

**การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุ**

**ประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก**

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

**ธีรธร รักษาเมือง**

**สุปวีณ์ สัญจร**

**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 สัญจร

ชญาดา เอียดทองใส

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์

สำนักวิชาวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยวลัยลักษณ์

ปีการศึกษา 2567

**บทคัดย่อ**

**บทคัดย่อ อังกฤษ**

**กิตติกรรมประกาศ**

**สารบัญ**

**สารบัญตาราง**

**สารบัญภาพ**

**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. สร้างโมเดลที่สามารถตรวจจับวัตถุได้อย่างแม่นยำและมีความเร็วในการประมวลผล โดยใช้ข้อมูลที่มีจำนวนจำกัดในการฝึกโมเดล
3. เปรียบเทียบวิธีการเพิ่มปริมาณชุดข้อมูลภาพที่นำมาใช้ในการฝึกโมเดล และวิเคราะห์ผลลัพธ์เพื่อให้ได้โมเดลที่มีประสิทธิภาพสูงสุด

**1.3 ขอบเขตของงาน**

**1.3.1 การเก็บข้อมูลและจัดการข้อมูล**

* เก็บข้อมูลภาพขวดพลาสติกที่ถ่ายในบริบทที่ควบคุม เช่น พื้นหลังสีเขียว
* ใช้กระบวนการ Data Augmentation เช่น การหมุน การปรับขนาด และการย้ายตำแหน่ง เพื่อเพิ่มความหลากหลายของข้อมูล
* จัดเก็บข้อมูลที่แยกเป็น Training Set และ Test Set สำหรับการพัฒนาและทดสอบโมเดล
* ตรวจสอบและจัดเตรียมข้อมูล เช่น การสร้าง Mask ของพื้นที่น้ำ เพื่อให้เหมาะสมกับการสร้างชุดข้อมูลจำลอง (Synthetic Dataset)

**1.3.2. การพัฒนาและฝึกอบรมโมเดล**

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

**1.3.3. การพัฒนาระบบอัตโนมัติ**

* พัฒนากระบวนการอัตโนมัติสำหรับการสร้างชุดข้อมูลจำลอง ที่สามารถวางฟีเจอร์ขวดพลาสติกบนพื้นที่น้ำในภาพพื้นหลั
* เตรียมระบบสำหรับการนำข้อมูลเข้าแพลตฟอร์ม RoboFlow เพื่อจัดการชุดข้อมูลและการฝึกโมเดล
* ออกแบบอินเทอร์เฟซหรือเครื่องมือสำหรับผู้ใช้งาน เพื่อช่วยให้สามารถดำเนินการสร้างชุดข้อมูลและใช้งานระบบได้สะดวก

**1.3.4. การประเมินผลและปรับปรุงระบบ**

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

**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 และเปิดโอกาสใหม่ในการสร้างผลิตภัณฑ์ที่มีประสิทธิภาพสูงยิ่งขึ้น

**1.5 ผลที่คาดว่าจะได้รับเมื่อเสร็จสิ้นโครงการ**

เมื่อโครงการการพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็กเสร็จสิ้นลง คาดว่าจะได้รับผลลัพธ์ที่มีความสำคัญหลายประการ ดังนี้:

1. แนวทางประยุกต์ที่เหมาะสมในการขยายจำนวนของชุดข้อมูลภาพ
2. โมเดลที่สามารถตรวจจับวัตถุได้อย่างแม่นยำและมีความเร็วในการประมวลผล โดยใช้ข้อมูลที่มีจำนวนจำกัดในการฝึกโมเดล
3. วิธีการเพิ่มปริมาณชุดข้อมูลภาพที่นำมาใช้ในการฝึกโมเดล และวิเคราะห์ผลลัพธ์เพื่อให้ได้โมเดลที่มีประสิทธิภาพสูงสุด

ผลที่คาดว่าจะได้รับเหล่านี้จะทำให้โครงการมีผลกระทบเชิงบวกต่อการพัฒนาเทคโนโลยีและการดำเนินงานในภาคธุรกิจต่าง ๆ โดยเฉพาะในด้านการตรวจจับวัตถุที่ต้องการความแม่นยำและรวดเร็ว

**1.6 ขั้นตอนการดำเนินงาน**

**A grid of squares with a few squares in it

Description automatically generated with medium confidence**

หมายเหตุ หมายถึง ระยะเวลาแผนการดำเนินงาน

^ หมายถึง สอบข้อเสนอโครงงาน

^1 หมายถึง สอบกลางภาคโครงงาน

^2 หมายถึง สอบความก้าวหน้าโครงงาน

**1.7 อุปกรณ์ที่ใช้ในการพัฒนางาน**

การพัฒนาระบบ 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

**ซอฟต์แวร์สำหรับจัดการเอกสาร**

* **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**: ไลบรารีสำหรับลบพื้นหลังของภาพเพื่อตัดฟีเจอร์ออกจากแบกกราวด์
* **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 เทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ**

**2.1.1 เทคโนโลยีด้าน Machine Learning และ Deep Learning**

* **YOLO (You Only Look Once)**: เทคนิคในการตรวจจับวัตถุที่มีความเร็วสูงและความแม่นยำ โดยสามารถระบุวัตถุและตำแหน่งได้ในภาพเดียว
* **TensorFlow/Keras**: เฟรมเวิร์กสำหรับพัฒนาโมเดล Deep Learning ที่ใช้งานง่าย และมีความยืดหยุ่น
* **OpenCV**: ไลบรารีสำหรับประมวลผลภาพ เช่น การสร้าง Synthetic Dataset และการจัดการ Mask

**2.1.2 เทคโนโลยีการจัดการข้อมูล**

* **Firebase**: ใช้จัดเก็บข้อมูลผู้ใช้งานและ Metadata ที่เกี่ยวข้องกับระบบ
* **RoboFlow**: แพลตฟอร์มสำหรับการจัดการชุดข้อมูล เช่น การทำ Data Augmentation และการสร้าง Annotation

**2.1.3 ซอฟต์แวร์และเครื่องมือ**

* G**oogle Colab**: ใช้สำหรับทดลองโมเดลโดยใช้ทรัพยากรคลาวด์ (GPU/TPU)
* **VS Code**: ใช้เขียนและแก้ไขโค้ด โดยเฉพาะโค้ด Python สำหรับการพัฒนาโมเดลและระบบ
* **Jupyter Notebook**: ใช้สำหรับการพัฒนาโมเดลและการวิเคราะห์ข้อมูล

**2.2 ทฤษฎีที่เกี่ยวข้อง2.2.1 การตรวจจับวัตถุ (Object Detection)**

* เป็นเทคนิคในคอมพิวเตอร์วิชันที่ใช้ในการระบุวัตถุในภาพหรือวิดีโอ พร้อมทั้งบอกตำแหน่งของวัตถุ
* เทคนิคที่ใช้ เช่น YOLO, Faster R-CNN, SSD (Single Shot Multibox Detector)

**2.2.2 การเพิ่มข้อมูล (Data Augmentation)**

* เทคนิคที่ใช้ขยายชุดข้อมูลโดยการปรับเปลี่ยนภาพ เช่น การหมุน การปรับขนาด และการเปลี่ยนมุมมอง เพื่อเพิ่มความหลากหลายและช่วยให้โมเดลเรียนรู้ได้ดียิ่งขึ้น

**2.2.3 Mean Average Precision (mAP)**

* เป็นเมตริกสำหรับประเมินความแม่นยำของโมเดลตรวจจับวัตถุ โดยใช้ค่าความแม่นยำ (Precision) และค่าความครอบคลุม (Recall) ในการคำนวณ

**2.2.4 การประมวลผลภาพ (Image Processing)**

* การใช้เทคนิคต่าง ๆ เช่น การ Threshold, การแปลงภาพเป็น HSV, และการ Morphological Filtering เพื่อแยกพื้นที่น้ำออกจากภาพพื้นหลัง

**2.3 งานวิจัยหรือระบบงานใกล้เคียง**

ในปัจจุบันมีงานวิจัยและซอฟต์แวร์ที่มุ่งเน้นการสร้างและฝึกอบรมโมเดลตรวจจับวัตถุที่มีประสิทธิภาพสูง แม้ว่างานส่วนใหญ่จะเน้นการใช้ชุดข้อมูลขนาดใหญ่ แต่งานวิจัยบางชิ้นยังมุ่งเน้นไปที่การพัฒนาโมเดลสำหรับชุดข้อมูลขนาดเล็ก เพื่อลดความจำเป็นในการใช้ทรัพยากรที่สูงในขั้นตอนการฝึกอบรม โดยระบบงานใกล้เคียงที่มีแนวคิดใกล้เคียงกับโครงการนี้ ได้แก่ YOLO (You Only Look Once), การพัฒนาการตรวจจับรอยเปื้อนบนเสื้อผ้า (Development of Stain Detection on Clothes), และ พฤติกรรมการกินถุงพลาสติกของเต่าทะเลและผลกระทบต่อสุขภาพ: กรณีศึกษามลภาวะพลาสติกในทะเลไทย (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**

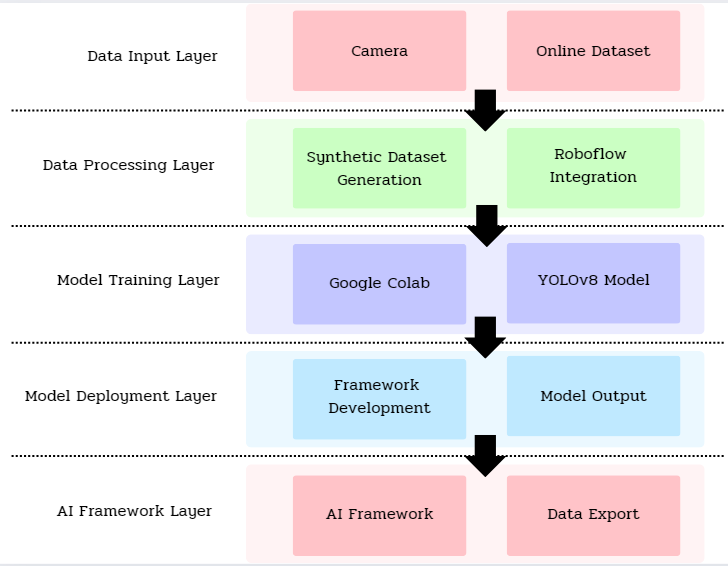
**เอกสารความต้องการของระบบ**

โครงการนี้มุ่งเน้นการพัฒนาโมเดลการตรวจจับวัตถุที่มีประสิทธิภาพสูงสำหรับการตรวจจับขวดพลาสติกบนพื้นที่น้ำ โดยใช้เทคโนโลยีการเรียนรู้ของเครื่อง (Machine Learning) และการสร้างชุดข้อมูลจำลอง (Synthetic Dataset) เพื่อสนับสนุนการแก้ปัญหามลพิษในแหล่งน้ำผ่านการเก็บขวดพลาสติกโดยอัตโนมัติ นอกจากนี้ยังมีการพัฒนาระบบอัตโนมัติที่ช่วยให้ผู้ใช้งานสามารถสร้างและจัดการข้อมูลได้สะดวก รวมถึงปรับปรุงกระบวนการฝึกโมเดลให้มีความแม่นยำสูง

โครงการนี้มีเป้าหมายเพื่อสนับสนุนการจัดการปัญหามลพิษในแหล่งน้ำอย่างยั่งยืน โดยเน้นการพัฒนาโมเดลที่ช่วยในการตรวจจับและแยกแยะขวดพลาสติกจากภาพถ่าย นำไปสู่การลดปริมาณขยะพลาสติกในแหล่งน้ำโดยการใช้ระบบอัตโนมัติ สนับสนุนการป้องกันและแก้ปัญหามลพิษในสิ่งแวดล้อมด้วยเทคโนโลยี Deep Learning สร้างแนวทางการใช้ปัญญาประดิษฐ์เพื่อเพิ่มประสิทธิภาพในกระบวนการตรวจจับและเก็บขวดพลาสติก ส่งเสริมการประยุกต์ใช้เทคโนโลยีปัญญาประดิษฐ์ในด้านสิ่งแวดล้อมอย่างมีประสิทธิภาพ

**3.1 การวิเคราะห์ระบบ (System Analysis)**

* + 1. **ภาพรวมสถาปัตยกรรมระบบ (System Architecture Diagram)**

****

*ภาพที่ 3.1* *System Architecture Diagram ของทั้งระบบ*

* + 1. **การศึกษาความต้องการของผู้ใช้งาน (User Requirement Analysis)**

การเก็บรวบรวมข้อมูลจากกลุ่มผู้ใช้งานเป้าหมายเพื่อวิเคราะห์ความต้องการที่ แท้จริงเป็นขั้นตอนสำคัญ ตารางที่ 3.1 คือตารางความต้องการของผู้ใช้ที่ต้องการให้มีการ พัฒนา มีรายละเอียดดังต่อไปนี้

**ตารางที่ 3.1** ตารางแสดงความต้องการของผู้ใช้

|  |  |  |
| --- | --- | --- |
| **ลำดับ** | **คำอธิบายความต้องการ** | **ความสำคัญ** |
| Req 01 | ระบบที่รองรับการตรวจจับขวดพลาสติกบนพื้นที่น้ำจากภาพถ่าย | Must have |
| Req 02 | ระบบสามารถสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) พร้อมกับ Annotation ที่เหมาะสม | Must have |
| Req 03 | ระบบต้องสามารถแสดงผลการตรวจจับ เช่น ตำแหน่งขวดพลาสติกบนพื้นน้ำในภาพถ่าย | Must have |
| Req 04 | ระบบต้องรองรับการประเมินผลการทำงานของโมเดล ด้วยเมตริก เช่น Accuracy, Precision, Recall | Must have |
| Req 05 | ระบบสามารถสร้าง Mask น้ำอัตโนมัติจากภาพพื้นหลัง | Must have |
| Req 06 | ระบบต้องมีอินเทอร์เฟซที่ช่วยให้ผู้ใช้งานสามารถใช้งานได้สะดวก | 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 | ระบบสามารถประเมินข้อผิดพลาดของการตรวจจับ และให้คำแนะนำสำหรับการปรับปรุงโมเดล | Could have |

เมื่อได้ทำการรวบรวมและจัดลำดับความสำคัญของความต้องการจากผู้ใช้งานเรียบร้อยแล้ว ขั้นตอนถัดไปคือการตรวจสอบและวิเคราะห์เพื่อระบุฟังก์ชันที่เหมาะสมสำหรับตอบสนองความต้องการเหล่านั้น โดยให้ความสำคัญกับการพิจารณาฟังก์ชันที่สามารถแก้ไขปัญหาได้อย่างมีประสิทธิภาพ พร้อมทั้งกำหนดความเชื่อมโยงระหว่างฟังก์ชันที่ออกแบบและความต้องการของผู้ใช้งานในแต่ละข้ออย่างชัดเจน เพื่อให้มั่นใจว่าฟังก์ชันเหล่านั้นสามารถตอบสนองความต้องการของผู้ใช้ได้อย่างครบถ้วนและสามารถตรวจสอบได้ตามมาตรฐานที่กำหนด

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

|  |  |  |
| --- | --- | --- |
| **คุณลักษณะของ UC01:**  การสร้างชุดข้อมูลภาพจำลองและการจัดการข้อมูล (Synthetic Dataset and Data Management) | | |
| **คำอธิบาย:**  ผู้พัฒนาสามารถสร้างชุดข้อมูลภาพจำลอง พร้อม Annotation และสามารถจัดการชุดข้อมูลภาพที่สร้างขึ้นได้ เช่น เพิ่ม ลบ หรือแก้ไขภาพและ Annotation | | |
| **Actor:** ผู้พัฒนา | | |
| **เงื่อนไข (ข้อกำหนด):**  1. ระบบสามารถสร้างชุดข้อมูลภาพจำลอง พร้อม Mask และ Annotation ได้อย่างถูกต้อง  2. 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: การตรวจจับขวดพลาสติกบนพื้นที่น้ำ (Plastic Bottle Detection)

|  |  |  |
| --- | --- | --- |
| **คุณลักษณะของ UC02:**  การตรวจจับขวดพลาสติกบนพื้นที่น้ำ (Plastic Bottle Detection) | | |
| **คำอธิบาย:**  ระบบสามารถตรวจจับและแสดงตำแหน่งของขวดพลาสติกบนพื้นที่น้ำในภาพถ่าย | | |
| **Actor:** ผู้พัฒนา, ผู้ใช้งาน | | |
| **เงื่อนไข (ข้อกำหนด):**  1. ระบบสามารถแสดงผลการตรวจจับตำแหน่งขวดพลาสติกบนภาพถ่าย  2. ต้องรองรับการประเมินผลการทำงานของโมเดล | | |
| **หมายเลข** | **คำอธิบายข้อกำหนดความต้องการ** | **ความสำคัญ**  **(M,S,C,W)** |
| Req 01 | ระบบที่รองรับการตรวจจับขวดพลาสติกบนพื้นที่น้ำจากภาพถ่าย | M |
| Req 04 | ระบบต้องรองรับการประเมินผลการทำงานของโมเดล เช่น Accuracy, Precision, Recall | M |

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

|  |  |  |
| --- | --- | --- |
| **คุณลักษณะของ UC03:**  การสร้าง Mask น้ำอัตโนมัติ (Water Mask Generation) | | |
| **คำอธิบาย:**  ระบบสามารถสร้าง Mask น้ำอัตโนมัติจากภาพพื้นหลัง | | |
| **Actor:** ผู้พัฒนา | | |
| **เงื่อนไข (ข้อกำหนด):**  1. ระบบสามารถปรับปรุงพารามิเตอร์สำหรับสร้าง Mask น้ำได้  2. Mask ต้องมีความแม่นยำและเหมาะสมกับการวางฟีเจอร์ | | |
| **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)

|  |  |  |
| --- | --- | --- |
| **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)

|  |  |  |
| --- | --- | --- |
| **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 การตรวจสอบด้วยตนเองยังไม่สามารถให้ข้อมูลในเชิงลึกที่ช่วยสนับสนุนการตัดสินใจในการบริหารจัดการขยะอย่างมีประสิทธิภาพ

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**

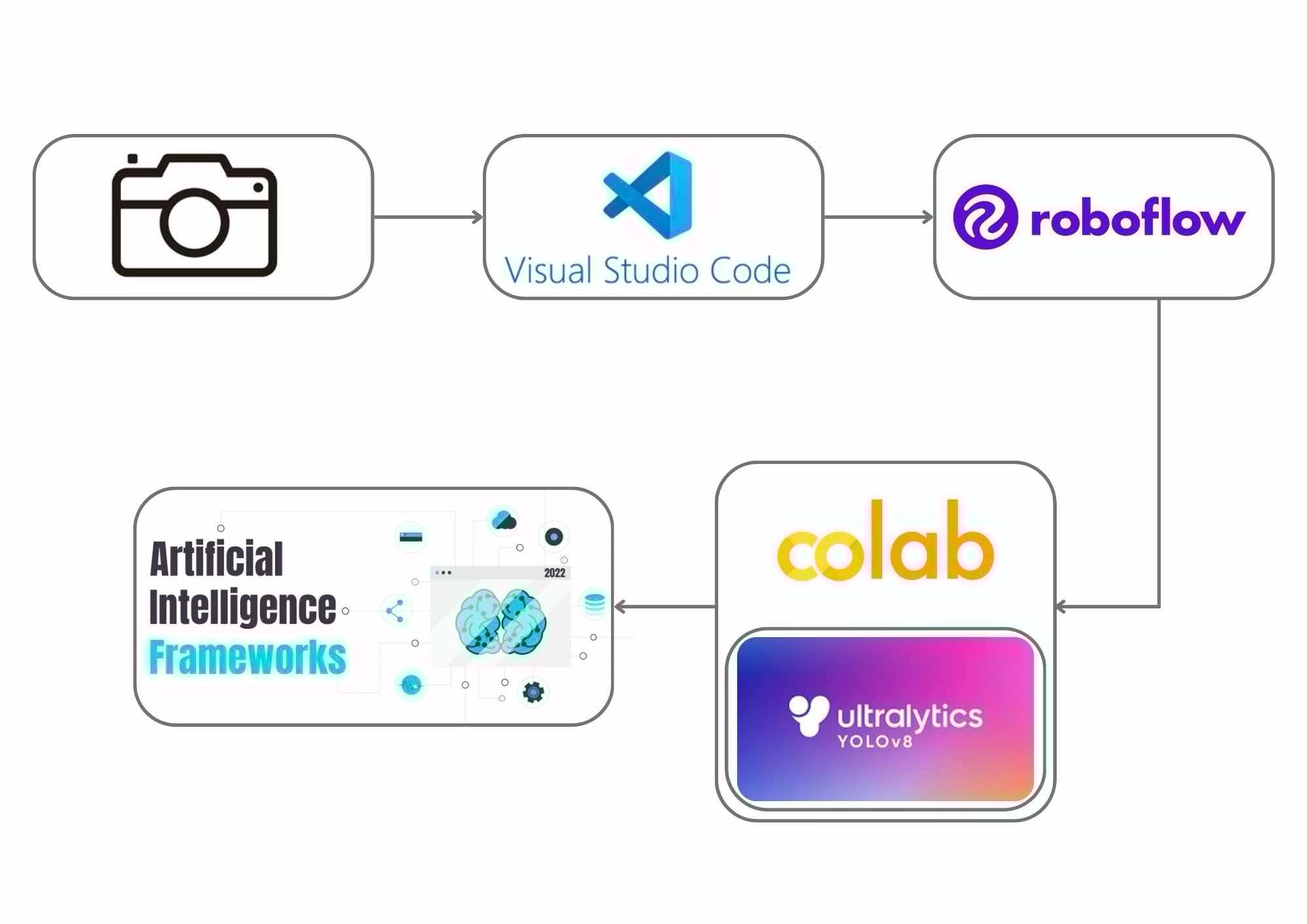
1. **การใช้ AI**:
   * การนำเทคโนโลยี AI เช่น Deep Learning มาประมวลผลข้อมูลภาพจากแหล่งน้ำ ช่วยเพิ่มความแม่นยำและความรวดเร็วในการตรวจจับและจำแนกประเภทของขยะลอยน้ำ เช่น ขวดพลาสติก กล่องโฟม หรือเศษวัสดุอื่น ๆ
   * การประยุกต์ใช้ AI ยังสามารถช่วยคาดการณ์ปริมาณขยะในพื้นที่ที่กำหนด ซึ่งช่วยในการวางแผนการจัดการได้ดียิ่งขึ้น
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 สำหรับการตรวจจับขยะลอยน้ำที่สามารถปรับแต่งได้ 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 โดยแสดงตามดัง รูปแผนผังสถาปัตยกรรมระบบ



ภาพ 4.1 แผนผังสถาปัตยกรรมระบบ

ซึ่งสามารถแบ่งและอธิบายรายละเอียดขั้นตอนการทำงานได้ ดังนี้

1. **การเก็บรวบรวมข้อมูล**  
   ทำการเก็บรวบรวมข้อมูลจากการถ่ายภาพแหล่งน้ำที่มีขยะ เช่น ขวดพลาสติก และภาพพื้นหลังที่ไม่มีขวดพลาสติก รวมถึงการนำข้อมูลภาพจากแหล่งข้อมูลออนไลน์ที่เกี่ยวข้อง เช่น Roboflow เพื่อใช้เป็นชุดข้อมูลเริ่มต้นสำหรับการพัฒนาโมเดลตรวจจับ
2. **การสร้างและจัดเตรียมชุดข้อมูล (Synthetic Dataset)**  
   นำข้อมูลภาพพื้นหลังและฟีเจอร์ (ขวดพลาสติก) มาสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) พร้อม Annotation โดยใช้เทคนิคการวางฟีเจอร์บนภาพพื้นหลังแบบสุ่ม และปรับปรุงคุณภาพข้อมูลด้วยการสร้าง Mask น้ำ เพื่อกำหนดพื้นที่น้ำในภาพให้เหมาะสม
3. **การจัดการข้อมูลและอัปโหลดเข้า Roboflow**  
   นำชุดข้อมูลที่สร้างขึ้นไปอัปโหลดเข้าสู่แพลตฟอร์ม Roboflow เพื่อทำการจัดการและปรับปรุง Annotation และใช้ฟีเจอร์ 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
   1. **การเก็บรวบรวมข้อมูล**

การเก็บรวบรวมข้อมูลในโครงการนี้มุ่งเน้นการเตรียมข้อมูลที่เหมาะสมสำหรับการพัฒนาโมเดลตรวจจับขวดพลาสติกบนพื้นน้ำ โดยมีการวางแผนขั้นตอนการดำเนินงานอย่างเป็นระบบ เพื่อให้ได้ข้อมูลที่มีคุณภาพสูงเพียงพอและครอบคลุมความหลากหลายของสถานการณ์จริงที่อาจเกิดขึ้นในแหล่งน้ำธรรมชาติ

**4.2.1 การถ่ายภาพฟีเจอร์ (ขวดพลาสติก)**

กระบวนการถ่ายภาพฟีเจอร์เริ่มจากการรวบรวมขวดพลาสติกหลากหลายประเภท เช่น ขวดใส ขวดที่มีฉลากสีต่าง ๆ และขวดที่มีรูปทรงหลากหลาย เพื่อให้ข้อมูลที่ได้ครอบคลุมสถานการณ์ที่พบได้ในแหล่งน้ำจริง ภาพถ่ายจะถูกถ่ายบนพื้นหลังสีเขียวเรียบเพื่อช่วยให้การแยกฟีเจอร์ออกจากพื้นหลังในขั้นตอนถัดไปทำได้ง่ายและแม่นยำ นอกจากนี้ ขวดพลาสติกแต่ละใบยังถูกถ่ายในมุมมองที่หลากหลาย ได้แก่ ด้านหน้า ด้านซ้าย ด้านขวา และด้านหลัง เพื่อเพิ่มความหลากหลายของชุดข้อมูลที่ใช้ในการฝึกโมเดล



ภาพ 4.2 ตัวอย่างภาพถ่ายขวดพลาสติกบนพื้นหลังสีเขียว

* + 1. **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 ตัวอย่างภาพฟีเจอร์ในรูปแบบไฟล์ PNG ที่มีพื้นหลังโปร่งใส

**4.2.3 การเก็บรวบรวมภาพพื้นหลัง**

ภาพพื้นหลังสำหรับโครงการนี้ถูกถ่ายจากแหล่งน้ำจริง เช่น แม่น้ำ ลำคลอง และบ่อน้ำ โดยเลือกภาพที่ไม่มีขวดพลาสติกหรือวัตถุอื่นในเฟรม เพื่อใช้เป็นพื้นฐานในการสร้างชุดข้อมูลจำลอง นอกจากนี้ยังมีการนำภาพจากแหล่งข้อมูลออนไลน์ที่มีคุณภาพสูง เช่น เว็บไซต์ที่ให้บริการภาพฟรี เพื่อเพิ่มจำนวนและความหลากหลายของภาพพื้นหลังที่ใช้ในโครงการ



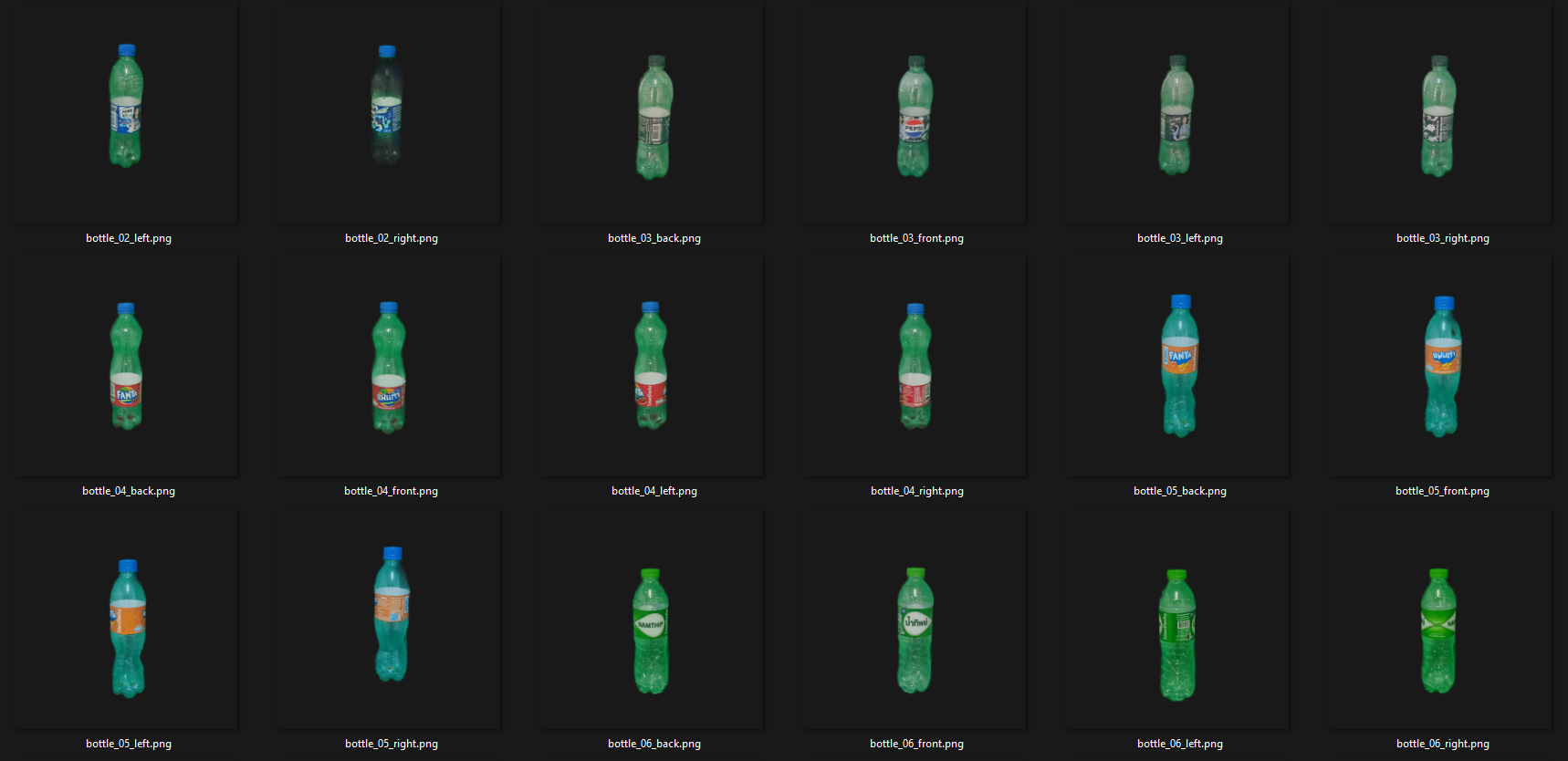
ภาพ 4.4 ตัวอย่างภาพพื้นหลังจากแหล่งข้อมูลออนไลน์

* + 1. **การจัดเก็บข้อมูล**

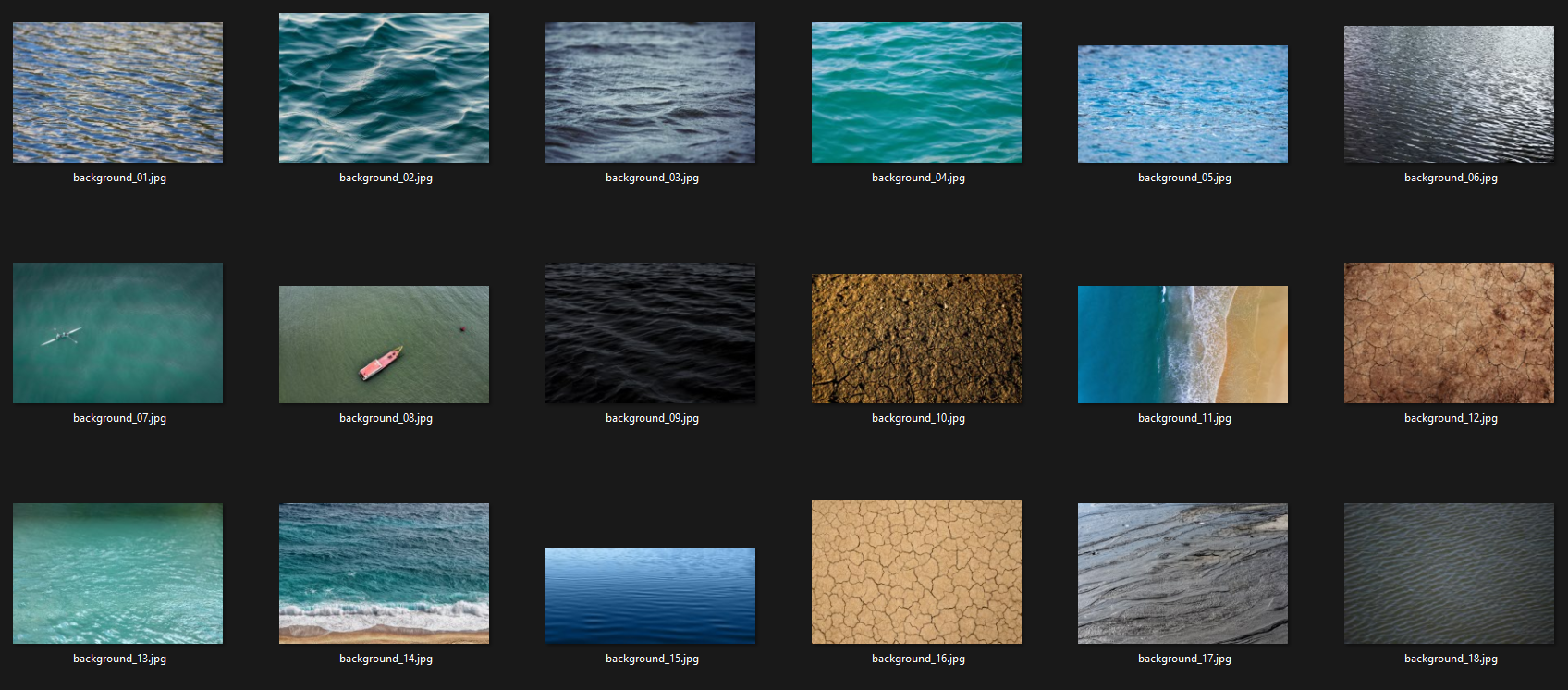
ภาพฟีเจอร์และพื้นหลังที่ได้จะถูกจัดเก็บอย่างเป็นระบบในโฟลเดอร์เฉพาะ ได้แก่:

* + โฟลเดอร์ features/: สำหรับเก็บภาพขวดพลาสติกพร้อมพื้นหลังโปร่งใส
  + โฟลเดอร์ backgrounds/: สำหรับเก็บภาพพื้นหลังจากแหล่งน้ำ

แต่ละไฟล์จะถูกตั้งชื่อให้สื่อความหมาย เช่น bottle\_01\_front.png สำหรับฟีเจอร์ หรือ background\_01.jpg สำหรับพื้นหลัง เพื่อความสะดวกในการเรียกใช้งานในขั้นตอนถัดไป



ภาพ 4.5 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ features/

ภาพ 4.6 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ backgrounds/

* + 1. **การตรวจสอบคุณภาพข้อมูล**

ข้อมูลทั้งหมดจะถูกตรวจสอบคุณภาพก่อนนำไปใช้ เพื่อให้มั่นใจว่า ไม่มีสิ่งรบกวน เช่น เงาสะท้อน หรือการแยกพื้นหลังที่ผิดพลาด ภาพมีความคมชัดและรายละเอียดเพียงพอสำหรับกระบวนการสร้างชุดข้อมูลจำลอง ข้อมูลที่ได้จากกระบวนการนี้จะถูกนำไปใช้ในขั้นตอนการสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) เพื่อสนับสนุนการพัฒนาโมเดลตรวจจับขวดพลาสติกในขั้นตอนถัดไป

* 1. **การสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset)**

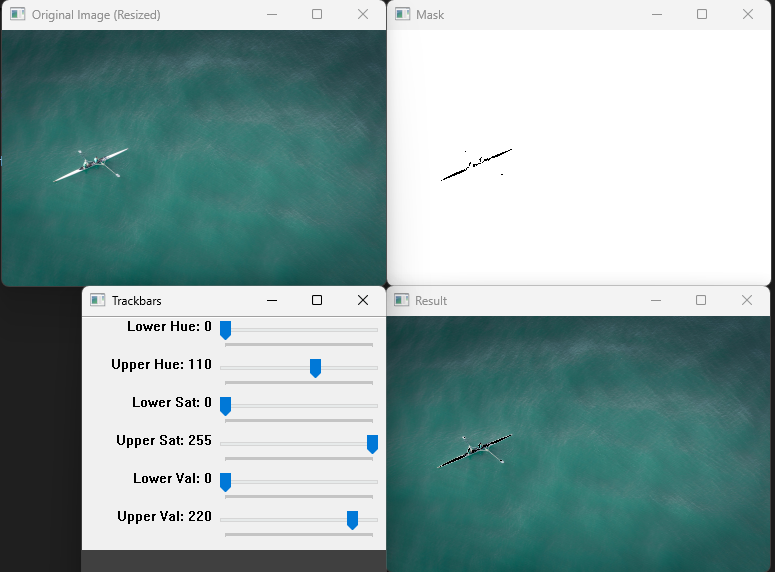
การสร้างชุดข้อมูลภาพจำลองในโครงการนี้ใช้กระบวนการเชิงโปรแกรมที่ออกแบบมาเพื่อเพิ่มปริมาณและความหลากหลายของข้อมูล ชุดข้อมูลที่สร้างขึ้นจะช่วยสนับสนุนการพัฒนาโมเดลตรวจจับขวดพลาสติกบนพื้นที่น้ำ โดยการวางฟีเจอร์ (ขวดพลาสติก) บนภาพพื้นหลังด้วยวิธีการสุ่มและการสร้าง Annotation อัตโนมัติ ซึ่งสามารถอธิบายรายละเอียดได้ดังนี้:

**4.3.1 การนำฟีเจอร์และภาพพื้นหลังมาใช้งาน**

ฟีเจอร์ที่แยกพื้นหลังแล้ว (ขวดพลาสติก) จะถูกนำมาจากโฟลเดอร์ features/ ในขณะที่ภาพพื้นหลังจะนำมาจากโฟลเดอร์ backgrounds/ สำหรับแต่ละรอบของการสร้างภาพจำลอง ฟีเจอร์และภาพพื้นหลังจะถูกสุ่มเลือกเพื่อสร้างความหลากหลายของชุดข้อมูล ช่วยให้ข้อมูลมีความครอบคลุมในลักษณะที่ใกล้เคียงกับสถานการณ์จริงในแหล่งน้ำ

**4.3.2 การตรวจจับพื้นที่น้ำ (Water Mask Generation)**

พื้นที่น้ำในภาพพื้นหลังถูกตรวจจับโดยใช้ฟังก์ชัน detect\_water\_area ที่พัฒนาด้วย OpenCV การตรวจจับเริ่มจากการแปลงภาพพื้นหลังเป็นโหมดสี HSV และใช้ค่าช่วงสี (HSV Range) ที่กำหนดไว้ (LOWER\_BOUND = [0, 0, 0] และ UPPER\_BOUND = [110, 255, 220]) เพื่อสร้าง Mask น้ำ ตัวกรอง Morphological เช่น Closing และ Opening ถูกใช้เพื่อลด Noise และปรับปรุงความแม่นยำของ Mask ซึ่ง Mask ที่ได้จะถูกใช้เพื่อกำหนดพื้นที่น้ำที่เหมาะสมสำหรับการวางฟีเจอร์



ภาพ 4.6 ตัวอย่างการค่าช่วงสี HSV ที่กำหนดเพื่อสร้าง Mask ของพื้นที่น้ำ

**4.3.3 การวางฟีเจอร์บนพื้นที่น้ำ**

การวางฟีเจอร์ (ขวดพลาสติก) บนพื้นที่น้ำในภาพพื้นหลังดำเนินการโดยฟังก์ชัน place\_feature\_on\_water ซึ่งสุ่มขนาด (Scale) ของฟีเจอร์ระหว่าง 0.08x ถึง 0.3x ของขนาดเดิม และสุ่มพิกัด (x, y) ในพื้นที่ Mask น้ำ ระบบจะตรวจสอบให้ฟีเจอร์ไม่หลุดจาก Mask น้ำ และกำหนดว่าฟีเจอร์อย่างน้อย 50% ต้องอยู่ในพื้นที่น้ำ

**4.3.4 การสร้างภาพจำลอง**

การรวมฟีเจอร์เข้ากับภาพพื้นหลังใช้ฟังก์ชัน overlay\_feature ซึ่งจะหมุนฟีเจอร์แบบสุ่ม (0° ถึง 360°) ด้วย Rotation Matrix จากนั้นใช้เทคนิค Alpha Blending เพื่อผสมฟีเจอร์กับภาพพื้นหลังให้ดูสมจริง



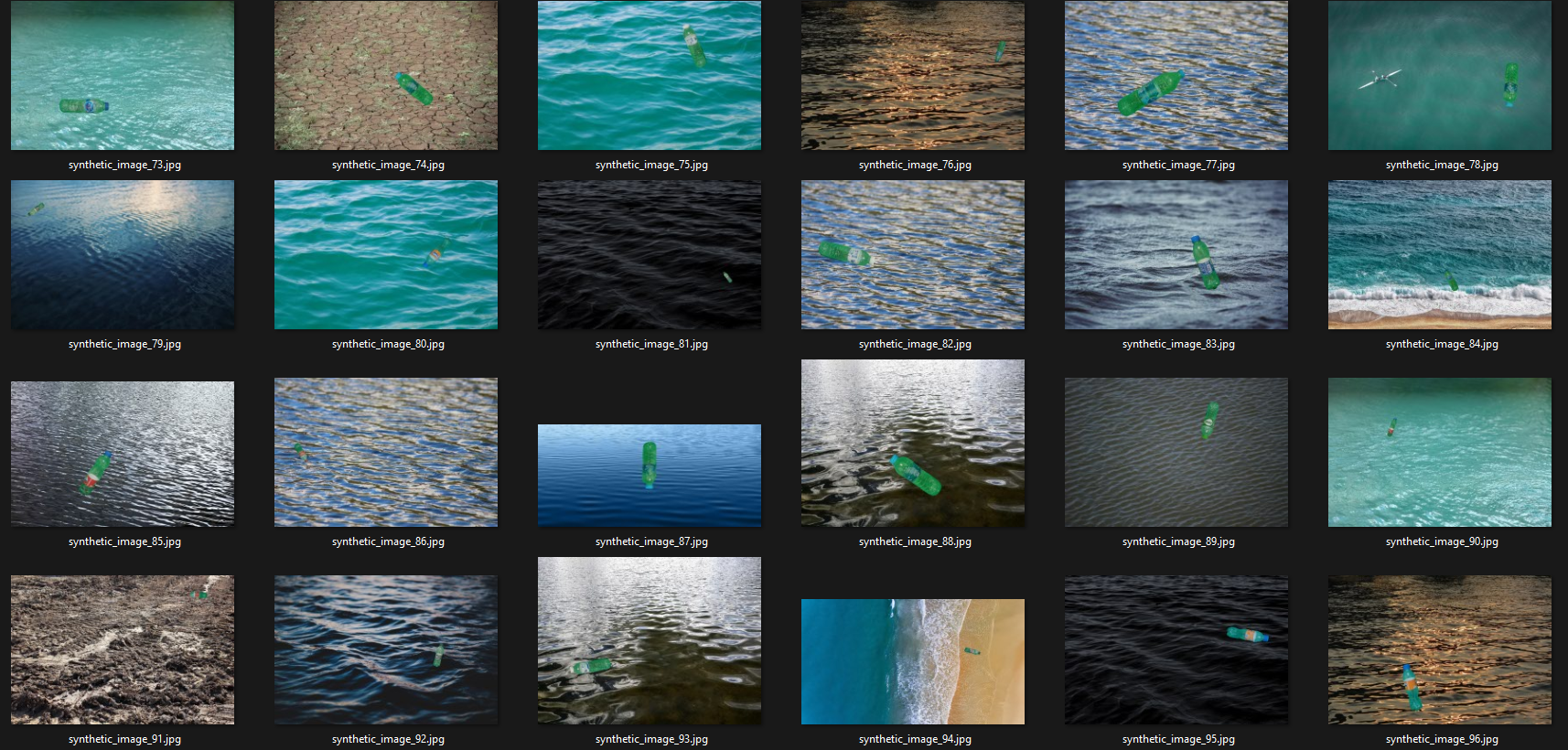
ภาพ 4.7 ตัวอย่างการวางฟีเจอร์บนพื้นที่น้ำโดยสุ่มขนาด พิกัด และมุมหมุน

* + 1. **การสร้าง Annotation**

Annotation สำหรับ Bounding Box ของฟีเจอร์ถูกสร้างโดยฟังก์ชัน save\_annotation ซึ่งบันทึกค่า x\_center, y\_center, width, และ height ในรูปแบบ YOLO ค่าเหล่านี้ถูก Normalize ตามขนาดของภาพ Annotation ที่สร้างขึ้นจะถูกจัดเก็บในโฟลเดอร์ annotations/ โดยมีชื่อไฟล์ตรงกับชื่อภาพ เช่น synthetic\_image\_1.txt

* + 1. **การจัดเก็บข้อมูล**

ภาพที่สร้างสำเร็จจะถูกจัดเก็บในโฟลเดอร์ synthetic\_dataset/ โดยตั้งชื่อไฟล์ในรูปแบบ synthetic\_image\_1.jpg หากการวางฟีเจอร์ไม่สำเร็จ เช่น ไม่มีพื้นที่น้ำเพียงพอ ระบบจะบันทึกภาพพื้นหลังและ Mask น้ำในโฟลเดอร์สำหรับการตรวจสอบและการปรับปรุงพารามิเตอร์ในอนาคต



ภาพ 4.8 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ synthetic\_dataset/

* + 1. **การตรวจสอบและประเมินผล**

ระบบจะบันทึกข้อมูลภาพที่ล้มเหลวในการวางฟีเจอร์ เพื่อใช้ในการวิเคราะห์และปรับปรุงพารามิเตอร์ เช่น การปรับช่วงค่า HSV หรือการขยายขนาด Mask น้ำ หลังจากระบบสร้างภาพครบตามจำนวนที่กำหนด (เช่น num\_images = 1000) ระบบจะหยุดทำงาน

ชุดข้อมูลภาพจำลองที่ได้จากขั้นตอนนี้จะมีความหลากหลายและสมจริง พร้อมทั้ง Annotation ในรูปแบบที่เหมาะสมสำหรับการฝึกโมเดล YOLO ในขั้นตอนถัดไป

* 1. **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**

ชุดข้อมูลที่สร้างขึ้นจากขั้นตอนก่อนหน้าประกอบด้วยไฟล์ภาพและไฟล์ 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 รองรับการฝึกโมเดล YOLOv8 ได้อย่างมีประสิทธิภาพและแม่นยำในขั้นตอนต่อไป

* 1. **การฝึกฝนแบบจำลอง YOLOv8**

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

**4.5.1 การเตรียมสภาพแวดล้อม**

กระบวนการฝึกโมเดลเริ่มต้นด้วยการเตรียมสภาพแวดล้อมบน Google Colab ซึ่งรองรับการประมวลผลด้วย GPU และเหมาะสำหรับการทำงานที่ใช้ทรัพยากรสูง ในขั้นตอนนี้ ระบบจะทำการติดตั้งไลบรารี 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 โมเดลนี้จะถูกนำไปใช้ในการตรวจจับขวดพลาสติกบนพื้นน้ำในระบบที่ออกแบบไว้

**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 (ขวดพลาสติก) และภาพพื้นหลังจากโฟลเดอร์ที่กำหนดมาใช้ ระบบจะสุ่มเลือกฟีเจอร์และภาพพื้นหลังในแต่ละรอบเพื่อเพิ่มความหลากหลาย หลังจากนั้น พื้นที่น้ำในภาพพื้นหลังจะถูกตรวจจับด้วย Mask ที่สร้างจากค่าช่วงสี HSV การวางฟีเจอร์บนพื้นที่น้ำจะดำเนินการโดยอัตโนมัติด้วยฟังก์ชันที่ตรวจสอบขนาด พิกัด และมุมหมุนของฟีเจอร์ เพื่อให้ได้ภาพที่ดูสมจริง ระบบจะสร้างไฟล์ Annotation ที่ตรงกับภาพโดยใช้ฟอร์แมต YOLO และจัดเก็บข้อมูลที่สร้างสำเร็จในโฟลเดอร์เฉพาะ

**4.6.2 การ Automate Data Augmentation บน Roboflow**

หลังจากสร้าง Synthetic Dataset แล้ว ระบบจะสนับสนุนการอัปโหลดชุดข้อมูลเข้าสู่ Roboflow เพื่อทำ Data Augmentation โดยอัตโนมัติ กระบวนการนี้ช่วยเพิ่มความหลากหลายของชุดข้อมูล 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:

* การสุ่มวางฟีเจอร์บนพื้นที่น้ำ
* 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