่แม่นยำ

(<https://pjreddie.com/darknet/yolo/>)

จุดเด่น

- สามารถตรวจจับวัตถุได้อย่างรวดเร็วและแม่นยำ
- สามารถตรวจจับวัตถุหลายชนิดในภาพเดียวได้
- เหมาะกับการใช้งานแบบเรียลไทม์

จุดด้อย

- ต้องใช้ชุดข้อมูลขนาดใหญ่ในการฝึกอบรม

- อาจมีปัญหาในการตรวจจับวัตถุที่มีขนาดเล็กหรืออยู่ใกล้กันมาก

2.3.2 การพัฒนาการตรวจจับรอยเปื้อนบนเสื้อผ้า (Development of Stain Detection on Clothes)

การเสริมข้อมูล (Data Augmentation) เป็นเทคนิคสำคัญที่ใช้เพิ่มประสิทธิภาพของโมเดลตรวจจับรอยเปื้อนบนเสื้อผ้า โดยการปรับเปลี่ยนข้อมูลภาพที่มีอยู่ เช่น การหมุนภาพ ปรับขนาด เปลี่ยนสี หรือเพิ่มสัญญาณรบกวน เพื่อเพิ่มจำนวนและความหลากหลายของชุดข้อมูล

(มหาวิทยาลัยวลัยลักษณ์)

จุดเด่น

- เพิ่มปริมาณข้อมูล: ช่วยเพิ่มจำนวนภาพรอยเปื้อนบนเสื้อผ้า ทำให้โมเดลเรียนรู้ได้ดีขึ้น
- เพิ่มความหลากหลาย: สร้างความหลากหลายของรอยเปื้อน เช่น ขนาด รูปร่าง และตำแหน่ง เพื่อให้โมเดลมีความยืดหยุ่นในการตรวจจับ
- ลดการเกิด Overfitting: ช่วยให้โมเดลไม่จดจำเฉพาะรูปแบบของข้อมูลที่มีอยู่ แต่สามารถทั่วไปกับข้อมูลใหม่ได้

จุดด้อย

- ความสมจริงของข้อมูล: การเสริมข้อมูลอาจไม่สามารถสร้างรอยเปื้อนที่มีความสมจริงเหมือนในชีวิตจริงได้ทั้งหมด
- ความซับซ้อนในการสร้างข้อมูล: การสร้างรอยเปื้อนที่เหมือนจริงต้องใช้ความรู้และเทคนิคที่ซับซ้อน
- ความเสี่ยงในการสร้างข้อมูลที่ไม่เหมาะสม: หากการเสริมข้อมูลไม่ถูกต้อง อาจทำให้โมเดลเรียนรู้ข้อมูลที่ไม่ถูกต้องและลดประสิทธิภาพ

2.3.3 พฤติกรรมการกินถุงพลาสติกของเต่าทะเลและผลกระทบต่อสุขภาพ: กรณีศึกษามลภาวะพลาสติกในทะเลไทย (The Feeding Behavior of Sea Turtles on Plastic Bags and Its Health Impacts: A Case Study of Marine Plastic Pollution in Thailand)

การพัฒนาชุดข้อมูลเทียม (Synthesize Dataset) สำหรับศึกษาพฤติกรรมของเต่าทะเลในการกินถุงพลาสติก มีบทบาทสำคัญในการสร้างข้อมูลที่หลากหลายและเหมือนจริง โดยเฉพาะเมื่อการเก็บข้อมูลภาคสนามมีข้อจำกัด ชุดข้อมูลนี้สามารถใช้ฝึกโมเดลการเรียนรู้เชิงลึก (Deep Learning) เพื่อตรวจจับหรือทำนายพฤติกรรมของเต่ารวมถึงการคาดการณ์ผลกระทบจากพลาสติกในระบบนิเวศ

(https://accstr.ufl.edu/wpcontent/uploads/sites/98/Pham_et_al_MarPolBull_2017.pdf)

จุดเด่น

- ลดข้อจำกัดของข้อมูลภาคสนาม
- เพิ่มความหลากหลายของข้อมูล
- สนับสนุนการพัฒนาระบบอัตโนมัติ
- เสริมการทดลองในหลากหลายบริบท
- ลดผลกระทบต่อสัตว์จริง

จุดด้อย

- ขาดความสมจริงในข้อมูล
- การสร้างข้อมูลที่มีคุณภาพต้องใช้ทรัพยากรสูง
- ความเสี่ยงของการแปลผลที่ไม่ถูกต้อง
- ข้อจำกัดของการประยุกต์ใช้

2.4 เปรียบเทียบระบบงานใกล้เคียงกับระบบที่พัฒนา

- ระบบงานใกล้เคียงยังไม่สามารถใช้ประโยชน์จากชุดข้อมูลขนาดเล็กได้อย่างมีประสิทธิภาพ เนื่องจากระบบส่วนใหญ่ต้องการชุดข้อมูลขนาดใหญ่ในการฝึกอบรม
- ระบบงานใกล้เคียงไม่สามารถสร้างขั้นตอนการฝึกอบรมที่เป็นอัตโนมัติเต็มรูปแบบ ซึ่งโครงการนี้พยายามลดขั้นตอนที่ซับซ้อนและเพิ่มความสะดวกในการสร้างโมเดล

- ระบบงานใกล้เคียงไม่มีเครื่องมือที่เน้นเฉพาะการทำงานกับชุดข้อมูลขนาดเล็ก ซึ่งโครงการนี้จะช่วยพัฒนาโมเดลที่สามารถฝึกอบรมได้บนชุดข้อมูลที่จำกัด

บทที่ 3

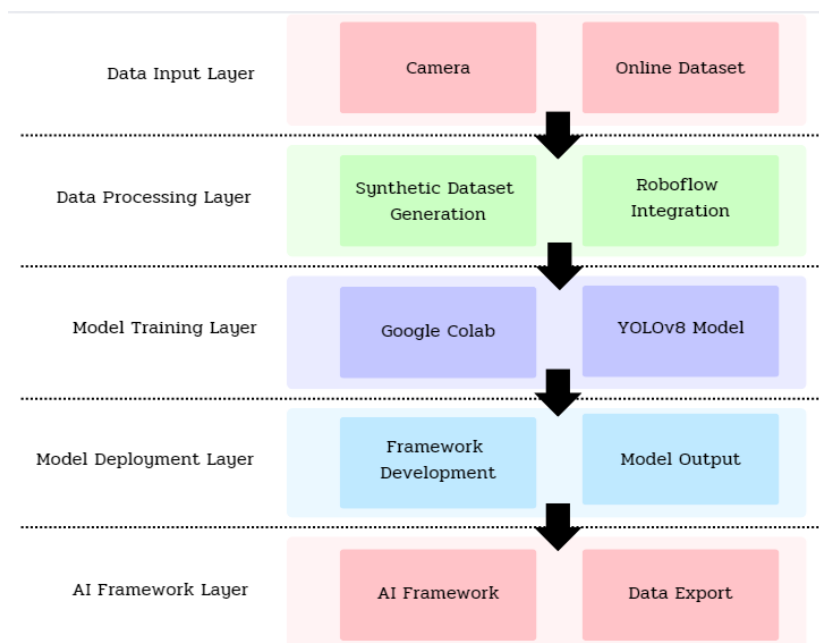
เอกสารความต้องการของระบบ

โครงการนี้มุ่งเน้นการพัฒนาโมเดลการตรวจจับวัตถุที่มีประสิทธิภาพสูงสำหรับการตรวจจับขวดพลาสติกบนพื้นที่น้ำ โดยใช้เทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) และการสร้างชุดข้อมูลจำลอง (Synthetic Dataset) เพื่อสนับสนุนการแก้ปัญหาพิษในแหล่งน้ำผ่าน การเก็บขวดพลาสติกโดยอัตโนมัติ นอกจากนี้ยังมีการพัฒนาระบบอัตโนมัติที่ช่วยให้ผู้ใช้งาน สามารถสร้างและจัดการข้อมูลได้สะดวก รวมถึงปรับปรุงกระบวนการฝึกโมเดลให้มีความแม่นยำ สูง

โครงการนี้มีเป้าหมายเพื่อสนับสนุนการจัดการปัญหาพิษในแหล่งน้ำอย่างยั่งยืน โดยเน้น การพัฒนาโมเดลที่ช่วยในการตรวจจับและแยกแยะขวดพลาสติกจากภาพถ่าย นำไปสู่การลด ปริมาณขยะพลาสติกในแหล่งน้ำโดยการใช้ระบบอัตโนมัติ สนับสนุนการป้องกันและแก้ปัญหา พิษในสิ่งแวดล้อมด้วยเทคโนโลยี Deep Learning สร้างแนวทางการใช้ปัญญาประดิษฐ์เพื่อ เพิ่มประสิทธิภาพในกระบวนการตรวจจับและเก็บขวดพลาสติก ส่งเสริมการประยุกต์ใช้ เทคโนโลยีปัญญาประดิษฐ์ในด้านสิ่งแวดล้อมอย่างมีประสิทธิภาพ

3.1 การวิเคราะห์ระบบ (System Analysis)

3.1.1. ภาพรวมสถาปัตยกรรมระบบ (System Architecture Diagram)



รูป 2 System Architecture Diagram ของทั้งระบบ

3.1.2. การศึกษาความต้องการของผู้ใช้งาน (User Requirement Analysis)

การเก็บรวบรวมข้อมูลจากกลุ่มผู้ใช้งานเป้าหมายเพื่อวิเคราะห์ความต้องการที่แท้จริงเป็นขั้นตอนสำคัญ ตารางที่ 3.1 คือตารางความต้องการของผู้ใช้ที่ต้องการให้มีการ พัฒนา มีรายละเอียดดังต่อไปนี้

ตารางที่ 3.1 ตารางแสดงความต้องการของผู้ใช้

ตาราง 1 ตารางแสดงความต้องการของผู้ใช้

ลำดับ	คำอธิบายความต้องการ	ความสำคัญ
Req 01	ระบบที่รองรับการตรวจจับวัตถุพลาสติกบนพื้นน้ำจากภาพถ่าย	Must have
Req 02	ระบบสามารถสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) พร้อมกับ Annotation ที่เหมาะสม	Must have
Req 03	ระบบต้องสามารถแสดงผลการตรวจจับ เช่น ตำแหน่งวัตถุพลาสติกบนพื้นน้ำในภาพถ่าย	Must have
Req 04	ระบบต้องรองรับการประเมินผลการทำงานของโมเดล ด้วยเมตริก เช่น Accuracy, Precision, Recall	Must have
Req 05	ระบบสามารถสร้าง Mask น้ำอัตโนมัติจากภาพพื้นหลัง	Must have
Req 06	ระบบต้องมีอินเทอร์เฟซที่ช่วยให้ผู้ใช้งานสามารถใช้งานได้ง่ายสะดวก	Could have

Req 07	ผู้ใช้งานสามารถปรับปรุงพารามิเตอร์สำหรับสร้าง Mask น้ำหรือ Annotation ด้วยตนเอง	Could have
Req 08	ระบบรองรับการส่งออกข้อมูล Annotation เพื่อใช้ร่วมกับแพลตฟอร์มภายนอก เช่น RoboFlow	Must have
Req 09	ระบบสามารถจัดการข้อมูล เช่น เพิ่ม ลบ หรือแก้ไขชุดข้อมูลภาพที่สร้างขึ้น	Must have
Req 10	โมเดลที่พัฒนาขึ้นต้องมีความแม่นยำไม่น้อยกว่า 80% และมีความยืดหยุ่นในการประยุกต์ใช้กับสถานการณ์ใหม่ ๆ	Must have
Req 11	ระบบสามารถปรับปรุงโมเดลเพิ่มเติมได้เมื่อมีข้อมูลใหม่เข้ามา	Should have
Req 12	ระบบสามารถประเมินข้อผิดพลาดของการตรวจจับและให้คำแนะนำสำหรับการปรับปรุงโมเดล	Could have

เมื่อได้ทำการรวบรวมและจัดลำดับความสำคัญของความต้องการจากผู้ใช้งานเรียบร้อยแล้ว ขั้นตอนถัดไปคือการตรวจสอบและวิเคราะห์เพื่อระบุฟังก์ชันที่เหมาะสมสำหรับตอบสนองความต้องการเหล่านั้น โดยให้ความสำคัญกับการพิจารณาฟังก์ชันที่สามารถแก้ไขปัญหาได้อย่างมีประสิทธิภาพ พร้อมทั้งกำหนดความเชื่อมโยงระหว่างฟังก์ชันที่ออกแบบและความต้องการของ

ผู้ใช้งานในแต่ละข้ออย่างชัดเจน เพื่อให้มั่นใจว่าฟังก์ชันเหล่านั้นสามารถตอบสนองความต้องการของผู้ใช้ได้อย่างครบถ้วนและสามารถตรวจสอบได้ตามมาตรฐานที่กำหนด

ตารางที่ 3.2 ตารางแสดงคุณลักษณะของ UC01: การสร้างชุดข้อมูลภาพจำลองและการจัดการข้อมูล (Synthetic Dataset and Data Management)

ตาราง 2 ตารางแสดงคุณลักษณะของ UC01

คุณลักษณะของ UC01: การสร้างชุดข้อมูลภาพจำลองและการจัดการข้อมูล (Synthetic Dataset and Data Management)		
คำอธิบาย: ผู้พัฒนาสามารถสร้างชุดข้อมูลภาพจำลอง พร้อม Annotation และสามารถจัดการชุดข้อมูลภาพที่สร้างขึ้นได้ เช่น เพิ่ม ลบ หรือแก้ไขภาพและ Annotation		
Actor: ผู้พัฒนา		
เงื่อนไข (ข้อกำหนด): <ol style="list-style-type: none"> ระบบสามารถสร้างชุดข้อมูลภาพจำลอง พร้อม Mask และ Annotation ได้อย่างถูกต้อง ระบบต้องมีอินเทอร์เฟซสำหรับจัดการข้อมูล เช่น เพิ่ม ลบ หรือแก้ไข 		
หมายเลข	คำอธิบายข้อกำหนดความต้องการ	ความสำคัญ (M,S,C,W)
Req 02	ระบบสามารถสร้างชุดข้อมูลภาพจำลอง พร้อม Annotation ที่เหมาะสม	M
Req 09	ระบบสามารถจัดการข้อมูล เช่น เพิ่ม ลบ หรือแก้ไขชุดข้อมูลภาพที่สร้างขึ้น	M

ตารางที่ 3.3 ตารางแสดงคุณลักษณะของ UC02: การตรวจจับขวดพลาสติกบนพื้นทีน้ำ (Plastic Bottle Detection)

ตาราง 3 ตารางแสดงคุณลักษณะของ UC02

คุณลักษณะของ UC02: การตรวจจับขวดพลาสติกบนพื้นทีน้ำ (Plastic Bottle Detection)		
คำอธิบาย: ระบบสามารถตรวจจับและแสดงตำแหน่งของขวดพลาสติกบนพื้นทีน้ำในภาพถ่าย		
Actor: ผู้พัฒนา, ผู้ใช้งาน		
เงื่อนไข (ข้อกำหนด): <ol style="list-style-type: none"> ระบบสามารถแสดงผลการตรวจจับตำแหน่งขวดพลาสติกบนภาพถ่าย ต้องรองรับการประเมินผลการทำงานของโมเดล 		
หมายเลข	คำอธิบายข้อกำหนดความต้องการ	ความสำคัญ (M,S,C,W)
Req 01	ระบบที่รองรับการตรวจจับขวดพลาสติกบนพื้นทีน้ำจากภาพถ่าย	M
Req 04	ระบบต้องรองรับการประเมินผลการทำงานของโมเดล เช่น Accuracy, Precision, Recall	M

ตารางที่ 3.4 ตารางแสดงคุณลักษณะของ UC03: การสร้าง Mask น้ำอัตโนมัติ (Water Mask Generation)

ตาราง 4 ตารางแสดงคุณลักษณะของ UC03

คุณลักษณะของ UC03: การสร้าง Mask น้ำอัตโนมัติ (Water Mask Generation)		
คำอธิบาย: ระบบสามารถสร้าง Mask น้ำอัตโนมัติจากภาพพื้นหลัง		
Actor: ผู้พัฒนา		
เงื่อนไข (ข้อกำหนด): <ol style="list-style-type: none"> ระบบสามารถปรับปรุงพารามิเตอร์สำหรับสร้าง Mask น้ำได้ Mask ต้องมีความแม่นยำและเหมาะสมกับการวางพีเจอร์ 		
หมายเลข	คำอธิบายข้อกำหนดความต้องการ	ความสำคัญ (M,S,C,W)

Req 05	ระบบสามารถสร้าง Mask น้ำอัตโนมัติจากภาพพื้นหลัง	M
Req 07	ผู้ใช้งานสามารถปรับปรุงพารามิเตอร์สำหรับสร้าง Mask น้ำได้ด้วยตนเอง	C

ตารางที่ 3.5 ตารางแสดงคุณลักษณะของ UC04: การส่งออก Annotation (Annotation Export)

ตาราง 5 ตารางแสดงคุณลักษณะของ UC04:

คุณลักษณะของ UC04: การส่งออก Annotation (Annotation Export)		
คำอธิบาย: ระบบรองรับการส่งออกข้อมูล Annotation เพื่อใช้ร่วมกับแพลตฟอร์มภายนอก เช่น RoboFlow		
Actor: ผู้พัฒนา		
เงื่อนไข (ข้อกำหนด): 1. Annotation ต้องเป็นฟอร์แมตที่รองรับโดย RoboFlow		
หมายเลข	คำอธิบายข้อกำหนดความต้องการ	ความสำคัญ (M,S,C,W)
Req 08	ระบบรองรับการส่งออกข้อมูล Annotation เพื่อใช้ร่วมกับแพลตฟอร์มภายนอก เช่น RoboFlow	M

ตารางที่ 3.6 ตารางแสดงคุณลักษณะของ UC05: การปรับปรุงและอัปเดตโมเดล (Model Improvement and Update)

ตาราง 6 ตารางแสดงคุณลักษณะของ UC05

คุณลักษณะของ UC05: การปรับปรุงและอัปเดตโมเดล (Model Improvement and Update)		
คำอธิบาย: ระบบต้องสามารถปรับปรุงและอัปเดตโมเดลเพิ่มเติมเมื่อมีข้อมูลใหม่เข้ามา		
Actor: ผู้พัฒนา		
เงื่อนไข (ข้อกำหนด): 1. โมเดลที่พัฒนาต้องมีความแม่นยำไม่น้อยกว่า 80% 2. ระบบรองรับการปรับปรุงโมเดลเพิ่มเติมเมื่อมีข้อมูลใหม่		

หมายเลข	คำอธิบายข้อกำหนดความต้องการ	ความสำคัญ (M,S,C,W)
Req 10	โมเดลที่พัฒนาขึ้นต้องมีความแม่นยำไม่น้อยกว่า 80%	M
Req 11	ระบบสามารถปรับปรุงโมเดลเพิ่มเติมได้เมื่อมีข้อมูลใหม่เข้า	S

3.2 การวิเคราะห์ระบบปัจจุบัน (Existing System Analysis)

ในปัจจุบัน ระบบที่เกี่ยวข้องกับการจัดการขยะในแหล่งน้ำ เช่น การตรวจจับและติดตามขยะลอยน้ำในแม่น้ำ ลำคลอง หรือแหล่งน้ำธรรมชาติ ยังขาดความแม่นยำและระบบอัตโนมัติที่สามารถใช้งานได้อย่างมีประสิทธิภาพ โดยทั่วไป การจัดการขยะลอยน้ำในปัจจุบันมักใช้วิธีการตรวจสอบด้วยตนเองหรืออุปกรณ์พื้นฐาน เช่น การเก็บขยะด้วยแรงงานคนหรือเรือเก็บขยะที่ควบคุมด้วยคน ซึ่งมีข้อจำกัดในด้านความแม่นยำ ประสิทธิภาพ และความครอบคลุมของการปฏิบัติงาน นอกจากนี้ การตรวจสอบด้วยตนเองยังไม่สามารถให้ข้อมูลเชิงลึกที่ช่วยสนับสนุนการตัดสินใจในการบริหารจัดการขยะอย่างมีประสิทธิภาพ

ในตลาดปัจจุบัน เทคโนโลยีการตรวจจับและติดตามวัตถุลอยน้ำเริ่มมีการพัฒนา เช่น การใช้โดรนหรืออุปกรณ์ติดกล้องเพื่อบันทึกภาพและส่งข้อมูล แต่ระบบเหล่านี้มักอิงกับการตรวจสอบด้วยคนเป็นหลัก และขาดการนำ AI เข้ามาประมวลผลข้อมูลเพื่อตรวจจับและจำแนกประเภทของวัตถุ เช่น ขวดพลาสติก กล่องโฟม หรือขยะอื่น ๆ ที่ลอยน้ำอย่างอัตโนมัติ ตัวอย่างเช่น โดรนตรวจสอบจาก DJI รุ่น Phantom 4 RTK สามารถบันทึกภาพแหล่งน้ำได้อย่างละเอียด แต่ไม่มีระบบปัญญาประดิษฐ์สำหรับการวิเคราะห์ภาพในเชิงลึก (DJI, n.d.).

อีกตัวอย่างหนึ่งคือระบบกล้องตรวจจับวัตถุจาก Axis Communications ที่ใช้ AI ในการตรวจจับและติดตามวัตถุในแหล่งน้ำ แต่ระบบดังกล่าวเน้นการใช้งานในอุตสาหกรรมหรือพื้นที่จำกัด และยังขาดความสามารถในการจัดการข้อมูลแบบอัตโนมัติหรือการแจ้งเตือนแบบเรียลไทม์ที่ปรับให้เหมาะกับแหล่งน้ำธรรมชาติ (Axis Communications, n.d.).

สุดท้ายคือระบบการจัดการข้อมูลจากบริษัทการจัดการขยะที่มีการใช้เทคโนโลยี Machine Learning สำหรับการคาดการณ์ปริมาณขยะ แต่ยังไม่มีการประยุกต์ใช้ในบริบทของการตรวจจับขยะลอยน้ำในพื้นที่แบบเฉพาะเจาะจง และขาดการบูรณาการข้อมูลเพื่อสนับสนุนการตัดสินใจเชิงลึกที่สามารถช่วยลดผลกระทบต่อสิ่งแวดล้อมได้อย่างแท้จริง

ดังนั้น การพัฒนาระบบตรวจจับขยะลอยน้ำด้วย AI ในโครงการนี้จึงมุ่งเน้นไปที่การเพิ่มความแม่นยำ ประสิทธิภาพ และความสามารถในการวิเคราะห์ข้อมูลเชิงลึก เพื่อช่วยสนับสนุนการจัดการขยะในแหล่งน้ำธรรมชาติได้อย่างยั่งยืน

3.3 การระบุปัญหาและโอกาสพัฒนา (Problem and Opportunity Identification)

ในปัจจุบัน ระบบจัดการขยะในแหล่งน้ำ เช่น แม่น้ำ ลำคลอง หรือแหล่งน้ำธรรมชาติ ยังมีข้อจำกัดหลายประการ ซึ่งส่วนใหญ่อาศัยแรงงานคนหรือการตรวจสอบด้วยวิธีการแบบดั้งเดิม ทำให้เกิดความล่าช้าและความไม่แม่นยำ นอกจากนี้ ระบบปัจจุบันยังขาดการวิเคราะห์ข้อมูลเชิงลึก เช่น การตรวจจับประเภทของขยะลอยน้ำหรือการแสดงผลข้อมูลที่ช่วยสนับสนุนการตัดสินใจในการจัดการขยะอย่างเหมาะสม ข้อมูลส่วนใหญ่ที่ได้จากการตรวจสอบเป็นเพียงภาพถ่ายหรือข้อมูลดิบ ซึ่งต้องใช้เวลามากในการประมวลผลและแปลผลด้วยตนเอง ส่งผลให้การจัดการขยะไม่สามารถดำเนินการได้อย่างเต็มประสิทธิภาพ และไม่สามารถลดผลกระทบต่อสิ่งแวดล้อมได้อย่างยั่งยืน

โอกาสในการพัฒนา:

1. การใช้ AI:

- การนำเทคโนโลยี AI เช่น Deep Learning มาประมวลผลข้อมูลภาพจากแหล่งน้ำ ช่วยเพิ่มความแม่นยำและความรวดเร็วในการตรวจจับและจำแนกประเภทของขยะลอยน้ำ เช่น ขวดพลาสติก กล่องโฟม หรือเศษวัสดุอื่น ๆ
- การประยุกต์ใช้ AI ยังสามารถช่วยคาดการณ์ปริมาณขยะในพื้นที่ที่กำหนด ซึ่งช่วยในการวางแผนการจัดการได้ดียิ่งขึ้น

2. Roboflow Integration:

- การใช้แพลตฟอร์มอย่าง Roboflow ในการจัดการและปรับปรุงคุณภาพของข้อมูลภาพ เช่น การทำ Data Augmentation เพื่อเพิ่มความหลากหลายของข้อมูลสำหรับการฝึกโมเดล
- Roboflow ยังช่วยให้สามารถจัดรูปแบบ Annotation และสร้าง Dataset ที่เหมาะสมสำหรับการพัฒนาโมเดล AI ได้อย่างสะดวก

3. การพัฒนา Framework สำหรับการจัดการข้อมูลและตรวจจับขยะลอยน้ำ:

- การสร้าง Framework สำหรับการจัดการตรวจจับขยะลอยน้ำที่สามารถปรับแต่งได้ เช่น การปรับพารามิเตอร์การตรวจจับและการประมวลผลข้อมูล
- Framework นี้จะช่วยให้ผู้ใช้งานสามารถนำไปปรับใช้ในบริบทอื่น ๆ ได้ เช่น การตรวจจับขยะในแหล่งน้ำต่าง ๆ หรือในภูมิภาคที่แตกต่างกัน
- การออกแบบ Framework ที่สามารถรองรับการขยายตัวในอนาคต เช่น การเพิ่มฟีเจอร์สำหรับการวิเคราะห์ข้อมูลเพิ่มเติม หรือการบูรณาการกับแพลตฟอร์มอื่น ๆ

4. การบูรณาการข้อมูล:

- ระบบที่รวมข้อมูลภาพ ข้อมูลการตรวจจับ และข้อมูลวิเคราะห์ไว้ในที่เดียว ช่วยให้สามารถวิเคราะห์ข้อมูลเชิงลึกและให้คำแนะนำที่เหมาะสมสำหรับการจัดการขยะ

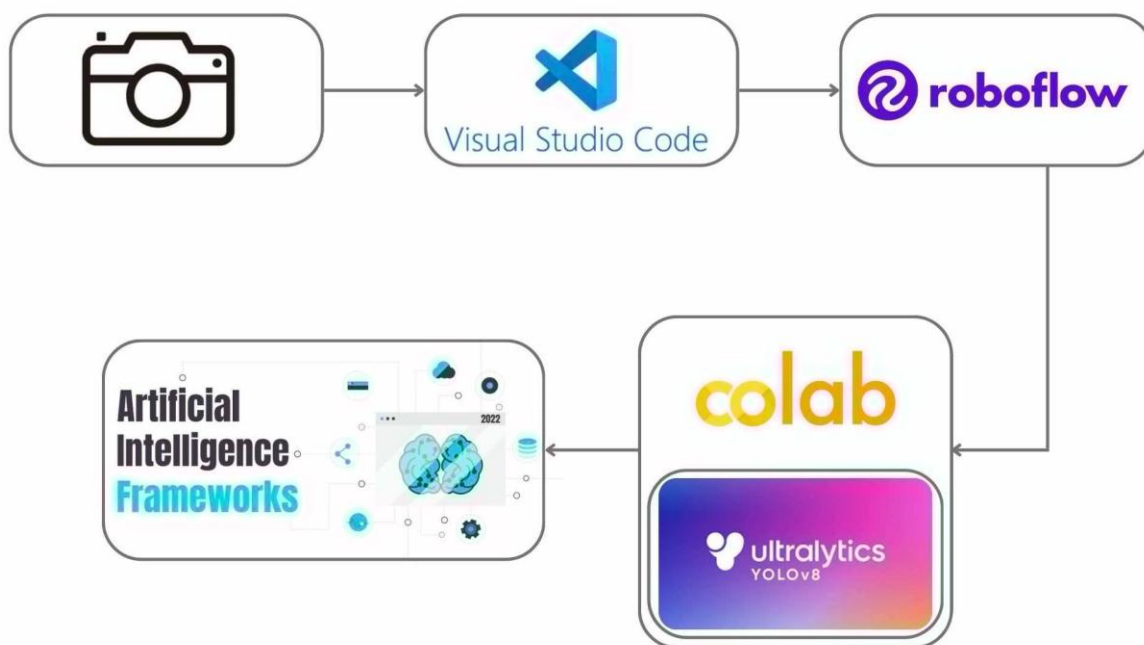
- การแสดงผลข้อมูลในรูปแบบรายงานหรือสถิติ ช่วยให้ผู้ใช้งานสามารถติดตามและประเมินผลการจัดการขยะได้ง่ายขึ้น

บทที่ 4

การออกแบบระบบ

4.1. สถาปัตยกรรมของระบบ (System architecture)

ในการศึกษาครั้งนี้ ผู้พัฒนาได้ออกแบบขอบเขตขั้นตอนการดำเนินงานไว้ดังต่อไปนี้ โดยแสดงตามผังสถาปัตยกรรมระบบ



ภาพ 4.1 ผังสถาปัตยกรรมระบบ

รูป 3 ผังสถาปัตยกรรมระบบ

ซึ่งสามารถแบ่งและอธิบายรายละเอียดขั้นตอนการทำงานได้ ดังนี้

1) การเก็บรวบรวมข้อมูล

ทำการเก็บรวบรวมข้อมูลจากการถ่ายภาพแหล่งน้ำที่มีขยะ เช่น ขวดพลาสติก และภาพพื้นหลังที่ไม่มีขวดพลาสติก รวมถึงการนำข้อมูลภาพจากแหล่งข้อมูลออนไลน์ที่เกี่ยวข้อง เช่น Roboflow เพื่อใช้เป็นชุดข้อมูลเริ่มต้นสำหรับการพัฒนาโมเดลตรวจจับ

2) การสร้างและจัดเตรียมชุดข้อมูล (Synthetic Dataset)

นำข้อมูลภาพพื้นหลังและพีเจอร์ (ขวดพลาสติก) มาสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) พร้อม Annotation โดยใช้เทคนิคการวางพีเจอร์บนภาพพื้นหลังแบบสุ่ม และปรับปรุงคุณภาพข้อมูลด้วยการสร้าง Mask น้ำ เพื่อกำหนดพื้นที่น้ำในภาพให้เหมาะสม

3) การจัดการข้อมูลและอัปโหลดเข้า Roboflow

นำชุดข้อมูลที่สร้างขึ้นไปอัปโหลดเข้าสู่แพลตฟอร์ม Roboflow เพื่อทำการจัดการและปรับปรุง Annotation และใช้พีเจอร์ Data Augmentation เพื่อเพิ่มความหลากหลายของชุดข้อมูล ชุดข้อมูลจะถูกส่งออกในฟอร์แมตที่เหมาะสมสำหรับการฝึกโมเดล YOLO

4) การฝึกฝนแบบจำลอง YOLO

นำชุดข้อมูลจาก Roboflow ไปฝึกโมเดลตรวจจับขวดพลาสติกโดยใช้ Ultralytics YOLOv8 บนแพลตฟอร์ม Google Colab เพื่อใช้ทรัพยากรประมวลผลจากคลาวด์ ในขั้นตอนนี้จะทำการประเมินและปรับปรุงโมเดลด้วยเมตริก เช่น Accuracy, Precision และ Recall เพื่อหาโมเดลที่มีประสิทธิภาพสูงสุด

5) การพัฒนาเฟรมเวิร์ค AI

เมื่อได้โมเดลที่มีคุณภาพสูงสุด ระบบจะถูกพัฒนาเป็นเฟรมเวิร์ค AI โดยใช้ Visual Studio Code เพื่อออกแบบโครงสร้างการทำงานและการใช้งาน เฟรมเวิร์คดังกล่าวจะรองรับการประมวลผลภาพ การตรวจจับขยะลอยน้ำ และการแสดงผลข้อมูลในรูปแบบที่เข้าใจง่าย

4.2. การเก็บรวบรวมข้อมูล

การเก็บรวบรวมข้อมูลในโครงการนี้มุ่งเน้นการเตรียมข้อมูลที่เหมาะสมสำหรับการพัฒนาโมเดลตรวจจับขวดพลาสติกบนพื้นน้ำ โดยมีการวางแผนขั้นตอนการดำเนินงานอย่างเป็นระบบ เพื่อให้ได้ข้อมูลที่มีคุณภาพสูงเพียงพอและครอบคลุมความหลากหลายของสถานการณ์จริงที่อาจเกิดขึ้นในแหล่งน้ำธรรมชาติ

4.2.1. การถ่ายภาพพีเจอร์ (ขวดพลาสติก)

กระบวนการถ่ายภาพพีเจอร์เริ่มจากการรวบรวมขวดพลาสติกหลากหลายประเภท เช่น ขวดใส ขวดที่มีฉลากสีต่าง ๆ และขวดที่มีรูปทรงหลากหลาย เพื่อให้ข้อมูลที่ครอบคลุมสถานการณ์ที่พบได้ในแหล่งน้ำจริง ภาพถ่ายจะถูกถ่ายบนพื้นหลังสีเขียวเรียบเพื่อช่วยให้การแยกพีเจอร์ออกจากพื้นหลังในขั้นตอนถัดไปทำได้ง่ายและแม่นยำ นอกจากนี้ ขวดพลาสติกแต่ละใบยังถูกถ่ายในมุมมองที่หลากหลาย ได้แก่ ด้านหน้า ด้านซ้าย ด้านขวา และด้านหลัง เพื่อเพิ่มความหลากหลายของชุดข้อมูลที่ใช้ในการฝึกโมเดล



ภาพ 4.2 ตัวอย่างภาพถ่ายขวดพลาสติกบนพื้นหลังสีเขียว

รูป 4 ตัวอย่างภาพถ่ายขวดพลาสติกบนพื้นหลังสีเขียว

4.2.2. การจัดการพื้นหลังของฟีเจอร์ (Feature Extraction)

หลังจากถ่ายภาพขวดพลาสติก ระบบจะทำการลบพื้นหลังสีเขียวออกโดยใช้สคริปต์ Python ที่พัฒนาขึ้นโดยใช้ไลบรารี OpenCV กระบวนการนี้ประกอบด้วยการใช้ค่าช่วงสี (HSV Range) เพื่อแยกพื้นหลังออกจากฟีเจอร์ และบันทึกฟีเจอร์ที่ได้ในรูปแบบไฟล์ PNG พร้อมพื้นหลังโปร่งใส สำหรับปรับปรุงคุณภาพของภาพจะมีการใช้ตัวกรอง Gaussian Blur เพื่อลบขอบที่ไม่สมบูรณ์ และ Sharpen Filter เพื่อเพิ่มความคมชัดของฟีเจอร์ก่อนบันทึก



ภาพ 4.3 ตัวอย่างภาพพีเจอร์ในรูปแบบไฟล์ PNG ที่มีพื้นหลังโปร่งใส

รูป 5 ตัวอย่างภาพพีเจอร์ในรูปแบบไฟล์ PNG ที่มีพื้นหลังโปร่งใส

4.2.3. การเก็บรวบรวมภาพพื้นหลัง

ภาพพื้นหลังสำหรับโครงการนี้ถูกถ่ายจากแหล่งน้ำจริง เช่น แม่น้ำ ลำคลอง และบ่อน้ำ โดยเลือกภาพที่ไม่มีขวดพลาสติกหรือวัตถุอื่นในเฟรม เพื่อใช้เป็นพื้นฐานในการสร้างชุดข้อมูลจำลอง นอกจากนี้ยังมีการนำภาพจากแหล่งข้อมูลออนไลน์ที่มีคุณภาพสูง เช่น เว็บไซต์ที่ให้บริการภาพฟรี เพื่อเพิ่มจำนวนและความหลากหลายของภาพพื้นหลังที่ใช้ในโครงการ



ภาพ 4.4 ตัวอย่างภาพพื้นหลังจากแหล่งข้อมูลออนไลน์

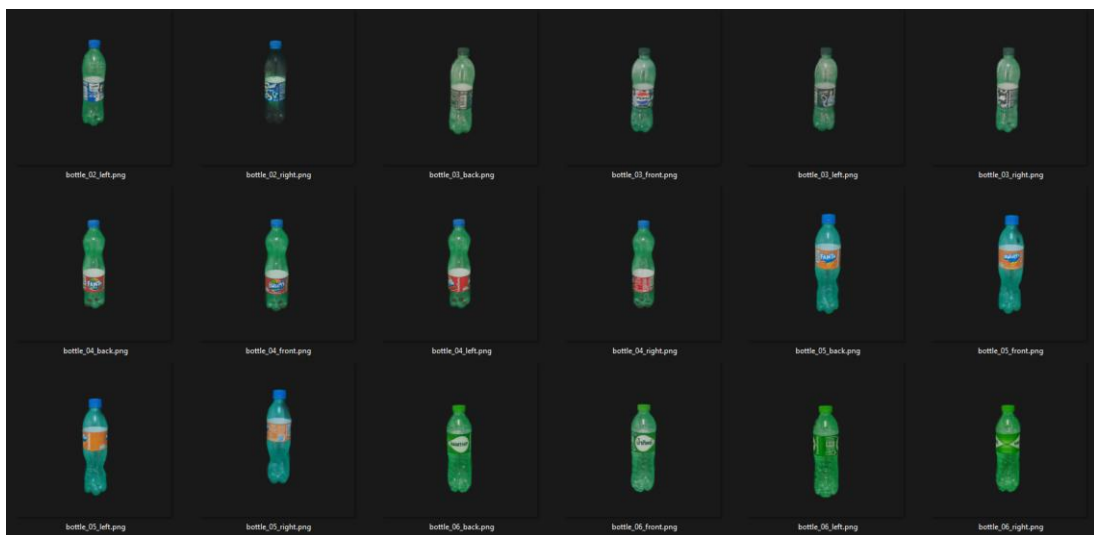
รูป 6 ตัวอย่างภาพพื้นหลังจากแหล่งข้อมูลออนไลน์

4.2.4. การจัดเก็บข้อมูล

ภาพพีเจอร์และพื้นหลังที่ได้จะถูกจัดเก็บอย่างเป็นระบบในโฟลเดอร์เฉพาะ ได้แก่:

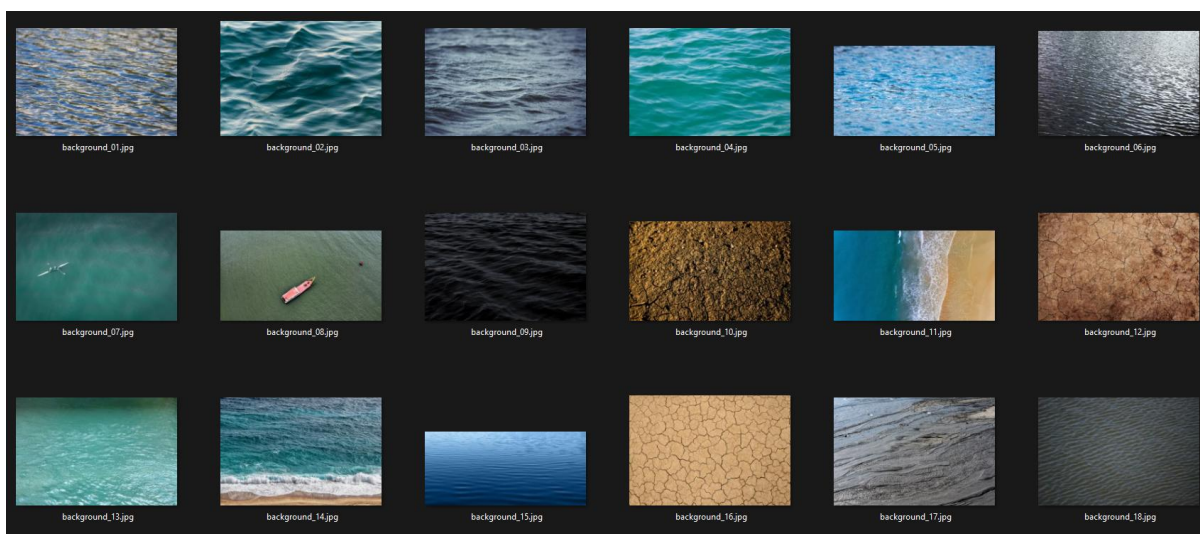
- โฟลเดอร์ features/: สำหรับเก็บภาพขวดพลาสติกพร้อมพื้นหลังโปร่งใส
- โฟลเดอร์ backgrounds/: สำหรับเก็บภาพพื้นหลังจากแหล่งน้ำ

แต่ละไฟล์จะถูกตั้งชื่อให้สื่อความหมาย เช่น bottle_01_front.png สำหรับพีเจอร์ หรือ background_01.jpg สำหรับพื้นหลัง เพื่อความสะดวกในการเรียกใช้งานในขั้นตอนถัดไป



ภาพ 4.5 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ features/

รูป 7 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ features/



ภาพ 4.6 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ backgrounds/

รูป 8 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ *backgrounds/*

4.2.5. การตรวจสอบคุณภาพข้อมูล

ข้อมูลทั้งหมดจะถูกตรวจสอบคุณภาพก่อนนำไปใช้ เพื่อให้มั่นใจว่า ไม่มีสิ่งรบกวน เช่น เงามสะท้อน หรือการแยกพื้นหลังที่ผิดพลาด ภาพมีความคมชัดและรายละเอียดเพียงพอ สำหรับกระบวนการสร้างชุดข้อมูลจำลอง ข้อมูลที่ได้จากกระบวนการนี้จะถูกนำไปใช้ใน ขั้นตอนการสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) เพื่อสนับสนุนการพัฒนาโมเดล ตรวจจับขวดพลาสติกในขั้นตอนถัดไป

4.3. การสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset)

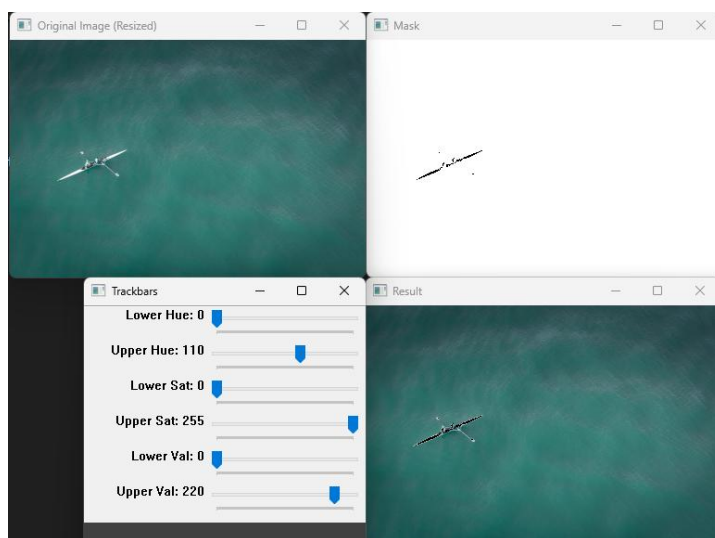
การสร้างชุดข้อมูลภาพจำลองในโครงการนี้ใช้กระบวนการเชิงโปรแกรมที่ออกแบบมาเพื่อเพิ่มปริมาณและความหลากหลายของข้อมูล ชุดข้อมูลที่สร้างขึ้นจะช่วยสนับสนุนการพัฒนาโมเดล ตรวจจับขวดพลาสติกบนพื้นน้ำ โดยการวางพีเจอร์ (ขวดพลาสติก) บนภาพพื้นหลังด้วยวิธีการสุ่ม และการสร้าง Annotation อัตโนมัติ ซึ่งสามารถอธิบายรายละเอียดได้ดังนี้:

4.3.1. การนำพีเจอร์และภาพพื้นหลังมาใช้งาน

พีเจอร์ที่แยกพื้นหลังแล้ว (ขวดพลาสติก) จะถูกนำมาจากโฟลเดอร์ features/ ในขณะที่ภาพพื้นหลังจะนำมาจากโฟลเดอร์ backgrounds/ สำหรับแต่ละรอบของการสร้างภาพจำลอง พีเจอร์และภาพพื้นหลังจะถูกสุ่มเลือกเพื่อสร้างความหลากหลายของชุดข้อมูล ช่วยให้ข้อมูลมีความครอบคลุมในลักษณะที่ใกล้เคียงกับสถานการณ์จริงในแหล่งน้ำ

4.3.2. การตรวจจับพื้นที่น้ำ (Water Mask Generation)

พื้นที่น้ำในภาพพื้นหลังถูกตรวจจับโดยใช้ฟังก์ชัน `detect_water_area` ที่พัฒนาด้วย OpenCV การตรวจจับเริ่มจากการแปลงภาพพื้นหลังเป็นโหมดสี HSV และใช้ค่าช่วงสี (HSV Range) ที่กำหนดไว้ ($\text{LOWER_BOUND} = [0, 0, 0]$ และ $\text{UPPER_BOUND} = [110, 255, 220]$) เพื่อสร้าง Mask น้ำ ตัวกรอง Morphological เช่น Closing และ Opening ถูกใช้เพื่อลด Noise และปรับปรุงความแม่นยำของ Mask ซึ่ง Mask ที่ได้จะถูกใช้เพื่อกำหนดพื้นที่น้ำที่เหมาะสมสำหรับการวางพีเจอร์



ภาพ 4.6 ตัวอย่างการค่าช่วงสี HSV ที่กำหนดเพื่อสร้าง Mask ของพื้นที่น้ำ

รูป 9 ตัวอย่างการค่าช่วงสี HSV ที่กำหนดเพื่อสร้าง Mask ของพื้นที่น้ำ

4.3.3. การวางพีเจอร์บนพื้นที่น้ำ

การวางพีเจอร์ (ขวดพลาสติก) บนพื้นที่น้ำในภาพพื้นหลังดำเนินการโดยฟังก์ชัน `place_feature_on_water` ซึ่งสุ่มขนาด (Scale) ของพีเจอร์ระหว่าง $0.08x$ ถึง $0.3x$ ของขนาดเดิม และสุ่มพิกัด (x, y) ในพื้นที่ Mask น้ำ ระบบจะตรวจสอบให้พีเจอร์ไม่หลุดจาก Mask น้ำ และกำหนดว่าพีเจอร์อย่างน้อย 50% ต้องอยู่ในพื้นที่น้ำ

4.3.4. การสร้างภาพจำลอง

การรวมพีเจอร์เข้ากับภาพพื้นหลังใช้ฟังก์ชัน `overlay_feature` ซึ่งจะหมุนพีเจอร์แบบสุ่ม (0° ถึง 360°) ด้วย Rotation Matrix จากนั้นใช้เทคนิค Alpha Blending เพื่อผสมพีเจอร์กับภาพพื้นหลังให้ดูสมจริง



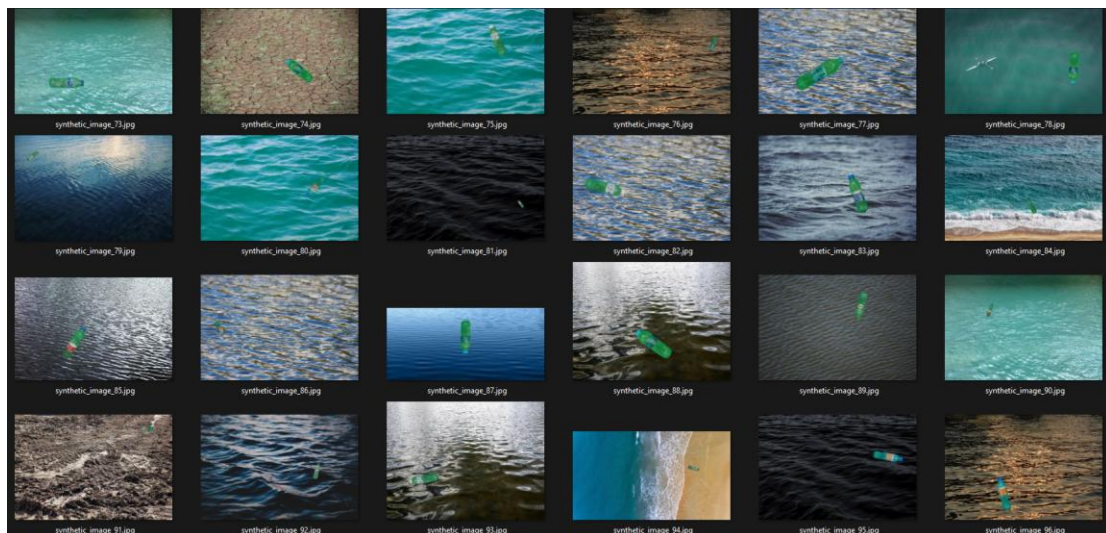
ภาพ 4.7 ตัวอย่างการวางพีเจอรบนพื้นน้ำโดยสุ่มขนาด พิกัด และมุมหมุน
รูป 10 ตัวอย่างการวางพีเจอรบนพื้นน้ำโดยสุ่มขนาด พิกัด และมุมหมุน

4.3.5. การสร้าง Annotation

Annotation สำหรับ Bounding Box ของพีเจอรถูกสร้างโดยฟังก์ชัน `save_annotation` ซึ่งบันทึกค่า `x_center`, `y_center`, `width`, และ `height` ในรูปแบบ YOLO ค่าเหล่านี้ถูก Normalize ตามขนาดของภาพ Annotation ที่สร้างขึ้นจะถูกจัดเก็บในโฟลเดอร์ `annotations/` โดยมีชื่อไฟล์ตรงกับชื่อภาพ เช่น `synthetic_image_1.txt`

4.3.6. การจัดเก็บข้อมูล

ภาพที่สร้างสำเร็จจะถูกจัดเก็บในโฟลเดอร์ `synthetic_dataset/` โดยตั้งชื่อไฟล์ในรูปแบบ `synthetic_image_1.jpg` หากการวางพีเจอรไม่สำเร็จ เช่น ไม่มีพื้นน้ำเพียงพอ ระบบจะบันทึกภาพพื้นหลังและ Mask น้ำในโฟลเดอร์สำหรับการตรวจสอบและการปรับปรุงพารามิเตอร์ในอนาคต



ภาพ 4.8 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ `synthetic_dataset/`

รูป 11 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ `synthetic_dataset/`

4.3.7. การตรวจสอบและประเมินผล

ระบบจะบันทึกข้อมูลภาพที่ล้มเหลวในการวางพีเจอร์ เพื่อใช้ในการวิเคราะห์และปรับปรุงพารามิเตอร์ เช่น การปรับช่วงค่า HSV หรือการขยายขนาด Mask น้ำ หลังจากระบบสร้างภาพครบตามจำนวนที่กำหนด (เช่น `num_images = 1000`) ระบบจะหยุดทำงาน

ชุดข้อมูลภาพจำลองที่ได้จากขั้นตอนนี้จะมีความหลากหลายและสมจริง พร้อมทั้ง Annotation ในรูปแบบที่เหมาะสมสำหรับการฝึกโมเดล YOLO ในขั้นตอนถัดไป

4.4. การอัปโหลดชุดข้อมูลเข้า Roboflow (สำหรับ YOLOv8)

การอัปโหลดชุดข้อมูลเข้าแพลตฟอร์ม Roboflow สำหรับโมเดล YOLOv8 เป็นขั้นตอนสำคัญที่ช่วยปรับปรุงคุณภาพข้อมูลและทำให้ข้อมูลพร้อมสำหรับการฝึกโมเดล Roboflow ช่วยจัดการ Annotation, เพิ่มความหลากหลายของข้อมูลด้วย Data Augmentation และแบ่งข้อมูลออกเป็น Training, Validation และ Test Sets พร้อมทั้งส่งออกในฟอร์แมตที่รองรับ YOLOv8 ได้อย่างมีประสิทธิภาพ

4.4.1. การจัดเตรียมชุดข้อมูล

ชุดข้อมูลที่สร้างขึ้นจากขั้นตอนก่อนหน้านี้ประกอบด้วยไฟล์ภาพและไฟล์ Annotation ที่จับคู่กันอย่างถูกต้อง ชุดข้อมูลภาพถูกจัดเก็บในโฟลเดอร์ `synthetic_dataset/` เช่น `synthetic_image_001.jpg` และไฟล์ Annotation ในโฟลเดอร์

annotations/ เช่น synthetic_image_001.txt ชื่อไฟล์ของภาพและ Annotation ต้องตรงกันเพื่อให้ Roboflow สามารถประมวลผลข้อมูลได้อย่างถูกต้อง

4.4.2. การสร้างโปรเจกต์ใน Roboflow

เข้าสู่ระบบ Roboflow โดยลงชื่อเข้าใช้บนเว็บไซต์ Roboflow และสร้างโปรเจกต์ใหม่ผ่านหน้าจอ “Create New Project” ตั้งชื่อโปรเจกต์ เช่น YOLOv8 Plastic Bottle Detection และเลือกฟอร์แมตเป็น YOLOv8 PyTorch TXT เพื่อรองรับการใช้งานโมเดล YOLOv8 ในขั้นตอนการฝึก

4.4.3. การอัปโหลดชุดข้อมูล

หลังจากสร้างโปรเจกต์เรียบร้อยแล้ว ไฟล์ภาพและไฟล์ Annotation จากโฟลเดอร์ที่เตรียมไว้จะถูกอัปโหลดเข้าสู่ Roboflow โดยเลือกไฟล์ภาพ เช่น synthetic_image_001.jpg และไฟล์ Annotation เช่น synthetic_image_001.txt พร้อมกัน Roboflow จะตรวจสอบความถูกต้องของข้อมูลและแสดงผลภาพพร้อม Annotation ให้ผู้ใช้งานตรวจสอบ นอกจากนี้ ยังมีการระบุ Class สำหรับการตรวจจับ เช่น ขวดพลาสติก เพื่อกำหนดเป้าหมายของโมเดล

4.4.4. การปรับแต่งชุดข้อมูล

Roboflow มีฟีเจอร์สำหรับเพิ่มความหลากหลายของข้อมูล (Data Augmentation) เช่น การหมุนภาพ (Rotation), การปรับความสว่าง (Brightness Adjustment) และการเพิ่ม Noise หรือ Blur เพื่อให้ชุดข้อมูลมีความครอบคลุมและเหมาะสมกับการฝึกโมเดล หลังจากการปรับแต่งชุดข้อมูล ระบบจะแบ่งข้อมูลเป็น Training Set (80%), Validation Set (10%) และ Test Set (10%) โดยอัตโนมัติ เพื่อเตรียมความพร้อมสำหรับการฝึกโมเดลและการประเมินผล

4.4.5. การส่งออกชุดข้อมูล

เมื่อปรับแต่งชุดข้อมูลเสร็จสิ้นแล้ว Roboflow จะให้ผู้ใช้งานเลือกส่งออกชุดข้อมูลในฟอร์แมต YOLOv8 PyTorch TXT ไฟล์ที่ได้จะอยู่ในรูปแบบ ZIP ซึ่งประกอบด้วยภาพและ Annotation พร้อมใช้งานสำหรับการฝึกโมเดล YOLOv8 ในขั้นตอนถัดไป ผู้ใช้งานสามารถดาวน์โหลดไฟล์ ZIP และนำไปใช้ในระบบฝึกโมเดลบนแพลตฟอร์ม เช่น Google Colab หรือ Local Environment ได้อย่างสะดวก

ชุดข้อมูลที่ผ่านกระบวนการใน Roboflow นี้ ช่วยให้ข้อมูลพร้อมใช้งานและมีความหลากหลาย รองรับการใช้งานโมเดล YOLOv8 ได้อย่างมีประสิทธิภาพและแม่นยำในขั้นตอนต่อไป

4.5. การฝึกฝนแบบจำลอง YOLOv8

การฝึกฝนแบบจำลอง YOLOv8 เป็นขั้นตอนสำคัญในการพัฒนาโมเดลที่สามารถตรวจจับวัตถุพลาสติกบนพื้นที่น่าได้อย่างมีประสิทธิภาพ โดยใช้ชุดข้อมูลที่ปรับแต่งและจัดการผ่าน Roboflow และดำเนินการฝึกโมเดลบนแพลตฟอร์ม Google Colab ซึ่งมีทรัพยากร GPU สำหรับประมวลผลที่เหมาะสม ช่วยเพิ่มความเร็วและประสิทธิภาพในการฝึกโมเดล ขั้นตอนการฝึกประกอบด้วยรายละเอียดดังนี้:

4.5.1. การเตรียมสภาพแวดล้อม

กระบวนการฝึกโมเดลเริ่มต้นด้วยการเตรียมสภาพแวดล้อมบน Google Colab ซึ่งรองรับการประมวลผลด้วย GPU และเหมาะสำหรับการทำงานที่ใช้ทรัพยากรสูง ในขั้นตอนนี้ระบบจะทำการติดตั้งไลบรารี Ultralytics YOLOv8 เพื่อใช้เป็นโครงสร้างพื้นฐานสำหรับการพัฒนาโมเดล หลังการติดตั้ง จะมีการตรวจสอบเพื่อให้แน่ใจว่าไลบรารีทำงานได้อย่างถูกต้อง

4.5.2. การตั้งค่าและการโหลดชุดข้อมูล

ชุดข้อมูลที่ได้จาก Roboflow จะถูกอัปโหลดเข้าสู่ Google Colab และจัดเก็บในโครงสร้างที่เหมาะสม ไฟล์ YAML ถูกสร้างขึ้นเพื่อกำหนดโครงสร้างของชุดข้อมูลสำหรับการฝึกโมเดล โดยกำหนดเส้นทางของ Training Set และ Validation Set รวมถึงระบุจำนวนคลาสที่ต้องตรวจจับ ในกรณีนี้คือ 1 คลาส สำหรับขวดพลาสติก

4.5.3. กระบวนการฝึกโมเดล

โมเดล YOLOv8n (nano) ถูกเลือกใช้เนื่องจากมีขนาดเล็กและเหมาะสมกับทรัพยากรที่ใช้ การตั้งค่าพารามิเตอร์สำหรับการฝึกประกอบด้วยจำนวนรอบ (Epoch) ที่ตั้งไว้ 50 รอบ ขนาดภาพ (Image Size) ที่ 640x640 ขนาด Batch ที่ 16 และจำนวน Worker ที่ 4 หลังจากการฝึก ระบบจะสร้างผลลัพธ์ เช่น mAP (Mean Average Precision), Precision และ Recall เพื่อประเมินประสิทธิภาพของโมเดล

4.5.4. การประเมินผลลัพธ์ของโมเดล

หลังการฝึก โมเดลจะถูกประเมินด้วยชุดข้อมูล Validation เพื่อวิเคราะห์ประสิทธิภาพ โดยใช้ตัวชี้วัดหลัก ได้แก่:

- mAP (Mean Average Precision): วัดความสามารถของโมเดลในการตรวจจับวัตถุ
- Precision: วัดความถูกต้องของการตรวจจับ

- Recall: วัดความครอบคลุมของการตรวจจับ

4.5.5. การปรับปรุงโมเดล (Fine-tuning)

หากผลลัพธ์ของโมเดลยังไม่เป็นที่น่าพอใจ การปรับปรุงโมเดลสามารถทำได้โดยการเพิ่มจำนวน Epoch เพื่อให้โมเดลเรียนรู้มากขึ้น ปรับพารามิเตอร์ เช่น Batch Size หรือใช้ Data Augmentation เพิ่มเติมใน Roboflow เพื่อเพิ่มความหลากหลายของข้อมูล

4.5.6. การบันทึกและส่งออกโมเดล

เมื่อการฝึกเสร็จสิ้น โมเดลที่ผ่านการปรับแต่งจะถูกบันทึกและส่งออกในฟอร์แมตที่เหมาะสม เช่น ONNX หรือ PyTorch เพื่อใช้งานในเฟรมเวิร์คที่พัฒนาขึ้นในขั้นตอนถัดไป โมเดลนี้จะถูกนำไปใช้ในการตรวจจับวัตถุพลาสติกบนพื้นน้ำในระบบที่ออกแบบไว้

4.6. การพัฒนาเฟรมเวิร์ค AI

เฟรมเวิร์ค AI ในโครงการนี้ถูกออกแบบมาเพื่อช่วยสนับสนุนการทำงานของระบบตั้งแต่การเตรียมข้อมูลไปจนถึงการฝึกโมเดล โดยเน้นกระบวนการ **Automate** ในการเตรียม Training Dataset เพื่อลดความซับซ้อนของกระบวนการ และเพิ่มประสิทธิภาพในการสร้างชุดข้อมูลสำหรับการฝึกโมเดล YOLOv8 รายละเอียดดังนี้:

4.6.1. การ Automate การสร้าง Synthetic Dataset

กระบวนการสร้าง Synthetic Dataset ได้รับการออกแบบให้สามารถทำงานได้โดยอัตโนมัติ โดยมีการนำฟีเจอร์ (วัตถุพลาสติก) และภาพพื้นหลังจากโพลเดอร์ที่กำหนดมาใช้ ระบบจะสุ่มเลือกฟีเจอร์และภาพพื้นหลังในแต่ละรอบเพื่อเพิ่มความหลากหลาย หลังจากนั้น พื้นที่น้ำในภาพพื้นหลังจะถูกตรวจจับด้วย Mask ที่สร้างจากค่าช่วงสี HSV การวางฟีเจอร์บนพื้นที่น้ำจะดำเนินการโดยอัตโนมัติด้วยฟังก์ชันที่ตรวจสอบขนาด พิกัด และมุมหมุนของฟีเจอร์ เพื่อให้ได้ภาพที่ดูสมจริง ระบบจะสร้างไฟล์ Annotation ที่ตรงกับภาพโดยใช้ฟอร์แมต YOLO และจัดเก็บข้อมูลที่สร้างสำเร็จในโพลเดอร์เฉพาะ

4.6.2. การ Automate Data Augmentation บน Roboflow

หลังจากสร้าง Synthetic Dataset แล้ว ระบบจะสนับสนุนการอัปโหลดชุดข้อมูลเข้าสู่ Roboflow เพื่อทำ Data Augmentation โดยอัตโนมัติ กระบวนการนี้ช่วยเพิ่มความ

หลากหลายของชุดข้อมูล เช่น การหมุนภาพ (Rotation), การปรับแสง (Brightness Adjustment), และการเพิ่ม Noise หรือ Blur Roboflow มีฟีเจอร์สำหรับการตั้งค่าขั้นตอนเหล่านี้ล่วงหน้า ทำให้ผู้พัฒนาไม่ต้องดำเนินการด้วยตนเองในแต่ละชุดข้อมูล หลังจากปรับแต่งชุดข้อมูลเสร็จสิ้น Roboflow จะแบ่งข้อมูลเป็น Training Set, Validation Set และ Test Set พร้อมส่งออกในฟอร์แมตที่เหมาะสมกับ YOLOv8

4.6.3. การเชื่อมโยงกระบวนการแบบ Automate

เฟรมเวิร์คได้รับการออกแบบให้เชื่อมโยงกระบวนการตั้งแต่การสร้าง Synthetic Dataset ไปจนถึงการปรับแต่งชุดข้อมูลใน Roboflow โดยอัตโนมัติ ตัวอย่างการดำเนินงานคือการใช้สคริปต์ Python สำหรับเรียกใช้ฟังก์ชันต่าง ๆ เช่น:

- การสุ่มวางฟีเจอร์บนพื้นพื้นน้ำ
- การสร้าง Annotation อัตโนมัติ
- การอัปโหลดข้อมูลไปยัง Roboflow และเลือกค่าพารามิเตอร์สำหรับ Data Augmentation

4.6.4. การจัดเก็บและจัดการข้อมูล

ข้อมูลที่สร้างและปรับแต่งทั้งหมดจะถูกจัดเก็บในโครงสร้างโฟลเดอร์ที่ชัดเจน เช่น:

- Synthetic Dataset: สำหรับเก็บภาพและ Annotation ที่สร้างจากระบบ
- Roboflow Export: สำหรับเก็บข้อมูลที่ผ่านการปรับแต่งจาก Roboflow

ระบบจะตรวจสอบความสมบูรณ์ของข้อมูล เช่น ชื่อไฟล์ภาพและ Annotation ต้องตรงกัน และข้อมูลทั้งหมดจะถูกเตรียมพร้อมสำหรับการฝึกโมเดล YOLOv8 โดยอัตโนมัติ

4.6.5. การปรับปรุงเฟรมเวิร์คในอนาคต

เฟรมเวิร์คที่พัฒนาขึ้นมีความยืดหยุ่นสูง โดยสามารถปรับเปลี่ยนค่าพารามิเตอร์หรือเพิ่มฟีเจอร์เพิ่มเติม เช่น การปรับแต่งช่วงค่าสี HSV หรือการเพิ่มตัวเลือก Augmentation ใหม่ ๆ ผ่าน Roboflow การ Automate กระบวนการทั้งหมดช่วยลดเวลาและข้อผิดพลาดที่อาจเกิดจากการทำงานด้วยตนเอง