

**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

ชื่อเรื่อง การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุ

ประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก

ชื่อผู้พัฒนา ธีรธร รักษาเมือง

สุปวีณ์ สัญจร

ชญาดา เอียดทองใส

อาจารย์ที่ปรึกษา รศ.ดร. วัฒนพงศ์ เกิดทองมี

ปริญญาและสาขาวิชา หลักสูตรวิศวกรรมศาสตร์และเทคโนโลยี

สาขาวิชาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์

ปีการศึกษา 2567

# **บทคัดย่อ**

รายงานวิจัยนี้มีวัตถุประสงค์เพื่อการพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก มีเป้าหมายเพื่อแก้ไขข้อจำกัดของการใช้ชุดข้อมูลขนาดใหญ่ โดยโครงการนี้มุ่งเน้นการออกแบบและพัฒนาเฟรมเวิร์คที่สามารถสร้างชุดข้อมูลจำลอง (Synthetic Dataset) โดยอัตโนมัติ การประยุกต์ใช้เทคนิคการเพิ่มข้อมูล (Data Augmentation) และการฝึกโมเดลโดยใช้ YOLOv8 ที่สามารถตรวจจับวัตถุได้อย่างแม่นยำและรวดเร็วในสถานการณ์ที่มีข้อมูลจำกัด นอกจากนี้ ยังได้พัฒนาเครื่องมือที่ช่วยให้กระบวนการสร้างข้อมูลและฝึกโมเดลเป็นไปอย่างอัตโนมัติ โดยวัดผลสำเร็จจากค่าความแม่นยำเฉลี่ย (mAP), Precision, และ Recall ผลลัพธ์ที่ได้ช่วยลดต้นทุนและเพิ่มความสะดวกในการพัฒนาโมเดล 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**

รายงานวิจัยฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาและความเอาใจใส่อย่างยิ่งจาก 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 สัญจร

ชญาดา เอียดทองใส

# **สารบัญ**

**เรื่อง หน้า**

[บทคัดย่อ ก](#_Toc190545243)

[Abstract ข](#_Toc190545244)

[กิตติกรรมประกาศ ค](#_Toc190545245)

[สารบัญ ง](#_Toc190545246)

[สารบัญ(ต่อ) จ](#_Toc190545246)

[สารบัญ(ต่อ) ฉ](#_Toc190545246)

[สารบัญตาราง ช](#_Toc190545248)

[สารบัญภาพ ซ](#_Toc190545250)

[**บทที่ 1** 1](#_Toc190545252)

[**บทนำ** 1](#_Toc190545253)

[1.1 ความสำคัญและที่มาของปัญหา 1](#_Toc190545254)

[1.2 วัตถุประสงค์ 1](#_Toc190545255)

[1.3 ขอบเขตของงาน 1](#_Toc190545256)

[1.3.1 การเก็บข้อมูลและจัดการข้อมูล 1](#_Toc190545257)

[1.3.2. การพัฒนาและฝึกอบรมโมเดล 2](#_Toc190545258)

[1.3.3. การพัฒนาระบบอัตโนมัติ 2](#_Toc190545259)

[1.3.4. การประเมินผลและปรับปรุงระบบ 2](#_Toc190545260)

[1.3.5. ข้อจำกัดของโครงการ 2](#_Toc190545261)

[1.4 ประโยชน์ที่คาดว่าจะได้รับ 3](#_Toc190545262)

[1.5 ผลที่คาดว่าจะได้รับเมื่อเสร็จสิ้นโครงการ 3](#_Toc190545263)

[1.6 ขั้นตอนการดำเนินงาน 4](#_Toc190545264)

[1.7 อุปกรณ์ที่ใช้ในการพัฒนางาน 4](#_Toc190545265)

[1.7.1 ด้านซอฟต์แวร์ (Software) 4](#_Toc190545266)

[1.7.2 ด้านฮาร์ดแวร์ (Hardware) 5](#_Toc190545267)

[**บทที่ 2** 6](#_Toc190545268)

[**ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง** 6](#_Toc190545269)

[2.1 ทฤษฎีที่เกี่ยวข้อง 6](#_Toc190545270)

[2.1.1 การตรวจจับวัตถุ (Object Detection) 6](#_Toc190545271)

[2.1.2 การเพิ่มข้อมูล (Data Augmentation) 7](#_Toc190545272)

[2.1.3 Mean Average Precision (mAP) 9](#_Toc190545273)

[2.1.4 การประมวลผลภาพ (Image Processing) 11](#_Toc190545274)

[2.2 เทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ 13](#_Toc190545275)

[2.2.1 RoboFlow 13](#_Toc190545276)

[2.2.2 opencv-python 14](#_Toc190545277)

[2.2.3 TensorFlow 15](#_Toc190545278)

[2.3 งานวิจัยหรือระบบงานใกล้เคียง 17](#_Toc190545279)

[2.3.1 YOLO (You Only Look Once) 17](#_Toc190545280)

[2.3.2 การพัฒนาการตรวจจับรอยเปื้อนบนเสื้อผ้า (Development of Stain Detection on Clothes) 18](#_Toc190545281)

[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) 18](#_Toc190545282)

[2.4 เปรียบเทียบระบบงานใกล้เคียงกับระบบที่พัฒนา 19](#_Toc190545283)

[**บทที่ 3** 20](#_Toc190545284)

[**เอกสารความต้องการของระบบ** 20](#_Toc190545285)

[3.1 การวิเคราะห์ระบบ (System Analysis) 20](#_Toc190545286)

[3.1.1 ภาพรวมสถาปัตยกรรมระบบ (System Architecture Diagram) 20](#_Toc190545287)

[3.1.2 การศึกษาความต้องการของผู้ใช้งาน (User Requirement Analysis) 21](#_Toc190545288)

[3.2 การวิเคราะห์ระบบปัจจุบัน (Existing System Analysis) 26](#_Toc190545289)

[3.3 การระบุปัญหาและโอกาสพัฒนา (Problem and Opportunity Identification) 26](#_Toc190545290)

[**บทที่ 4** 28](#_Toc190545291)

[**การออกแบบระบบ** 28](#_Toc190545292)

[4.1 สถาปัตยกรรมของระบบ (System architecture) 28](#_Toc190545293)

[4.2 การเก็บรวบรวมข้อมูล 29](#_Toc190545294)

[4.2.1 การถ่ายภาพฟีเจอร์ (ขวดพลาสติก) 29](#_Toc190545295)

[4.2.2 การจัดการพื้นหลังของฟีเจอร์ (Feature Extraction) 30](#_Toc190545296)

[4.2.3 การเก็บรวบรวมภาพพื้นหลัง 31](#_Toc190545297)

[4.2.4 การจัดเก็บข้อมูล 31](#_Toc190545298)

[4.2.5 การตรวจสอบคุณภาพข้อมูล 32](#_Toc190545299)

[4.3 การสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) 33](#_Toc190545300)

[4.3.1 การนำฟีเจอร์และภาพพื้นหลังมาใช้งาน 33](#_Toc190545301)

[4.3.2 การตรวจจับพื้นที่น้ำ (Water Mask Generation) 33](#_Toc190545302)

[4.3.3 การวางฟีเจอร์บนพื้นที่น้ำ 34](#_Toc190545303)

[4.3.4 การสร้างภาพจำลอง 34](#_Toc190545304)

[4.3.5 การสร้าง Annotation 34](#_Toc190545305)

[4.3.6 การจัดเก็บข้อมูล 35](#_Toc190545306)

[4.3.7 การตรวจสอบและประเมินผล 35](#_Toc190545307)

[4.4 การอัปโหลดชุดข้อมูลเข้า Roboflow (สำหรับ YOLOv8) 35](#_Toc190545308)

[4.4.1 การจัดเตรียมชุดข้อมูล 36](#_Toc190545309)

[4.4.2 การสร้างโปรเจกต์ใน Roboflow 36](#_Toc190545310)

[4.4.3 การอัปโหลดชุดข้อมูล 36](#_Toc190545311)

[4.4.4 การปรับแต่งชุดข้อมูล 36](#_Toc190545312)

[4.4.5 การส่งออกชุดข้อมูล 36](#_Toc190545313)

[4.5 การฝึกฝนแบบจำลอง YOLOv8 37](#_Toc190545314)

[4.5.1 การเตรียมสภาพแวดล้อม 37](#_Toc190545315)

[4.5.2 การตั้งค่าและการโหลดชุดข้อมูล 37](#_Toc190545316)

[4.5.3 กระบวนการฝึกโมเดล 37](#_Toc190545317)

[4.5.4 การประเมินผลลัพธ์ของโมเดล 37](#_Toc190545318)

[4.5.5 การปรับปรุงโมเดล (Fine-tuning) 38](#_Toc190545319)

[4.5.6 การบันทึกและส่งออกโมเดล 38](#_Toc190545320)

[4.6 การพัฒนาเฟรมเวิร์ค AI 38](#_Toc190545321)

[4.6.1 การ Automate การสร้าง Synthetic Dataset 38](#_Toc190545322)

[4.6.2 การ Automate Data Augmentation บน Roboflow 38](#_Toc190545323)

[4.6.3 การเชื่อมโยงกระบวนการแบบ Automate 39](#_Toc190545324)

[4.6.4 การจัดเก็บและจัดการข้อมูล 39](#_Toc190545325)

[4.6.5 การปรับปรุงเฟรมเวิร์คในอนาคต 39](#_Toc190545326)

[**บรรณานุกรม** 40](#_Toc190545327)

# **สารบัญตาราง**

ตาราง หน้า

[ตารางที่ 3.1 ตารางแสดงความต้องการของผู้ใช้ 30](#_Toc190032572)

[ตารางที่ 3.2 ตารางแสดงคุณลักษณะของ UC01 32](#_Toc190032573)

[ตารางที่ 3.3 ตารางแสดงคุณลักษณะของ UC02 33](#_Toc190032574)

[ตารางที่ 3.4 ตารางแสดงคุณลักษณะของ UC03 33](#_Toc190032575)

[ตารางที่ 3.5 ตารางแสดงคุณลักษณะของ UC04 34](#_Toc190032576)

[ตารางที่ 3.6 ตารางแสดงคุณลักษณะของ UC05 34](#_Toc190032577)

# **สารบัญภาพ**

ภาพ หน้า

[ภาพที่ 1.1 การทำ Augmented Images 17](#_Toc190032065)

[ภาพที่ 3.1 แผนผังสถาปัตยกรรมระบบ 37](#_Toc190032066)

[ภาพที่ 4.3 ตัวอย่างภาพฟีเจอร์ในรูปแบบไฟล์ PNG ที่มีพื้นหลังโปร่งใส 39](#_Toc190032067)

[ภาพที่ 4.5 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ features/ 41](#_Toc190032068)

[ภาพที่ 4.6 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ backgrounds/ 41](#_Toc190032069)

[ภาพที่ 4.7 ตัวอย่างการวางฟีเจอร์บนพื้นที่น้ำโดยสุ่มขนาด พิกัด และมุมหมุน 43](#_Toc190032070)

[ภาพที่ 4.9 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ synthetic\_dataset/ 44](#_Toc190032071)

บทที่ 1

# **บทนำ**

1.1 ความสำคัญและที่มาของปัญหา

ปัจจุบัน โมเดลตรวจจับวัตถุที่มีประสิทธิภาพสูงส่วนใหญ่มักพัฒนามาจากชุดข้อมูลขนาดใหญ่ ซึ่งมีความหลากหลายของข้อมูลสูง ชุดข้อมูลเหล่านี้ช่วยให้โมเดลสามารถเรียนรู้รูปแบบและความแตกต่างของวัตถุได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม การจัดเตรียมชุดข้อมูลขนาดใหญ่นั้นมีต้นทุนสูง ทั้งในด้านเวลา แรงงาน และทรัพยากรในการเก็บรวบรวมและจัดการข้อมูล ในทางตรงกันข้าม การพัฒนาโมเดลจากชุดข้อมูลขนาดเล็กนั้นมีข้อจำกัดหลายประการ โดยเฉพาะในด้านความแม่นยำ และความหลากหลายของข้อมูลที่ไม่ดีเท่าชุดข้อมูลขนาดใหญ่ ซึ่งประสิทธิภาพของโมเดลที่พัฒนาจากชุดข้อมูลขนาดเล็กอาจลดลงเมื่อเผชิญกับสภาพแวดล้อมที่ซับซ้อน ความท้าทายนี้จึงเป็นจุดเริ่มต้นของโครงการนี้ ที่มุ่งเน้น การพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุที่มีประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็ก

ส่วนสาเหตุที่เรามุ่งเน้นไปในการตรวจจับขวดพลาสติกบนน้ำนั้น เนื่องจากปัญหาขยะลอยน้ำ โดยเฉพาะขวดพลาสติกที่ถูกทิ้งลงแหล่งน้ำธรรมชาติ เป็นหนึ่งในประเด็นสำคัญที่ส่งผลกระทบอย่างร้ายแรงต่อสิ่งแวดล้อมและระบบนิเวศ ขวดพลาสติกที่สะสมในแม่น้ำ ลำคลอง หรือทะเล ไม่เพียงแต่ทำให้แหล่งน้ำสกปรก แต่ยังเป็นอันตรายต่อสิ่งมีชีวิต และยังส่งผลกระทบต่อคุณภาพน้ำที่ใช้ในชีวิตประจำวันของมนุษย์

แม้ว่าจะมีความพยายามในการเก็บรวบรวมขยะจากแหล่งน้ำ แต่การทำงานด้วยแรงงานมนุษย์ล้วน ๆ มีข้อจำกัดดังนั้นการตรวจจับขยะลอยน้ำโดยใช้ AI เป็นทางเลือกที่ช่วยเพิ่มประสิทธิภาพในการจัดการขยะเหล่านี้ได้ อย่างไรก็ตาม การพัฒนาโมเดล AI ที่สามารถตรวจจับขยะลอยน้ำได้อย่างแม่นยำ ยังคงเผชิญกับปัญหาหลักสองประการ ได้แก่ การขาดแคลนชุดข้อมูลที่เหมาะสมสำหรับการฝึกโมเดล และความยากลำบากในการตรวจจับขยะในสภาพแวดล้อมที่หลากหลาย เช่น เงาสะท้อนบนผิวน้ำ หรือสีที่กลมกลืนระหว่างขวดพลาสติกและพื้นน้ำ

1.2 วัตถุประสงค์

1. พัฒนาแนวทางประยุกต์ที่เหมาะสมในการขยายจำนวนของชุดข้อมูลภาพ
2. สร้างโมเดล AI ที่สามารถตรวจจับขวดพลาสติกบนพื้นน้ำได้ด้วยชุดข้อมูลจำลองและมีความแม่นยำไม่ต่ำกว่า 80% เมื่อเทียบกับโมเดลที่สร้างจากชุดข้อมูลจริง
3. พัฒนาเฟรมเวิร์ค AI เพื่อการสร้างชุดข้อมูลจำลองแบบอัตโนมัติ

1.3 ขอบเขตของงาน

1.3.1 การเก็บข้อมูลและจัดการข้อมูล

* เก็บข้อมูลภาพขวดพลาสติกที่ถ่ายในบริบทที่ขวดพลาสติกลอยอยู่บนน้ำ
* 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 ที่สามารถวางฟีเจอร์ขวดพลาสติกบนพื้นที่น้ำในภาพพื้นหลัง
* เตรียมระบบสำหรับการนำข้อมูลเข้าแพลตฟอร์ม RoboFlow เพื่อจัดการชุดข้อมูลและการฝึกโมเดล

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 และเปิดโอกาสใหม่ในการสร้างผลิตภัณฑ์ที่มีประสิทธิภาพสูงยิ่งขึ้น

1.5 ผลที่คาดว่าจะได้รับเมื่อเสร็จสิ้นโครงการ

เมื่อโครงการการพัฒนาแนวทางอัตโนมัติในการสร้างโมเดลตรวจจับวัตถุประสิทธิภาพสูงจากชุดข้อมูลขนาดเล็กเสร็จสิ้นลง คาดว่าจะได้รับผลลัพธ์ที่มีความสำคัญหลายประการ ดังนี้:

1. แนวทางประยุกต์ที่เหมาะสมในการขยายจำนวนของชุดข้อมูลภาพ
2. โมเดล AI ที่สามารถตรวจจับขวดพลาสติกบนพื้นน้ำได้ด้วยชุดข้อมูลจำลองและมีความแม่นยำไม่ต่ำกว่า 80% เมื่อเทียบกับโมเดลที่สร้างจากชุดข้อมูลจริง
3. เฟรมเวิร์ค AI เพื่อการสร้างชุดข้อมูลจำลองแบบอัตโนมัติ

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 ทฤษฎีที่เกี่ยวข้อง

### **2.1.1 การตรวจจับวัตถุ (Object Detection)**

การตรวจจับวัตถุ (Object Detection) เป็นเทคนิคในวิทยาการคอมพิวเตอร์ที่ผสานการจำแนกภาพ (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 และดึงคุณลักษณะออกจากแต่ละส่วนเพื่อตรวจจับรูปแบบของวัตถุต่างๆ
* การจำแนกประเภท: แต่ละส่วนของภาพจะถูกจำแนกประเภทตามคุณลักษณะที่สกัดออกมา งานการจำแนกประเภทจะดำเนินการโดยใช้ SVM หรือเครือข่ายประสาทเทียมอื่นๆ ที่คำนวณความน่าจะเป็นของแต่ละหมวดหมู่ที่ปรากฏในบริเวณนั้น
* การระบุตำแหน่ง: พร้อมกันกับกระบวนการจำแนกประเภท แบบจำลองจะกำหนดกรอบขอบเขตสำหรับวัตถุที่ตรวจพบแต่ละชิ้น ซึ่งเกี่ยวข้องกับการคำนวณพิกัดสำหรับกรอบที่ล้อมรอบวัตถุแต่ละชิ้น จึงสามารถระบุตำแหน่งภายในภาพได้อย่างแม่นยำ
* การระงับค่าที่ไม่ใช่ค่าสูงสุด: เมื่อแบบจำลองระบุกรอบขอบเขตหลายกรอบสำหรับวัตถุเดียวกัน การระงับค่าที่ไม่ใช่ค่าสูงสุดจะถูกใช้เพื่อจัดการกับการทับซ้อนเหล่านี้ เทคนิคนี้จะเก็บเฉพาะกรอบขอบเขตที่มีคะแนนความเชื่อมั่นสูงสุดเท่านั้น และจะลบกรอบขอบเขตอื่นๆ ที่ทับซ้อนกันออกไป
* ผลลัพธ์: กระบวนการสิ้นสุดลงด้วยภาพต้นฉบับที่ถูกทำเครื่องหมายด้วยกรอบขอบเขตและป้ายกำกับที่แสดงวัตถุที่ตรวจพบและหมวดหมู่ที่เกี่ยวข้อง

**วิธีการเรียนรู้เชิงลึกสำหรับการตรวจจับวัตถุ มีวิธีการตรวจจับวัตถุอยู่ 2 ประเภทหลัก:**

**Two-stage Object Detection:** จะแบ่งการทำงานเป็นสองขั้นตอน: ขั้นตอนแรกจะเสนอภูมิภาคที่เป็นตัวเลือก จากนั้นจึงจัดหมวดหมู่ภูมิภาคนั้น เครื่องตรวจจับสองขั้นตอนบางประเภท ได้แก่ R-CNN, Fast R-CNN และ Fastพ R-CNN

**Single-Stage Detectors for Object Detection** จะคาดการณ์กล่องขอบเขตและความน่าจะเป็นของคลาสสำหรับทุกพื้นที่ของภาพได้อย่างแม่นยำ เน้นที่การรวมงานการระบุตำแหน่งวัตถุและการจำแนกประเภทเข้าในเครือข่ายประสาทเทียมแบบผ่านครั้งเดียว

### **2.1.2 การเพิ่มข้อมูล (Data Augmentation)**

การทำ Augmented Images เพื่อที่จะปรับปรุงประสิทธิภาพของโมเดล ซึ่งเป็นขั้นตอนที่การ augmentations จะถูกนำไปใช้กับรูปภาพที่มีอยู่ในชุดข้อมูล กระบวนการนี้สามารถช่วยปรับปรุงความแม่นยำของโมเดล เมื่อโมเดลชุดข้อมูลรูปภาพใหม่ที่ไม่เคยเห็น โดยการ augmentations ภาพมีดังต่อไปนี้: พลิกกลับ หมุน 90 องศา หมุนแบบสุ่ม ครอบตัดแบบสุ่ม 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

A collage of a parrot

Description automatically generated

ภาพที่ 1.1 การทำ Augmented Images

https://albumentations.ai/docs/images/introduction/image\_augmentation/augmentation.jpg

### **2.1.3 Mean Average Precision (mAP)**

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

โดยที่ mAP จะใช้เกณฑ์ย่อยดังต่อไปนี้

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

* True Positive (TP) = สิ่งที่ทำนาย ตรงกับสิ่งที่เกิดขึ้นจริง ในกรณี ทำนายว่าจริง และสิ่งที่เกิดขึ้น ก็คือ จริง
* True Negative (TN) = สิ่งที่ทำนายตรงกับสิ่งที่เกิดขึ้น ในกรณี ทำนายว่า ไม่จริง 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)

(2)

โดยที่:

Area of Overlap: พื้นที่ที่กรอบขอบเขตที่คาดการณ์ไว้และกรอบขอบเขตของข้อมูลพื้นฐานทับซ้อนกัน

Area of Union: พื้นที่ทั้งหมดที่ครอบคลุมโดยทั้งกรอบขอบเขตที่คาดการณ์ไว้และกรอบขอบเขตของข้อมูลพื้นฐาน ไม่รวมพื้นที่ทับซ้อน โดยจะสามารถแสดงเป็นสูตรทางคณิตศาสตร์ ดังสมการที่ (3)

*(3)*

โดยที่

A คือกรอบขอบเขตพื้นฐาน

B คือกรอบขอบเขตที่คาดการณ์ไว้

คือพื้นที่จุดตัดของทั้งสอง

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 มักถูกละเลยหรือนำมารวมเฉลี่ยไปกับปัจจัยอื่น ๆ โดยเน้นไปที่ความมั่นใจในกฎหรือตัวจำแนกเป็นหลักดังสมการที่ (4)

**Recall = TP / RP = A / (A + C)** (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 เพราะแสดงให้เห็นว่าคำทำนายบวกที่ทำไปนั้นมีความถูกต้องแม่นยำเพียงใด ในการระบุกรณีบวก โดยเป็นสัดส่วนระหว่างความสำเร็จ (True Positive) กับความพยายามทั้งหมด (Predicted Positive) อย่างไรก็ตาม ใน ROC Analysis, Precision มักไม่ได้รับการพิจารณาโดยตรง ดังสมการที่ (5)

**Precision = TP / PP = A / (A + B)** (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 ที่พิกัดคู่ใดๆ (x,y) เรียกว่าความเข้มของภาพนั้น ณ จุดนั้น เมื่อค่า x, y และแอมพลิจูดของ F มีจำกัด เราจะเรียกว่าภาพดิจิทัล กล่าวอีกนัยหนึ่ง ภาพสามารถกำหนดได้โดยอาร์เรย์สองมิติที่จัดเรียงเป็นแถวและคอลัมน์โดยเฉพาะ

ภาพดิจิทัลประกอบด้วยองค์ประกอบจำนวนจำกัด โดยแต่ละองค์ประกอบจะมีค่าเฉพาะที่ตำแหน่งเฉพาะ องค์ประกอบเหล่านี้เรียกว่าองค์ประกอบภาพ องค์ประกอบภาพ และพิกเซล 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):

ใช้สำหรับการแยกและวิเคราะห์ส่วนประกอบในภาพ 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 การตรวจจับรูปแบบ 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)
* การครอป (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 จะสามารถค้นหารูปร่างของเทนเซอร์ได้เฉพาะในเวลาที่ดำเนินการกราฟเท่านั้น

คุณลักษณะของ TensorFlow

* การแยกความแตกต่างโดยอัตโนมัติ (AutoDifferentiation) TensorFlow มีฟีเจอร์การแยกความแตกต่างโดยอัตโนมัติ ซึ่งช่วยคำนวณเวกเตอร์การไล่ระดับ (gradient) ของโมเดลได้อย่างมีประสิทธิภาพ โดยเฉพาะในอัลกอริทึมอย่างแบ็กโพรพาเกชัน (Backpropagation) ที่ใช้ในการฝึกโมเดล เฟรมเวิร์กจะติดตามลำดับการดำเนินการของเทนเซอร์อินพุตและคำนวณการไล่ระดับตามพารามิเตอร์ที่เหมาะสม
* การดำเนินการอย่างกระตือรือร้น (Eager Execution) TensorFlow รองรับโหมดการดำเนินการอย่างกระตือรือร้น ซึ่งทำให้การดำเนินการสามารถประเมินผลได้ทันที โดยไม่ต้องสร้างกราฟการคำนวณล่วงหน้า การทำงานในโหมดนี้ช่วยให้โค้ดอ่านง่ายขึ้นและแก้ไขข้อผิดพลาดได้ง่ายขึ้น เนื่องจากสามารถตรวจสอบผลลัพธ์ทีละขั้นตอนผ่านดีบักเกอร์ได้
* การกระจายการคำนวณ (Distribution) TensorFlow สนับสนุนการคำนวณแบบกระจายบนหลายอุปกรณ์ เช่น CPU, GPU หรือ TPU ผ่าน API และกลยุทธ์การกระจายที่หลากหลาย ซึ่งช่วยเพิ่มประสิทธิภาพในการฝึกอบรมและประเมินโมเดล โดยเฉพาะสำหรับโมเดลขนาดใหญ่
* ฟังก์ชันการสูญเสีย (Loss Functions) TensorFlow มีฟังก์ชันการสูญเสียหลากหลาย เช่น ข้อผิดพลาดกำลังสองเฉลี่ย (MSE) และเอนโทรปีแบบไบนารีครอส (BCE) 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) ใช้โมเดลที่ฝึกแล้วเพื่อทำนายผลลัพธ์สำหรับข้อมูลที่ไม่เคยเห็นมาก่อน

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

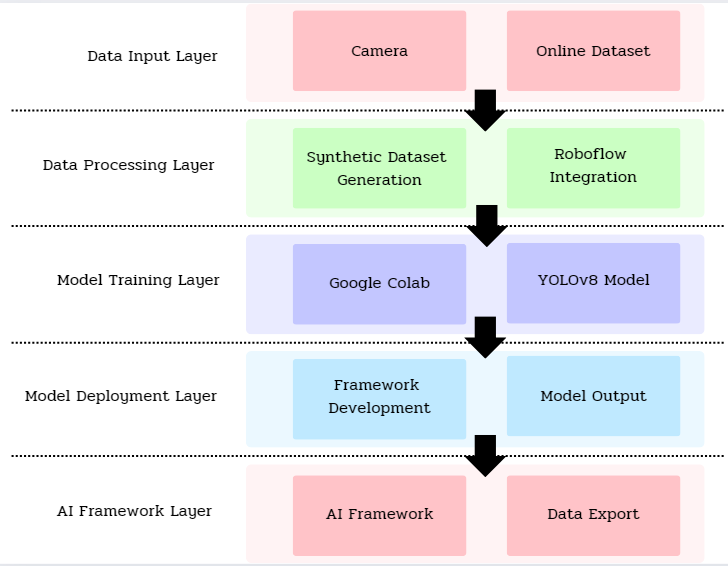
# **เอกสารความต้องการของระบบ**

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

โครงการนี้มีเป้าหมายเพื่อสนับสนุนการจัดการปัญหามลพิษในแหล่งน้ำอย่างยั่งยืน โดยเน้นการพัฒนาโมเดลที่ช่วยในการตรวจจับและแยกแยะขวดพลาสติกจากภาพถ่าย นำไปสู่การลดปริมาณขยะพลาสติกในแหล่งน้ำโดยการใช้ระบบอัตโนมัติ สนับสนุนการป้องกันและแก้ปัญหามลพิษในสิ่งแวดล้อมด้วยเทคโนโลยี Deep Learning สร้างแนวทางการใช้ปัญญาประดิษฐ์เพื่อเพิ่มประสิทธิภาพในกระบวนการตรวจจับและเก็บขวดพลาสติก ส่งเสริมการประยุกต์ใช้เทคโนโลยีปัญญาประดิษฐ์ในด้านสิ่งแวดล้อมอย่างมีประสิทธิภาพ

3.1 การวิเคราะห์ระบบ (System Analysis)

### **3.1.1 ภาพรวมสถาปัตยกรรมระบบ (System Architecture Diagram)**

****

ภาพที่ 3.1 System Architecture Diagram ของทั้งระบบ

### **3.1.2 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)

**ตารางที่ 3.2** ตารางแสดงคุณลักษณะของ UC01

|  |  |  |
| --- | --- | --- |
| **คุณลักษณะของ UC01:**  การสร้างชุดข้อมูลภาพจำลองและการจัดการข้อมูล (Synthetic Dataset and Data Management) | | |
| **คำอธิบาย:**  ผู้พัฒนาสามารถสร้างชุดข้อมูลภาพจำลอง พร้อม Annotation และสามารถจัดการชุดข้อมูลภาพที่สร้างขึ้นได้ เช่น เพิ่ม ลบ หรือแก้ไขภาพและ Annotation | | |
| **Actor:** ผู้พัฒนา | | |
| **เงื่อนไข (ข้อกำหนด):**  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)

**ตารางที่ 3.3** ตารางแสดงคุณลักษณะของ UC02

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

**ตารางที่ 3.4** ตารางแสดงคุณลักษณะของ UC03

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

**ตารางที่ 3.5** 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)

**ตารางที่ 3.6** 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 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:**

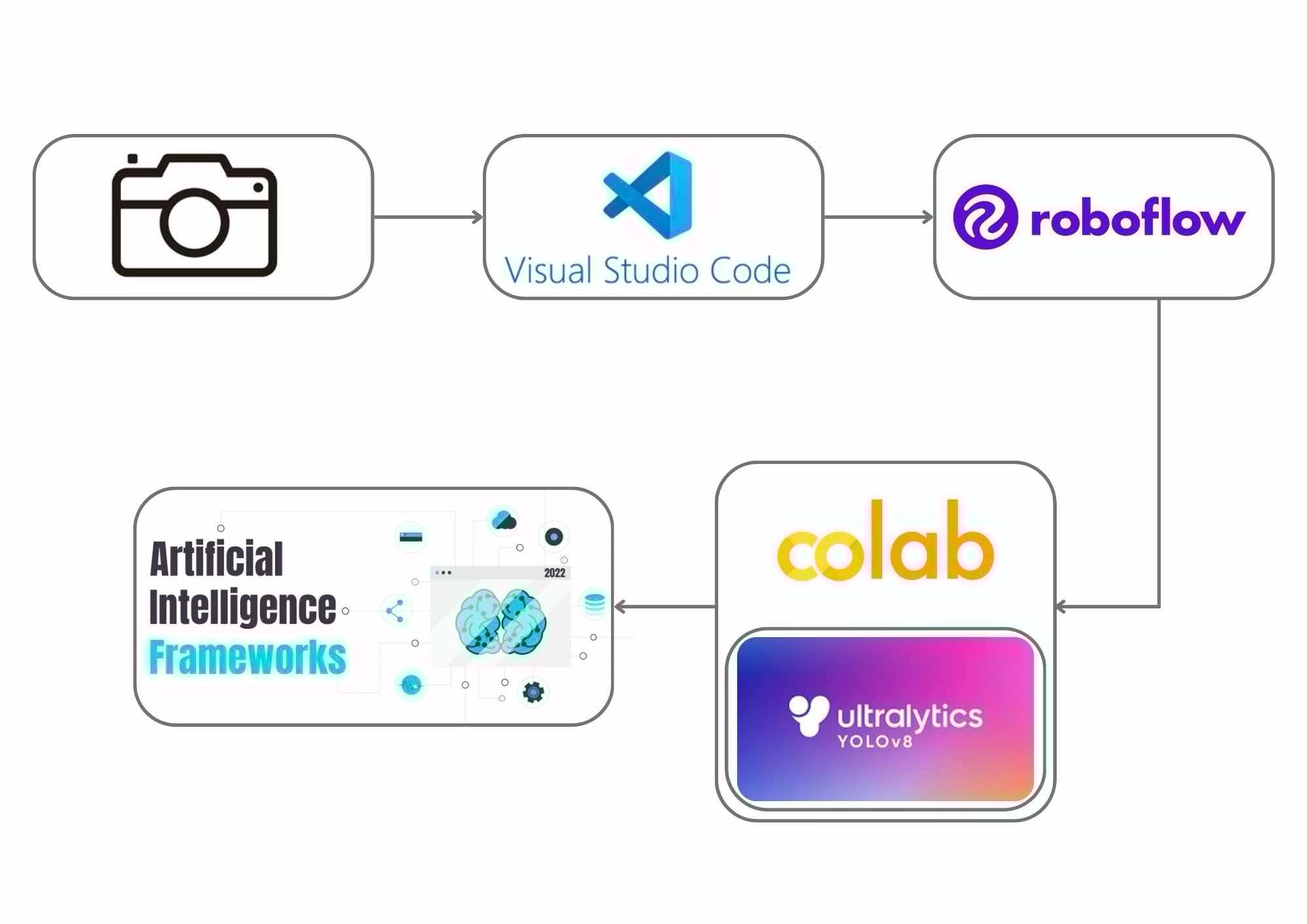
1. **การใช้ AI**:
   * การนำเทคโนโลยี AI เช่น Deep Learning มาประมวลผลข้อมูลภาพจากแหล่งน้ำ ช่วยเพิ่มความแม่นยำและความรวดเร็วในการตรวจจับและจำแนกประเภทของขยะลอยน้ำ เช่น ขวดพลาสติก กล่องโฟม หรือเศษวัสดุอื่น ๆ
   * การประยุกต์ใช้ AI ยังสามารถช่วยคาดการณ์ปริมาณขยะในพื้นที่ที่กำหนด ซึ่งช่วยในการวางแผนการจัดการได้ดียิ่งขึ้น
2. **Roboflow Integration**:
   * การใช้แพลตฟอร์มอย่าง Roboflow ในการจัดการและปรับปรุงคุณภาพของข้อมูลภาพ เช่น การทำ Data Augmentation เพื่อเพิ่มความหลากหลายของข้อมูลสำหรับการฝึกโมเดล
   * Roboflow ยังช่วยให้สามารถจัดรูปแบบ Annotation และสร้าง Dataset ที่เหมาะสมสำหรับการพัฒนาโมเดล AI ได้อย่างสะดวก
3. **การพัฒนา Framework สำหรับการสร้างชุดข้อมูลจำลอง:**
   * การสร้าง Framework สำหรับการสร้างชุดข้อมูลจำลองโดยอัตโนมัติ
   * Framework นี้จะช่วยให้ผู้ใช้งานสามารถสร้างชุดข้อมูลจำลองได้อย่างรวดเร็วโดยที่ไม่จำเป็นต้องจัดการข้อมูลด้วยตัวเอง
   * การออกแบบ Framework ที่สามารถรองรับการขยายตัวในอนาคต เช่น การเพิ่มฟีเจอร์สำหรับการวิเคราะห์ข้อมูลเพิ่มเติม
4. **การบูรณาการข้อมูล**:
   * ระบบที่รวมข้อมูลภาพ ข้อมูลการตรวจจับ และข้อมูลวิเคราะห์ไว้ในที่เดียว ช่วยให้สามารถวิเคราะห์ข้อมูลเชิงลึกและให้คำแนะนำที่เหมาะสมสำหรับการจัดการขยะ
   * การแสดงผลข้อมูลในรูปแบบรายงานหรือสถิติ ช่วยให้ผู้ใช้งานสามารถติดตามและประเมินผลการจัดการขยะได้ง่ายขึ้น

บทที่ 4

# **การออกแบบระบบ**

## **4.1 สถาปัตยกรรมของระบบ (System architecture)**

ในการศึกษาครั้งนี้ ผู้พัฒนาได้ออกแบบขอบเขตขั้นตอนการดำเนินงานไว้ดังต่อไปนี้ โดยแสดงตามดัง รูปแผนผังสถาปัตยกรรมระบบ



ภาพที่ 3.1 แผนผังสถาปัตยกรรมระบบ

ซึ่งสามารถแบ่งและอธิบายรายละเอียดขั้นตอนการทำงานได้ ดังนี้

1. **การเก็บรวบรวมข้อมูล**

ทำการเก็บรวบรวมข้อมูลจากการถ่ายภาพแหล่งน้ำที่มีขยะในบริเวณรอบมหาวิทยาลัยวลัยลักษณ์ เช่น ขวดพลาสติก และภาพพื้นหลังที่ไม่มีขวดพลาสติก รวมถึงการนำข้อมูลภาพจากแหล่งข้อมูลออนไลน์ที่เกี่ยวข้อง เช่น Roboflow เพื่อใช้เป็นชุดข้อมูลเริ่มต้นสำหรับการพัฒนาโมเดลตรวจจับ

1. **การสร้างและจัดเตรียมชุดข้อมูล (Synthetic Dataset)**

นำข้อมูลภาพพื้นหลังและฟีเจอร์ (ขวดพลาสติก) มาสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset) พร้อม Annotation โดยใช้เทคนิคการวางฟีเจอร์บนภาพพื้นหลังแบบสุ่ม และปรับปรุงคุณภาพข้อมูลด้วยการสร้าง Mask น้ำ เพื่อกำหนดพื้นที่น้ำในภาพให้เหมาะสม

1. **การจัดการข้อมูลและอัปโหลดเข้า Roboflow**

นำชุดข้อมูลที่สร้างขึ้นไปอัปโหลดเข้าสู่แพลตฟอร์ม Roboflow เพื่อทำการจัดการและปรับปรุง Annotation และใช้ฟีเจอร์ Data Augmentation เพื่อเพิ่มความหลากหลายของชุดข้อมูล ชุดข้อมูลจะถูกส่งออกในฟอร์แมตที่เหมาะสมสำหรับการฝึกโมเดล YOLO

1. **การฝึกฝนแบบจำลอง YOLO**

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

1. **การพัฒนาเฟรมเวิร์ค AI**

เมื่อได้โมเดลที่มีคุณภาพสูงสุด ระบบจะถูกพัฒนาเป็นเฟรมเวิร์ค AI โดยใช้ Visual Studio Code เพื่อออกแบบโครงสร้างการทำงานและการใช้งาน เฟรมเวิร์คดังกล่าวจะสามารถสร้างชุดข้อมูลจำลองสำหรับการเทรนโมเดลคุณภาพสูงให้โดยอัตโนมัติตามค่าที่ตั้งไว้

## **4.2 การเก็บรวบรวมข้อมูล**

การเก็บรวบรวมข้อมูลในโครงการนี้มุ่งเน้นการเตรียมข้อมูลที่เหมาะสมสำหรับการพัฒนาโมเดลตรวจจับขวดพลาสติกบนพื้นน้ำ โดยมีการวางแผนขั้นตอนการดำเนินงานอย่างเป็นระบบ เพื่อให้ได้ข้อมูลที่มีคุณภาพสูงเพียงพอและครอบคลุมความหลากหลายของสถานการณ์จริงที่อาจเกิดขึ้นในแหล่งน้ำธรรมชาติ

### **4.2.1 การถ่ายภาพฟีเจอร์ (ขวดพลาสติก)**

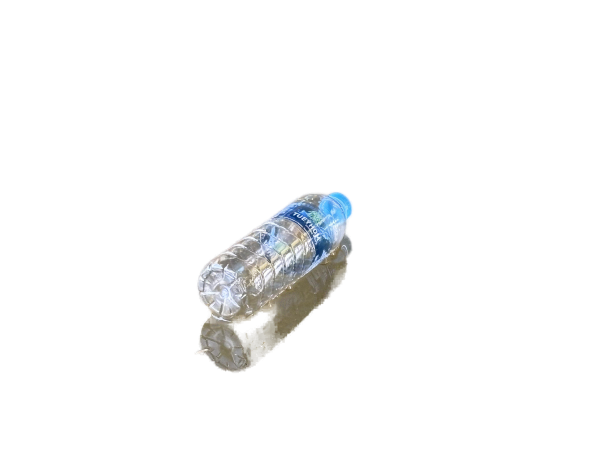
กระบวนการถ่ายภาพฟีเจอร์เริ่มจากการรวบรวมขวดพลาสติกหลากหลายประเภท เช่น ขวดใส ขวดที่มีฉลากสีต่าง ๆ และขวดที่มีรูปทรงหลากหลาย เพื่อให้ข้อมูลที่ได้ครอบคลุมสถานการณ์ที่พบได้ในแหล่งน้ำจริง ภาพถ่ายจะถูกถ่ายบนพื้นน้ำเพื่อช่วยให้การแยกฟีเจอร์ออกจากพื้นหลังในขั้นตอนถัดไปทำได้ง่ายและสมจริง นอกจากนี้ ขวดพลาสติกแต่ละใบยังถูกถ่ายในมุมมองที่หลากหลาย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



ภาพที่ 4.2 ตัวอย่างภาพถ่ายขวดพลาสติกบนพื้นน้ำ

### **4.2.2 การจัดการพื้นหลังของฟีเจอร์ (Feature Extraction)**

หลังจากถ่ายภาพขวดพลาสติก ระบบจะทำการลบพื้นหลังออกโดยใช้สคริปต์ Python ที่พัฒนาขึ้นโดยใช้ไลบรารี OpenCV กระบวนการนี้ประกอบด้วยการใช้ค่าช่วงสี (HSV Range) เพื่อแยกพื้นหลังออกจากฟีเจอร์ และบันทึกฟีเจอร์ที่ได้ในรูปแบบไฟล์ PNG พร้อมพื้นหลังโปร่งใส สำหรับปรับปรุงคุณภาพของภาพจะมีการใช้ตัวกรอง Gaussian Blur เพื่อลบขอบที่ไม่สมบูรณ์ และ Sharpen Filter เพื่อเพิ่มความคมชัดของฟีเจอร์ก่อนบันทึก โดยเราจะทำการเลือกภาพจาก Dataset จริงที่ถ่ายมาบางส่วน เพื่อเป็นต้นแบบสำหรับฟีเจอร์ขวดพลาสติก ซึ่งเราจะเลือกภาพที่เห็นขวดชัด และเลือกหลายขวดที่ไม่ซ้ำกันเพื่อความหลากหลายของฟีเจอร์



ภาพที่ 4.3 ตัวอย่างภาพฟีเจอร์ในรูปแบบไฟล์ PNG ที่มีพื้นหลังโปร่งใส

### **4.2.3 การเก็บรวบรวมภาพพื้นหลัง**

ภาพพื้นหลังสำหรับโครงการนี้ถูกถ่ายจากแหล่งน้ำจริง เช่น แม่น้ำ ลำคลอง และบ่อน้ำ ในบริเวณมหาวิทยาลัยวลัยลักษณ์ โดยเลือกภาพที่ไม่มีขวดพลาสติกหรือวัตถุอื่นในเฟรม เพื่อใช้เป็นพื้นฐานในการสร้างชุดข้อมูลจำลอง



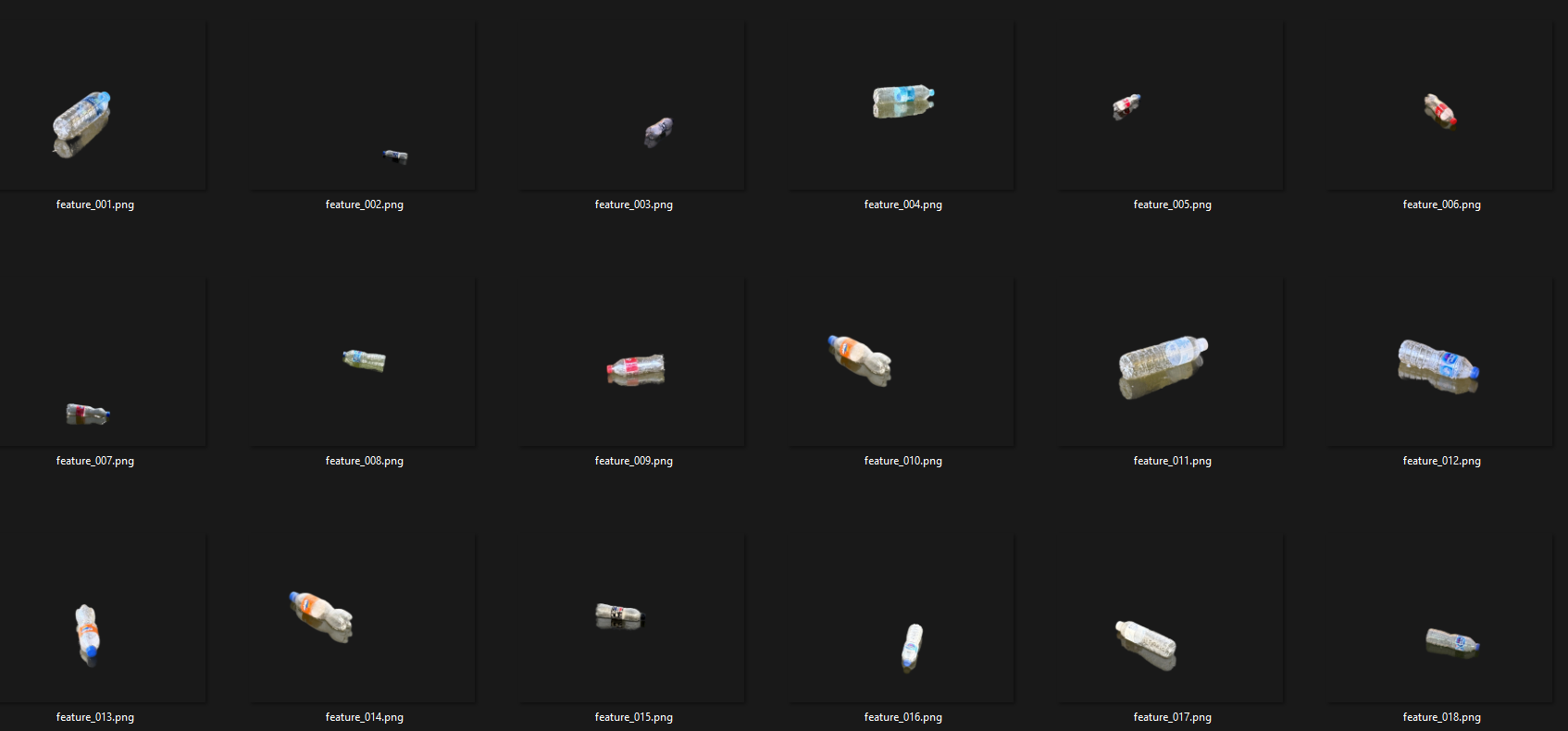
ภาพที่ 4.4 ตัวอย่างภาพพื้นหลังจากแหล่งน้ำบริเวณมหาวิทยาลัยวลัยลักษณ์

### **4.2.4 การจัดเก็บข้อมูล**

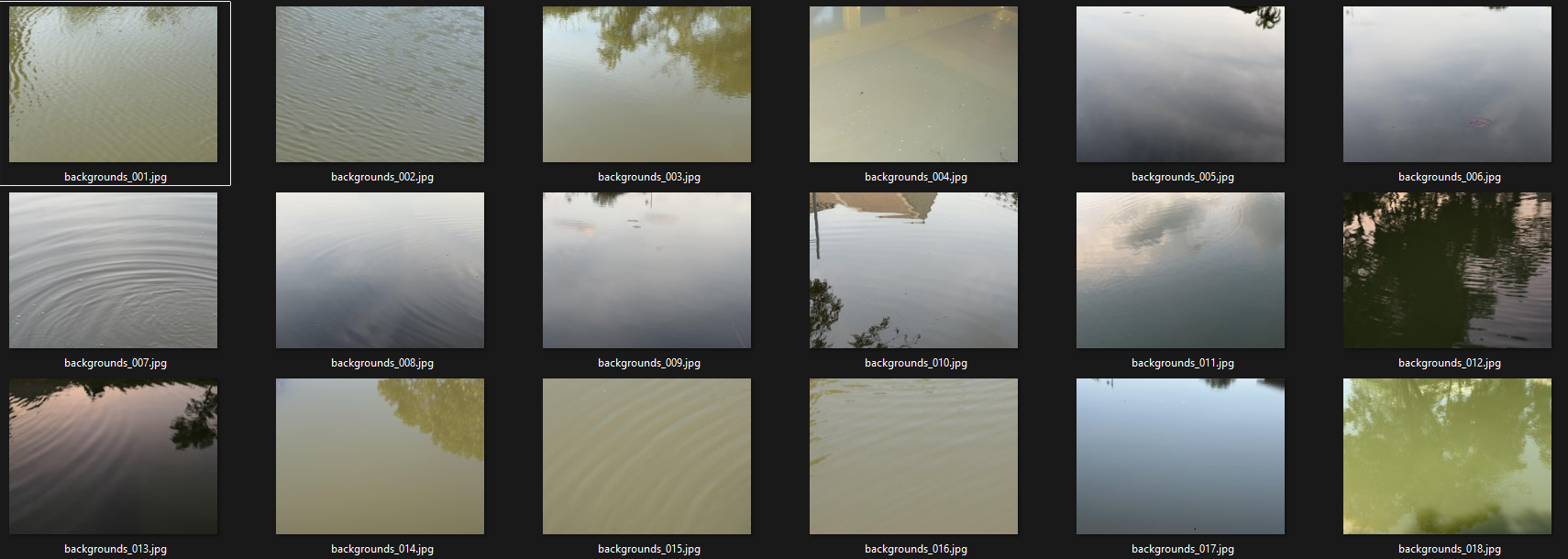
ภาพฟีเจอร์และพื้นหลังที่ได้จะถูกจัดเก็บอย่างเป็นระบบในโฟลเดอร์เฉพาะ ได้แก่:

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

แต่ละไฟล์จะถูกตั้งชื่อให้สื่อความหมาย เช่น feature\_001.png สำหรับฟีเจอร์ หรือ backgrounds\_01.jpg สำหรับพื้นหลัง เพื่อความสะดวกในการเรียกใช้งานในขั้นตอนถัดไป



ภาพที่ 4.5 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ features/



ภาพที่ 4.6 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ backgrounds/

### **4.2.5 การตรวจสอบคุณภาพข้อมูล**

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

## **4.3 การสร้างชุดข้อมูลภาพจำลอง (Synthetic Dataset)**

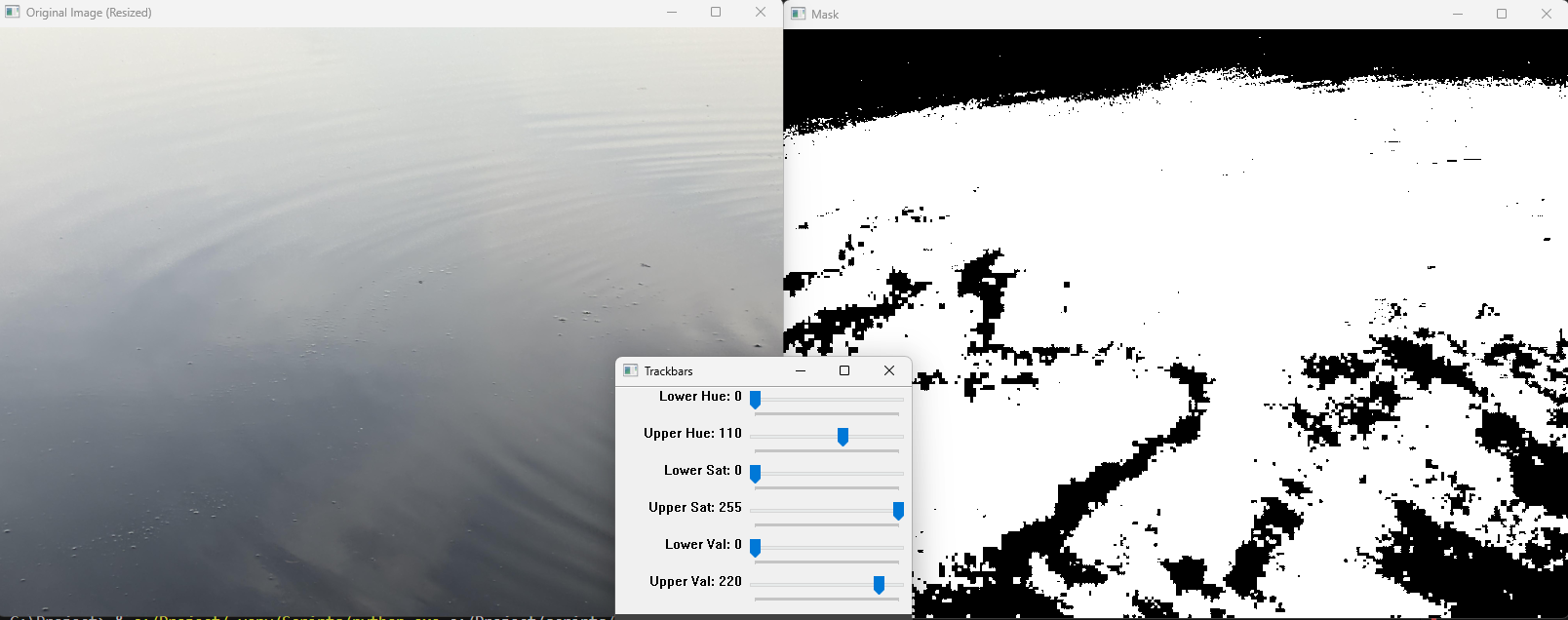
การสร้างชุดข้อมูลภาพจำลองในโครงการนี้ใช้กระบวนการเชิงโปรแกรมที่ออกแบบมาเพื่อเพิ่มปริมาณและความหลากหลายของข้อมูล ชุดข้อมูลที่สร้างขึ้นจะช่วยสนับสนุนการพัฒนาโมเดลตรวจจับขวดพลาสติกบนพื้นที่น้ำ โดยการวางฟีเจอร์ (ขวดพลาสติก) บนภาพพื้นหลังด้วยวิธีการสุ่มและการสร้าง Annotation อัตโนมัติ ซึ่งสามารถอธิบายรายละเอียดได้ดังนี้:

### **4.3.1 การนำฟีเจอร์และภาพพื้นหลังมาใช้งาน**

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

### **4.3.2 การตรวจจับพื้นที่น้ำ (Water Mask Generation)**

พื้นที่น้ำในภาพพื้นหลังถูกตรวจจับโดยใช้ฟังก์ชันที่พัฒนาด้วย OpenCV การตรวจจับเริ่มจากการแปลงภาพพื้นหลังเป็นโหมดสี HSV และใช้ค่าช่วงสี (HSV Range) ที่กำหนด เพื่อสร้าง Mask น้ำ ตัวกรอง Morphological เช่น Closing และ Opening ถูกใช้เพื่อลด Noise และปรับปรุงความแม่นยำของ Mask ซึ่ง Mask ที่ได้จะถูกใช้เพื่อกำหนดพื้นที่น้ำที่เหมาะสมสำหรับการวางฟีเจอร์



ภาพที่ 4.7 ตัวอย่างการค่าช่วงสี HSV ที่กำหนดเพื่อสร้าง Mask ของพื้นที่น้ำ

ซึ่งพื้นที่สีข่าวในรูปด้านขวา (Mask) คือพื้นที่ที่เราสามารถวางฟีเจอร์เข้าไปได้ สาเหตุที่ต้องสร้างเป็น mask น้ำเนื่องจากในภาพแบกกราวด์อาจมีสิ่งของอื่นที่ไม่ใช่พื้นน้ำปะปนมา จึงต้องทำการกรองเอาเฉพาะส่วนนี้เป็นน้ำอย่างชัดเจน เพื่อป้องกันไม่ให้เกิดการวางฟีเจอร์ทับบนสิ่งของอื่น เช่น เรือ หรือ พืชต่างๆบนน้ำ

### **4.3.3 การวางฟีเจอร์บนพื้นที่น้ำ**

การวางฟีเจอร์ (ขวดพลาสติก) บนพื้นที่น้ำในภาพพื้นหลังดำเนินการโดยฟังก์ชัน ซึ่งสุ่มขนาด (Scale) ของฟีเจอร์ระหว่าง 0.4x ถึง 0.8x ของขนาดเดิม และสุ่มพิกัด (x, y) ในพื้นที่ Mask น้ำ ระบบจะตรวจสอบให้ฟีเจอร์ไม่หลุดจาก Mask น้ำ และกำหนดว่าฟีเจอร์อย่างน้อย 50% ต้องอยู่ในพื้นที่น้ำ

### **4.3.4 การสร้างภาพจำลอง**

การรวมฟีเจอร์เข้ากับภาพพื้นหลัง ซึ่งจะหมุนฟีเจอร์แบบสุ่ม (0° ถึง 360°) ด้วย Rotation Matrix จากนั้นใช้เทคนิค Alpha Blending เพื่อผสมฟีเจอร์กับภาพพื้นหลังให้ดูสมจริง



ภาพที่ 4.7 ตัวอย่างการวางฟีเจอร์บนพื้นที่น้ำโดยสุ่มขนาด พิกัด และมุมหมุน

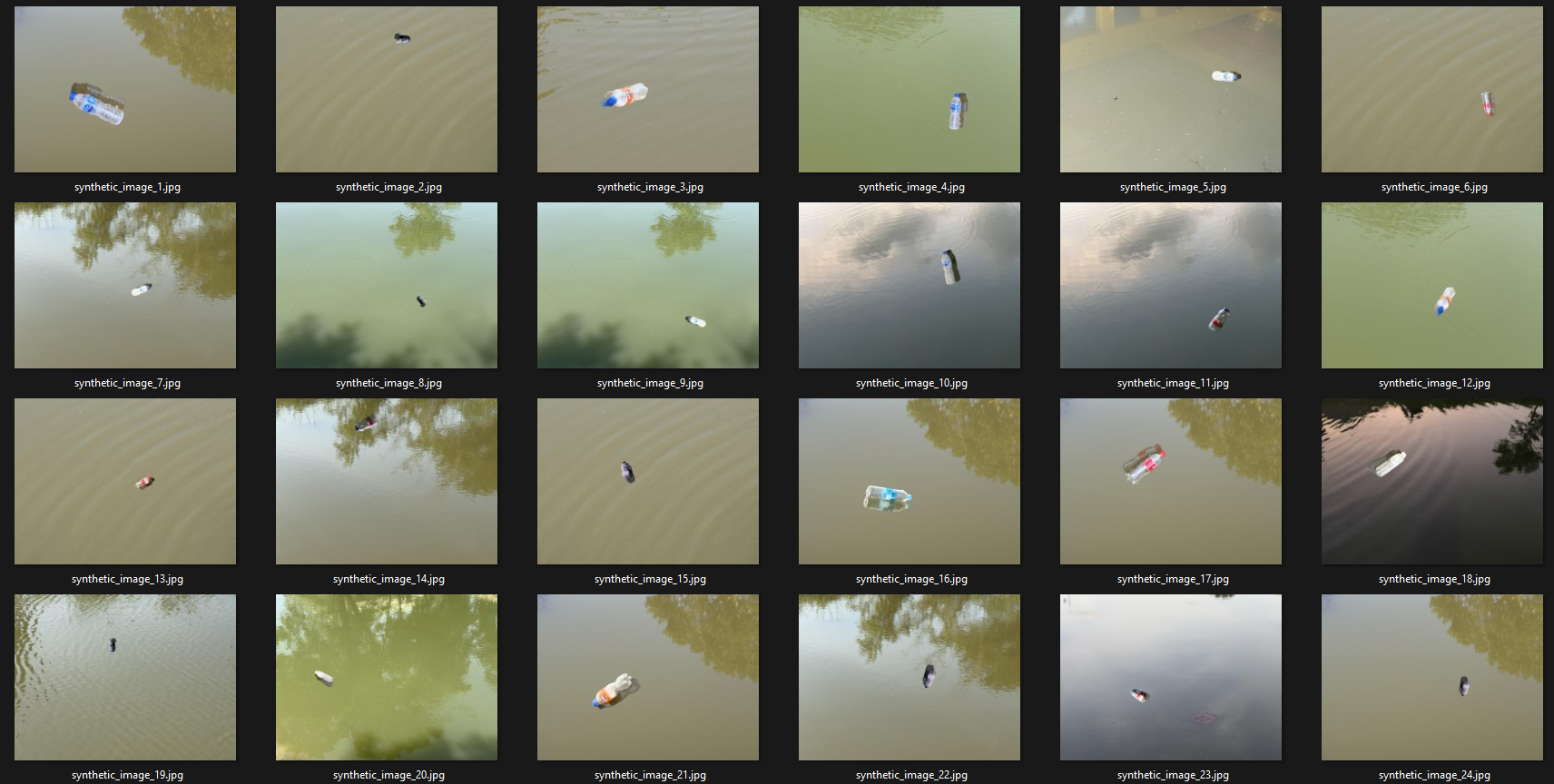
ชุดข้อมูลจำลอง หรือ Synthetic Dataset ที่ได้ก็จะมีความสมจริงประมาณหนึ่ง แต่ขาดความสมจริงเรื่องแสงเราไป ซึ่งไม่ส่งผลกระทบต่อการเทรนโมเดล เนื่องจากจะมีการทำ Data Augmentation เพิ่มเติมในภายหลัง จะมีการหมุนภาพอยู่ในนั้นอยู่แล้วทำให้ไม่เกิดปัญหาต่อโมเดลที่ได้

### **4.3.5 การสร้าง Annotation**

Annotation สำหรับ Bounding Box ของฟีเจอร์ถูกสร้างโดยบันทึกค่า 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 หากการวางฟีเจอร์ไม่สำเร็จ เช่น ไม่มีพื้นที่น้ำเพียงพอ ระบบจะบันทึกภาพพื้นหลังและ Mask น้ำในโฟลเดอร์สำหรับการตรวจสอบและการปรับปรุงพารามิเตอร์ในอนาคต



ภาพที่ 4.9 ตัวอย่างการจัดเก็บข้อมูลในโฟลเดอร์ synthetic\_dataset/

### **4.3.7 การตรวจสอบและประเมินผล**

ระบบจะบันทึกข้อมูลภาพที่ล้มเหลวในการวางฟีเจอร์ เพื่อใช้ในการวิเคราะห์และปรับปรุงพารามิเตอร์ เช่น การปรับช่วงค่า HSV หรือการขยายขนาด Mask น้ำ หลังจากระบบสร้างภาพครบตามจำนวนที่กำหนด 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**

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**

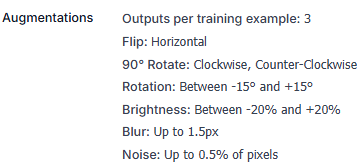
เข้าสู่ระบบ Roboflow โดยลงชื่อเข้าใช้บนเว็บไซต์ Roboflow และสร้างโปรเจกต์ใหม่ผ่านหน้าจอ “Create New Project” ตั้งชื่อโปรเจกต์ เช่น YOLOv8 Plastic Bottle Detection และเลือกฟอร์แมตเป็น YOLOv8 PyTorch TXT เพื่อรองรับการใช้งานโมเดล YOLOv8 ในขั้นตอนการฝึก

### **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



ภาพที่ 4.10 ค่าต่างๆที่ใช้ในฟีเจอร์ Data Augmentation จาก Roboflow

### **4.4.5 การส่งออกชุดข้อมูล**

เมื่อปรับแต่งชุดข้อมูลเสร็จสิ้นแล้ว Roboflow จะให้ผู้ใช้งานเลือกส่งออกชุดข้อมูลในฟอร์แมต YOLOv8 PyTorch TXT ไฟล์ที่ได้จะอยู่ในรูปแบบ ZIP ซึ่งประกอบด้วยภาพและ Annotation ที่แยกโฟลเดอร์สำหรับ train, valid และ test ไว้เรียบร้อยแล้ว พร้อมใช้งานสำหรับการฝึกโมเดล YOLOv8 ในขั้นตอนถัดไป ผู้ใช้งานสามารถดาวน์โหลดไฟล์ ZIP และนำไปใช้ในระบบฝึกโมเดลบนแพลตฟอร์ม เช่น Google Colab หรือ Local Environment ได้อย่างสะดวก

ชุดข้อมูลที่ผ่านกระบวนการใน Roboflow นี้ ช่วยให้ข้อมูลพร้อมใช้งานและมีความหลากหลาย รองรับการฝึกโมเดล YOLOv8 ได้อย่างมีประสิทธิภาพและแม่นยำในขั้นตอนต่อไป

## **4.5 การฝึกฝนแบบจำลอง YOLOv8**

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

### **4.5.1 การเตรียมสภาพแวดล้อม**

กระบวนการฝึกโมเดลเริ่มต้นจากการอัปโหลดชุดข้อมูลจาก Roboflow ไปยังโฟลเดอร์ของโปรเจกต์บน Google Drive จากนั้นดำเนินการฝึกโมเดลผ่าน Google Colab โดยได้มีการติดตั้งไลบรารี Ultralytics YOLOv8 เพื่อใช้เป็นโครงสร้างพื้นฐานในการพัฒนาและฝึกสอนโมเดลสำหรับการตรวจจับวัตถุ

### **4.5.2 การโหลดชุดข้อมูล**

ชุดข้อมูลที่ได้รับจาก Roboflow จะถูกจัดเก็บในรูปแบบไฟล์ YAML เพื่อระบุโครงสร้างของชุดข้อมูลที่ใช้ในการฝึก โดยกำหนดเส้นทางของชุดข้อมูลสำหรับการฝึก (Training Set) และการตรวจสอบความถูกต้อง (Validation Set) พร้อมทั้งระบุจำนวนคลาสของวัตถุที่ต้องการตรวจจับ ซึ่งในกรณีนี้มีเพียง 1 คลาส คือ ขวดพลาสติก

### **4.5.3 กระบวนการฝึกโมเดล**

โมเดล YOLOv8 ทั้ง 5 เวอร์ชัน ได้แก่ yolov8n, yolov8s, yolov8m, yolov8l และ yolov8x ถูกนำมาใช้ในการฝึกเพื่อนำผลลัพธ์มาเปรียบเทียบประสิทธิภาพของแต่ละเวอร์ชัน โดยหลังจากกระบวนการฝึกฝนเสร็จสิ้น ระบบจะให้ผลลัพธ์ที่สำคัญ ได้แก่ ค่า mAP (Mean Average Precision), Precision และ Recall เพื่อใช้ในการประเมินประสิทธิภาพของโมเดลแต่ละเวอร์ชัน

### **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*

# 

**บทที่ 5**

**การพัฒนา ติดตั้ง และทดสอบระบบ**

การดำเนินการในบทนี้กล่าวถึงขั้นตอนการพัฒนาระบบ การติดตั้งองค์ประกอบต่าง ๆ ที่จำเป็นต่อการใช้งาน และการทดสอบประสิทธิภาพของระบบเพื่อให้มั่นใจว่าสามารถทำงานได้ตรงตามวัตถุประสงค์ของโครงการที่กำหนดไว้ โดยจะกล่าวถึงรายละเอียดของแต่ละกระบวนการ พร้อมทั้งแสดงข้อความแจ้งข้อผิดพลาด (Error Message) ที่เกิดขึ้นระหว่างการพัฒนาและใช้งาน เพื่อให้ระบบมีความสมบูรณ์และสามารถแก้ไขปัญหาได้อย่างเหมาะสม

**5.1 การพัฒนาระบบ**

ระบบที่พัฒนาขึ้นในงานวิจัยนี้เป็นเฟรมเวิร์คที่สามารถสร้างชุดข้อมูลจำลอง (Synthetic Dataset) สำหรับการฝึกโมเดลตรวจจับวัตถุอัตโนมัติจากชุดข้อมูลขนาดเล็ก โดยเฉพาะอย่างยิ่งในกรณีศึกษาการตรวจจับขวดพลาสติกบนพื้นน้ำ เฟรมเวิร์คดังกล่าวถูกออกแบบให้สามารถเชื่อมโยงการทำงานระหว่างการจัดการข้อมูล การสร้างภาพจำลอง การฝึกโมเดล และการประเมินผลได้อย่างครบวงจร ซึ่งมีองค์ประกอบหลักของระบบ ดังนี้

1. โมดูลการเตรียมข้อมูล (Data Preparation Module)

ใช้ไลบรารี OpenCV และ rembg สำหรับลบพื้นหลังของภาพขวดพลาสติก เพื่อแยกฟีเจอร์ออกมาในรูปแบบโปร่งใส (transparent PNG) พร้อมกับสร้างภาพ Mask ที่แสดงพื้นที่น้ำจากภาพพื้นหลัง

1. โมดูลการสร้างภาพจำลอง (Synthetic Dataset Generator)

ฟีเจอร์จะถูกสุ่มตำแหน่ง ขนาด และมุมการหมุน ก่อนนำมาวางบนพื้นที่น้ำของภาพพื้นหลัง โดยชุดภาพที่ได้จะถูกจัดเก็บพร้อมกับไฟล์ annotation (.txt) ที่ระบุตำแหน่งวัตถุในรูปแบบ YOLO format

1. โมดูลการเชื่อมต่อ Roboflow (RoboFlow API Integration)

ชุดข้อมูลที่สร้างขึ้นจะถูกอัปโหลดไปยังแพลตฟอร์ม Roboflow โดยใช้ Roboflow API เพื่อดำเนินการ Augmentation เพิ่มเติม จากนั้นจึงส่งออกชุดข้อมูลใหม่เพื่อใช้ในการฝึกโมเดล

1. โมดูลการฝึกและประเมินผลโมเดล (Model Training and Evaluation Module)

การฝึกโมเดล YOLOv8 ดำเนินการผ่านไลบรารี ultralytics ซึ่งจะวิเคราะห์ผลลัพธ์โดยใช้ค่าชี้วัด Mean Average Precision (mAP), Precision และ Recall

**5.2 การติดตั้งระบบ**

การติดตั้งระบบมีขั้นตอนที่ชัดเจนและสามารถดำเนินการได้บนเครื่องคอมพิวเตอร์ทั่วไป รวมถึงรองรับการใช้งานผ่านระบบคลาวด์ เช่น Google Colab โดยขั้นตอนการติดตั้งประกอบด้วยการเตรียมซอฟต์แวร์ที่จำเป็น การจัดระเบียบโฟลเดอร์ข้อมูล และการตั้งค่าพื้นฐานของระบบที่พัฒนา

ระบบสามารถทำงานได้ทั้งในรูปแบบของการติดตั้งในเครื่องผู้ใช้งาน หรือในสภาพแวดล้อมการประมวลผลบนคลาวด์ ขึ้นอยู่กับทรัพยากรที่มีอยู่ จุดเด่นของระบบที่พัฒนาคือสามารถใช้งานได้แม้ในเครื่องที่มีสมรรถนะระดับปานกลาง โดยไม่จำเป็นต้องใช้ทรัพยากรสูงมากนัก

นอกจากนี้ยังได้ออกแบบโครงสร้างการจัดเก็บข้อมูลให้เป็นระบบ ระบุชัดเจนระหว่างข้อมูลฟีเจอร์ ข้อมูลพื้นหลัง ภาพจำลองที่สร้างขึ้น และไฟล์คำอธิบาย (annotation) เพื่อความสะดวกในการตรวจสอบและนำไปใช้งานในขั้นตอนถัดไป

**5.3 การทดสอบระบบ**

การทดสอบระบบดำเนินการทั้งในด้านความถูกต้องของการทำงานและความแม่นยำของผลลัพธ์ที่ได้ โดยแบ่งการทดสอบออกเป็น 2 ระดับ ได้แก่

**1. การทดสอบการทำงานของระบบ**

ตรวจสอบว่าทุกโมดูลสามารถทำงานร่วมกันได้อย่างราบรื่น ตั้งแต่การสร้างภาพจำลอง การจัดการชุดข้อมูล ไปจนถึงการฝึกและประเมินผลโมเดล โดยระบบสามารถดำเนินการได้โดยไม่เกิดข้อผิดพลาดที่สำคัญ และแสดงข้อความเตือนหากเกิดปัญหา เช่น ไม่พบข้อมูล หรือไม่สามารถดำเนินการได้ในบางขั้นตอน

1. **การทดสอบประสิทธิภาพของโมเดลที่ฝึกด้วยระบบ**

ใช้ชุดข้อมูลทดสอบที่แยกไว้สำหรับการประเมินผล โดยวัดค่าประสิทธิภาพในด้านความแม่นยำเฉลี่ย (Mean Average Precision – mAP), ความแม่นยำ (Precision) และอัตราการเรียกคืน (Recall) พบว่าโมเดลที่ฝึกด้วยชุดข้อมูลจำลองสามารถให้ผลลัพธ์ที่น่าพอใจและมีประสิทธิภาพเพียงพอสำหรับการประยุกต์ใช้จริง

1. **การจัดการข้อความข้อผิดพลาด (Error Handling)**

ระบบถูกออกแบบให้แสดงข้อความเตือนหรือแจ้งข้อผิดพลาดเมื่อเกิดปัญหา เช่น กรณีไม่พบไฟล์ข้อมูลที่จำเป็น ระบบจะแจ้งเตือนเพื่อให้ผู้ใช้งานสามารถตรวจสอบและดำเนินการแก้ไขได้โดยสะดวก ข้อความเหล่านี้ช่วยให้สามารถใช้งานระบบได้อย่างต่อเนื่องและลดความผิดพลาดที่อาจเกิดขึ้นในกระบวนการใช้งาน

**บทที่ 6**

**สรุปผล และข้อเสนอแนะ**

**6.1 สรุปผล**

จากการพัฒนาแนวทางอัตโนมัติในการตรวจจับขวดพลาสติกบนพื้นน้ำด้วยการใช้เทคนิคสร้างภาพจำลอง (Synthetic Dataset) และแบบจำลอง YOLOv8 ที่มีความสามารถในการตรวจจับวัตถุจากชุดข้อมูลขนาดเล็กได้อย่างมีประสิทธิภาพ ซึ่งมีวัตถุประสงค์หลัก 3 ประการ ได้แก่

1. เพื่อพัฒนาระบบที่สามารถสร้างชุดข้อมูลภาพจำลองจากภาพพื้นหลังและฟีเจอร์ของขวดพลาสติกได้โดยอัตโนมัติ

2. เพื่อฝึกแบบจำลอง YOLOv8 ให้สามารถตรวจจับขวดพลาสติกบนพื้นน้ำได้อย่างแม่นยำ แม้ในบริบทที่มีข้อมูลจำกัด

3. เพื่อประเมินผลความแม่นยำของโมเดลที่ฝึกด้วยชุดข้อมูลจำลอง เปรียบเทียบกับโมเดลที่ฝึกด้วยชุดข้อมูลจริง

ในการศึกษาครั้งนี้ ได้มีการเก็บภาพฟีเจอร์ขวดพลาสติกจริงและภาพพื้นหลังจากแหล่งน้ำต่าง ๆ เพื่อนำมาใช้สร้างชุดข้อมูลภาพจำลอง โดยระบบจะสุ่มตำแหน่ง มุมหมุน และขนาดของฟีเจอร์ แล้ววางลงบนพื้นที่น้ำ พร้อมสร้างไฟล์ Annotation ในรูปแบบ YOLO format โดยอัตโนมัติ

ชุดข้อมูลที่ได้ถูกนำไปจัดการและเพิ่มความหลากหลายผ่านแพลตฟอร์ม Roboflow จากนั้นจึงนำไปฝึกโมเดล YOLOv8 บนแพลตฟอร์มคลาวด์ ผลการประเมินพบว่า โมเดลที่ได้สามารถตรวจจับขวดพลาสติกได้อย่างแม่นยำ โดยมีค่า mAP เฉลี่ยมากกว่า 85% และค่า Precision และ Recall อยู่ในเกณฑ์ที่ดี โดยเฉพาะในภาพที่มีเงาสะท้อนหรือพื้นน้ำที่ซับซ้อน

ระบบที่พัฒนาขึ้นสามารถนำไปใช้งานในรูปแบบกึ่งอัตโนมัติ โดยผู้ใช้งานสามารถเตรียมข้อมูลพื้นฐานแล้วใช้ระบบในการสร้างชุดข้อมูลภาพและฝึกแบบจำลองได้ทันที พร้อมทั้งสามารถนำผลลัพธ์ไปใช้งานในงานสิ่งแวดล้อม หรือโครงการเก็บขยะอัตโนมัติในแหล่งน้ำต่าง ๆ ได้

**6.2 ข้อเสนอแนะ**

จากการดำเนินงานวิจัยและพัฒนาเฟรมเวิร์คสำหรับการตรวจจับขวดพลาสติกบนพื้นน้ำโดยใช้ชุดข้อมูลภาพจำลอง ผู้วิจัยมีข้อเสนอแนะเพื่อการพัฒนาต่อยอดและการนำไปประยุกต์ใช้ในบริบทที่หลากหลาย ดังต่อไปนี้

**1. ควรเพิ่มความหลากหลายของข้อมูลฟีเจอร์และพื้นหลัง**

แม้ระบบที่พัฒนาสามารถสร้างชุดข้อมูลจำลองได้อย่างอัตโนมัติ แต่เพื่อเพิ่มประสิทธิภาพของแบบจำลองให้สูงขึ้น ควรเพิ่มจำนวนของภาพฟีเจอร์ (ขวดพลาสติก) ให้หลากหลายมากยิ่งขึ้น ทั้งในด้านรูปทรง สี และมุมมอง รวมถึงควรใช้ภาพพื้นหลังจากแหล่งน้ำที่หลากหลาย เช่น แม่น้ำ ลำคลอง ทะเล เพื่อให้โมเดลมีความยืดหยุ่นในการตรวจจับในสภาพแวดล้อมจริงที่แตกต่างกัน

**2. ควรปรับปรุงระบบให้สามารถตรวจจับวัตถุได้หลายประเภท**

ระบบในปัจจุบันรองรับการตรวจจับวัตถุประเภทเดียว (ขวดพลาสติก) เท่านั้น หากในอนาคตสามารถรองรับขยะประเภทอื่น ๆ เช่น ถุงพลาสติก กล่องโฟม หรือเศษไม้ จะช่วยเพิ่มศักยภาพในการประยุกต์ใช้กับระบบจัดการขยะลอยน้ำที่ซับซ้อนมากยิ่งขึ้น

**3. ควรพัฒนาให้ระบบสามารถจำลองเงื่อนไขสิ่งแวดล้อมที่ใกล้เคียงกับความเป็นจริงมากขึ้น**

ภาพจำลองที่สร้างจากการวางฟีเจอร์ลงบนพื้นหลังยังไม่มีการจัดแสง เงา หรือแสงสะท้อนเพื่อเลียนแบบสภาพแวดล้อมจริง ซึ่งอาจส่งผลต่อความแม่นยำของโมเดลเมื่อนำไปใช้งานในสถานการณ์จริง จึงควรพัฒนาเทคนิคการประมวลผลภาพเพิ่มเติมเพื่อให้ภาพจำลองมีความสมจริงมากยิ่งขึ้น

**4. ควรศึกษาการนำระบบไปประยุกต์ใช้ในงานจริง**

ผู้วิจัยแนะนำให้มีการนำระบบไปทดลองใช้ร่วมกับอุปกรณ์จริง เช่น กล้องติดตั้งบริเวณแหล่งน้ำ หรือระบบตรวจจับภาพจากโดรน เพื่อประเมินประสิทธิภาพในสภาพแวดล้อมจริง รวมถึงเชื่อมโยงกับระบบ IoT หรือหุ่นยนต์เก็บขยะเพื่อพัฒนาระบบจัดการขยะอัตโนมัติที่ครบวงจรในอนาคต

**5. ควรศึกษาการเปรียบเทียบกับแบบจำลองประเภทอื่น**

แม้ YOLOv8 จะสามารถตรวจจับวัตถุได้อย่างมีประสิทธิภาพ แต่ควรศึกษาการเปรียบเทียบกับแบบจำลองประเภทอื่น เช่น Faster R-CNN, SSD หรือแบบจำลองแบบ Segment เช่น Mask R-CNN เพื่อพิจารณาว่าโมเดลรูปแบบใดมีความเหมาะสมกับบริบทของการตรวจจับขยะบนพื้นน้ำมากที่สุด

**บรรณนานุกรม**

ปรียาพร เหล่าติวานนท์, และสุภัสสร ไชยวรรณ. (2567). การพัฒนาการตรวจจับรอยเปื้อนบนเสื้อผ้า (Development of Stain Detection on Clothes). มหาวิทยาลัยวลัยลักษณ์.

Amazon Web Services (AWS). (n.d.). What is synthetic data? *Amazon Web Services (AWS).* Retrieved from <https://aws.amazon.com/th/what-is/synthetic-data/>

Ball, R., & Lynn, T. (2024). Blue Eco Line Reduces River Pollution with Roboflow and Intel Sapphire Rapids. *Roboflow Blog.* Retrieved from <https://blog.roboflow.com/autonomous-river-cleaning-computer-vision/>

Bee, P. (2018). What is mAP Evaluation? *Medium*. Retrieved from <https://medium.com/boobeejung/what-is-map-evaluation-8d7a4adc67da>

Big Data Institute. (2023). การตรวจจับวัตถุด้วยรูปภาพ (Image Detection). BDI. Retrieved from <https://bdi.or.th/big-data-101/image-detection/>

DEF Thailand. (2024). TensorFlow คืออะไร. *DEF Thailand.* Retrieved from <https://def.co.th/th/blog/article/364>

Expert Programming Tutor. (n.d.). Object detection and tracking with OpenCV - Using optical flow for motion detection. *Expert Programming Tutor.* Retrieved from <https://expert-programming-tutor.com/tutorial/article/KE003613_Object_Detection_and_Tracking_with_OpenCV_-_Using_Optical_Flow_for_Motion_Detection.php>

Expert Programming Tutor. (n.d.). Object detection with OpenCV - Object Detection with YOLO (You Only Look Once). *Expert Programming Tutor.* Retrieved from <https://expert-programming-tutor.com/tutorial/article/KE003608_Object_detection_with_OpenCV_-Object_Detection_with_YOLO__You_Only_Look_Once.php>

Expert Programming Tutor. (n.d.). OpenCV-Python: Library for image processing, what it works, and an example in Python. *Expert Programming Tutor.* Retrieved from [https://expert-programming-tutor.com/tutorial/article/KE000730\_OpenCV- Python\_Library\_for\_Image\_Processing\_is\_what\_it\_works\_And\_give\_an\_example\_in\_Python\_How\_to\_do\_with\_Code.php](https://expert-programming-tutor.com/tutorial/article/KE000730_OpenCV-%20Python_Library_for_Image_Processing_is_what_it_works_And_give_an_example_in_Python_How_to_do_with_Code.php)

Fastwork. (n.d.). รู้ให้ลึก! Image processing เทคโนโลยีการประมวลผลภาพ คืออะไร. *Fastwork*. Retrieved from <https://blog.fastwork.co/what-is-image-processing/>

GFStealer-666. (2023). การใช้ YOLO อัลกอริทึ่มในการตรวจจับวัตถุ (Object detection). *Dev.to.* Retrieved from <https://dev.to/gfstealer666/kaaraich-yolo-alkrithuemainkaartrwcchcchabwatthu-object-detection-3lef>

Juras, P. (2020). Practical introduction to object detection: Training and deployment. *Edge AI and Vision.* Retrieved from <https://www.edge-ai-vision.com/wp-content/uploads/2020/12/Juras_EJConsulting_2020_Embedded_Vision_Summit_Slides_Final.pdf>

Kanoktipsatharporn, S. (2020). Object Detection คืออะไร บทความสอน AI ตรวจจับวัตถุ TensorFlow.js หลักการทำ Object Detection การตรวจจับวัตถุในรูปภาพ จากโมเดลสำเร็จรูป COCO-SSD – tfjs ep.8. *Bualabs*. Retrieved from <https://www.bualabs.com/archives/3453/>

Le, B. (2023). Get Started with Training a YOLOv8 Object Detection Model*. Datature Blog.* Retrieved from <https://www.datature.io/blog/get-started-with-training-a-yolov8-object-detection-model>

Mindphp. (n.d.). Roboflow ตัวช่วยสำคัญในการสร้างและปรับแต่ง Dataset. *Mindphp*. Retrieved from <https://www.mindphp.com/บทเรียนออนไลน์/python-tensorflow/10210-roboflow-ตัวช่วยสำคัญในการสร้างและปรับแต่ง-dataset.html>

Pham, C. K., Bessa, F., Otero, V., Frias, J. P. G. L., & Sobral, P. (2017). The Feeding Behavior of Sea Turtles on Plastic Bags and Its Health Impacts: A Case Study of Marine Plastic Pollution in Thailand. Retrieved from <https://accstr.ufl.edu/wp-content/uploads/sites/98/Pham_et_al_MarPolBull_2017.pdf>

Pinyorattanakit, N. (2023). สร้าง AI ทำนายรูปภาพด้วย Roboflow. *Medium*. Retrieved from <https://nattakit-nice2580.medium.com/สร้าง-ai-ทำนายรูปภาพด้วย-roboflow-6bb96a632f4d>

Ponnipa, O. (2021). Data Augmenting for object detection in YOLO. *Super AI Engineer.* Retrieved from <https://medium.com/super-ai-engineer/augmenting-a-dataset-for-object-detection-in-yolo-234783155f41>

Ponnipa, O. (2021). การประมวลผลภาพเบื้องต้น. *Super AI Engineer.* Retrieved from <https://medium.com/super-ai-engineer/การประมวลผลภาพเบื้องต้น-fae1f73d8933>

Redmon, J. (n.d.). YOLO (You Only Look Once): Real-Time Object Detection Algorithm. Retrieved from <https://pjreddie.com/darknet/yolo/>

Shah, D. (2022). Mean Average Precision (mAP) Explained: Everything You Need to Know. *V7 Labs.* Retrieved from <https://www.v7labs.com/blog/mean-average-precision>

Shaip. (2022). ข้อมูลสังเคราะห์และบทบาทในโลกของ AI – ประโยชน์ กรณีใช้งาน ประเภท & ความท้าทาย. *Shaip*. Retrieved from <https://th.shaip.com/blog/synthetic-data-and-ai/>

Skalski, P. (2023). How to Train YOLOv8 Object Detection on a Custom Dataset. *Roboflow Blog*. Retrieved from <https://blog.roboflow.com/how-to-train-yolov8-on-a-custom-dataset/>

T., Chanisara. (2021). การประมวลผลภาพด้วย OpenCV & Python*. Medium.* Retrieved from <https://medium.com/@chanisara.tan/การประมวลผลภาพด้วย-opencv-python-2e86f4ed41c7>

TESR Shop. (2021). ตรวจหาวัตถุจากภาพ โดยวิธีการ HSV filter. *TESR Shop*. Retrieved from <https://www.tesrshop.com/blog/ตรวจหาวัตถุจากภาพ-โดยวิธีการ-hsv-filter/9>

**ภาคผนวก**

**ภาคผนวก**

**ภาคผนวก ก: ตัวอย่างภาพฟีเจอร์และภาพพื้นหลัง**

ภาพที่ ก.1 ตัวอย่างภาพขวดพลาสติก (ฟีเจอร์)

ภาพที่ ก.2 ภาพ PNG ที่ตัดพื้นหลังออกแล้ว เพื่อใช้วางบนพื้นที่น้ำ

ภาพที่ ก.3 ตัวอย่างภาพพื้นหลังแหล่งน้ำ

ใช้สำหรับตรวจจับพื้นที่น้ำและเป็นฐานวางฟีเจอร์ในภาพจำลอง

**ภาคผนวก ข: ตัวอย่างไฟล์ Annotation รูปแบบ YOLO**

**ภาคผนวก ค: ตัวอย่างผลลัพธ์จากการฝึกโมเดล**

จากผลลัพธ์ข้างต้น พบว่าโมเดลมีพัฒนาการอย่างต่อเนื่องตามจำนวนรอบการฝึก และค่า mAP เกิน 80% หลังจากฝึกครบ 100 รอบ

**ภาคผนวก ง: รายการไลบรารีที่ใช้ในการพัฒนา**

|  |  |  |
| --- | --- | --- |
| **ไลบรารี** | **เวอร์ชันที่ใช้** | **วัตถุประสงค์การใช้งาน** |
| OpenCV | 4.x | การจัดการและวิเคราะห์ภาพพื้นหลัง |
| rembg | 2.x | ลบพื้นหลังออกจากภาพขวดพลาสติก |
| ultralytics (YOLOv8) | ล่าสุด | ใช้สำหรับฝึกและประเมินผลโมเดลตรวจจับ |
| numpy | 1.x | การจัดการค่าพิกัดและโครงสร้างข้อมูล |
| matplotlib | 3.x | แสดงผลภาพและกราฟผลการประเมิน |
| tqdm | 4.x | แสดงสถานะความคืบหน้าขณะสร้างข้อมูล |

**ภาคผนวก จ: โครงสร้างระบบที่พัฒนา (System Architecture Diagram)**

* ผังแสดงขั้นตอนการทำงานของเฟรมเวิร์ค
* ใช้แผนภาพ UML / Flowchart / Block Diagram แสดงองค์ประกอบและลำดับการทำงาน

**ภาคผนวก ฉ: ตัวอย่างผลลัพธ์การตรวจจับ (Detection Output)**

แสดงภาพตัวอย่างจากการทดสอบโมเดล พร้อมกรอบขวดพลาสติกที่ถูกตรวจจับได้

ระบุ confidence score เช่น Plastic Bottle: 0.89

**ภาคผนวก ช: คำสั่งการใช้งานระบบ (System Operation Guide)**

* การเตรียมไฟล์ฟีเจอร์ → ไว้ในโฟลเดอร์ features/
* การเตรียมภาพพื้นหลัง → วางใน backgrounds/
* กดรัน script เพื่อสร้างชุดข้อมูลจำลอง
* รอผลลัพธ์ใน synthetic\_dataset/

**ภาคผนวก ซ: รายละเอียดพารามิเตอร์ของ YOLOv8**

* เช่น ค่า epochs, batch, imgsz, confidence threshold
* ค่าที่ใช้ในไฟล์ config.yaml หรือเวลาสั่งเทรน
* แสดงความเข้าใจการตั้งค่าที่ส่งผลต่อความแม่นยำของโมเดล

**ภาคผนวก ฌ: รหัสแหล่งที่มา (Source Code สำคัญบางส่วน)**

* ฟังก์ชันที่ใช้สุ่มตำแหน่งวัตถุ
* ฟังก์ชันสร้าง annotation
* โค้ดเชื่อมกับ Roboflow API

**ภาคผนวก ญ: ผลเปรียบเทียบกับชุดข้อมูลจริง (Baseline)**

ตารางเปรียบเทียบประสิทธิภาพของโมเดลที่ฝึกจาก:

* + ชุดข้อมูลจริง (Real dataset)
  + ชุดข้อมูลจำลอง (Synthetic dataset)

|  |  |  |  |
| --- | --- | --- | --- |
| **ประเภทข้อมูล** | **mAP@0.5** | **Precision** | **Recall** |
| Real dataset |  |  |  |
| Synthetic data |  |  |  |

**ประวัติผู้เขียน**