



SCHOOL OF
**INFORMATION TECHNOLOGY
AND INNOVATION**
BANGKOK UNIVERSITY

ระบบตรวจจับและแยกประเภทขยะด้วย AI (Object Detection)

จัดทำโดย

กลุ่ม Refreshing!

นส. ภัทรวดี อุ่นระโลม 1650706441 Section327F

นส. วรณธร แสงจันทร์ 1650708140 Section327F

นาย จิรานุวัฒน์ ม่วงแสง 1650701376 Section327F

นาย คุณานนต์ หิรัญรัตนพร 1650708777 Section327F

นาย ธนบูลย์ ทองประดา 1650706904 Section327F

เสนอ

อาจารย์นิรชา ชัยวงศ์

หลักสูตรวิทยาศาสตรบัณฑิต รหัสวิชา CS460 Artificial Intelligence

ภาคการศึกษาที่ 2 ปีการศึกษา 2567

ภาควิชาวิทยาการคอมพิวเตอร์มุ่งเน้นวิทยาการข้อมูลและความมั่นคงปลอดภัยไซเบอร์

คณะเทคโนโลยีสารสนเทศและนวัตกรรม มหาวิทยาลัยกรุงเทพ

เอกสารรายงานโปรเจกต์: ระบบตรวจจับและแยกประเภทขยะด้วย AI

1. การอธิบายบทบาทและผลงานที่รับผิดชอบ (Contribution)

a. อธิบายขอบเขตของผลงาน

- การวางแผน (Planning):
 - นส. ภัทรวดี อุ่นระโลม
- การเตรียมข้อมูล (Data Preprocessing):
 - นาย คุณานนต์ หิรัญรัตนพร
- การพัฒนาโมเดล (Model Development):
 - นาย จิรานุวัฒน์ ม่วงแสง
 - ศึกษาการ นำโมเดล YoloV8 มา Detect Custom Dataset ที่ต้องการ
เดิมที YoloV8 ตัวเดิมเก่งในด้าน Detect Obj ต่างๆไม่ว่าจะเป็น
ผู้คนและสิ่งของต่างๆแต่ในงานนี้ ต้องการที่จะให้ YOLOV8 รู้จักกับ ขยะแต่ละประเภท
เลยทำการนำ Dataset จาก Kaggle มาทำ Data Annotation เพื่อให้ เทรนกับ
YOLOV8 ได้ และทำการเทรนและเซฟตัวโมเดลออกมา
- การทดสอบ (Testing):
 - นาย ธนบุญ ทองประดา
 - หลักจากตัวโมเดลที่เทรนมาแล้วก็ทำการทดสอบจากขยะต่างๆ เช่น ขวดน้ำ
กล่องกระดาษ
- การออกแบบการนำเสนอ (Presentation Design):
 - นส. วรันธร แสงจันทร์
 - จัดทำข้อมูลในเอกสาร

2. สิ่งที่เราเรียนรู้จากการทำงานในโปรเจก (Reflection)

a. ความท้าทายหลักที่พบระหว่างการทำงานและการแก้ไข

ความท้าทาย	วิธีการแก้ไข
การเก็บและจัดเตรียมชุดข้อมูลขยะ	ตามหาข้อมูลdataset จากแหล่ง ออนไลน์เช่น kaggle
การกำหนด bounding box และ labeling ที่ถูกต้อง	นำ Dataset ที่ได้จาก Kaggle มาทำ Data Annotation Data Annotation
ปรับแต่งโมเดล YOLOv8 ให้มีประสิทธิภาพและตรงกับความต้องการ	Fine-tune model ด้วย Dataset ที่กลุ่มต้องการและทดลองค่าพารามิเตอร์หลายแบบ และประเมินผ่าน confusion matrix
เรื่องเวลาการทำงาน	วางแผนนัดเพื่อนทำงานล่วงหน้า จัดหน้าที่ตามความเหมาะสมแต่ละงาน

b. เทคนิค AI ที่ใช้

- การตรวจจับวัตถุ (Object Detection) ด้วยโมเดล YOLOv8
- การจำแนกประเภทภาพ (Image Classification) สำหรับแยกประเภทขยะ
- เรียนรู้การใช้ Deep Learning
ในการแก้ปัญหาสิ่งแวดล้อมโดยเฉพาะการแยกประเภทขยะเพื่อรีไซเคิล

c. ประโยชน์ที่ได้รับ

- ได้เรียนรู้วิธีการนำ YOLOv8 มาประยุกต์ใช้ในการตรวจจับวัตถุเฉพาะกลุ่มที่ต้องการ
- ได้เรียนรู้โครงสร้างการเตรียม Dataset สำหรับ YOLOv8 ว่ามีผลต่อการทำงานอย่างมาก
- ได้ฝึกการทำงานร่วมกับผู้อื่นอย่างมีระบบ ทั้งด้านเทคนิคและการจัดการเวลา

3. Feedback: การทำงานร่วมกันในทีม

a. การประสานงานในทีม

ทีมมีการแบ่งหน้าที่อย่างชัดเจนดังนี้

- การเก็บข้อมูล ค้นหาและรวบรวม Dataset ที่เกี่ยวข้องจากแหล่งต่าง ๆ
- การสรุปข้อมูล วิเคราะห์และเตรียมข้อมูลให้พร้อมใช้งาน
- การเก็บข้อมูล ค้นหาและรวบรวม Dataset ที่เกี่ยวข้องจากแหล่งต่าง ๆ
- การพัฒนาโมเดล ปรับแต่งและเทรน YOLOv8
- การทดสอบและนำเสนอ ตรวจสอบผลการทำงานของโมเดลและเตรียมสื่อการนำเสนอ

b. ปัญหาที่พบระหว่างการทำงาน

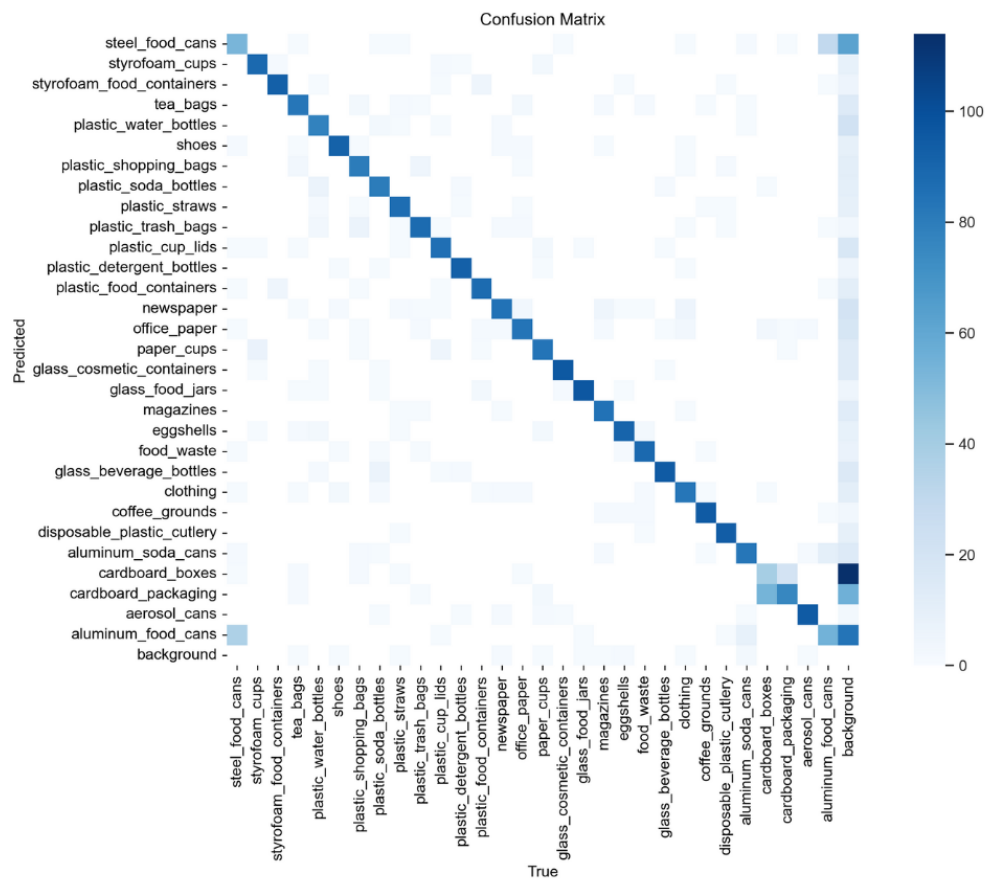
- Dataset ที่ได้จาก Kaggle ไม่ได้จัดเตรียมรูปแบบสำหรับเทรน YOLOV8
จำเป็นต้องแปลงและจัดใหม่
- ขั้นตอนการ Label ตัว Data ทำแบบ auto ผ่านตัว OpenAI
ตัวอื่นจึงทำให้ตัวโมเดลไม่สามารถแยกแยะขยะที่มีหลายประเภทในภาพเดียวกันได้อย่างถูกต้อง

c. ข้อเสนอแนะการพัฒนา

- ทำการ fine tune โมเดลเพิ่มเติม พร้อมปรับค่าพารามิเตอร์เพื่อเพิ่มประสิทธิภาพของโมเดล
- ทำการ Custom Labeling Data ด้วยตนเองโดยเฉพาะกับภาพที่มีวัตถุหลายประเภทเพื่อที่จะให้ตัว YOLO สามารถทำงานได้เต็มประสิทธิภาพ
- พิจารณาการเพิ่มชนิดขยะหรือกลุ่มข้อมูลที่ครอบคลุมมากขึ้น
เพื่อให้ระบบมีความหลากหลายและใช้งานได้จริง

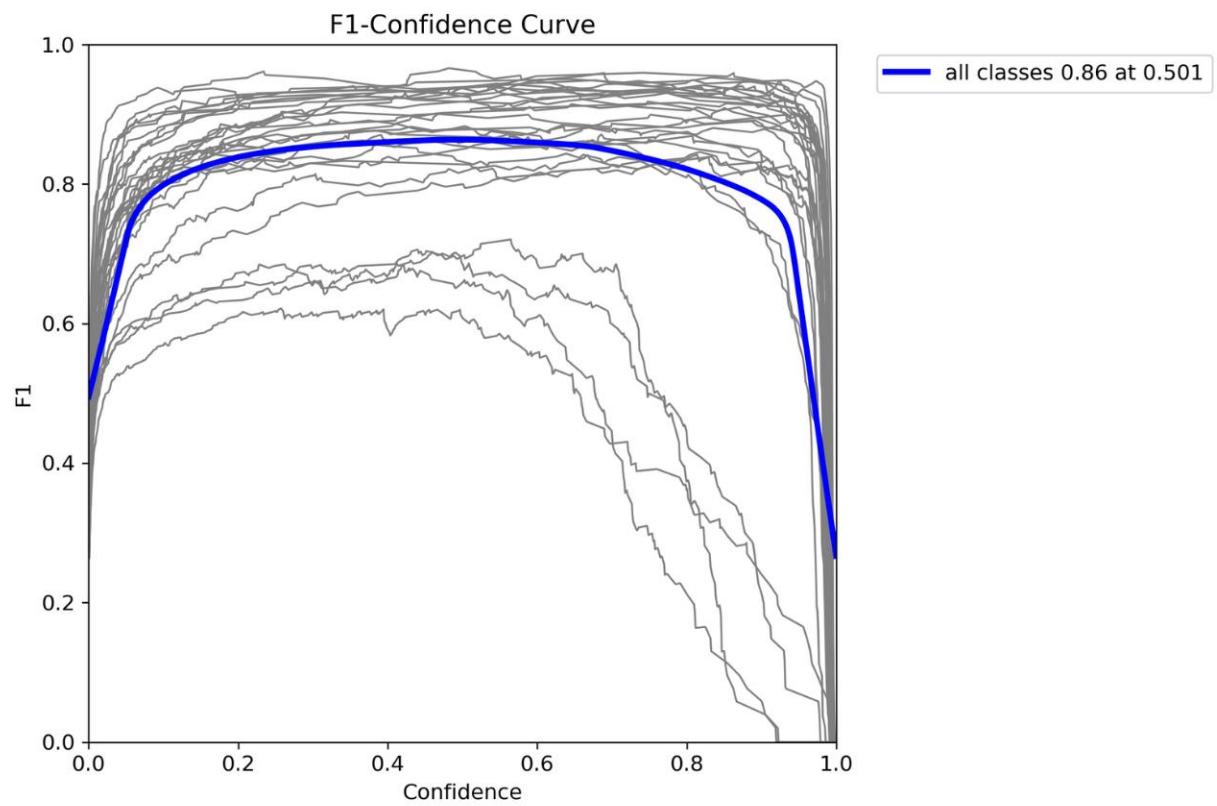
4. Key Result

4.1 Confusion Matrix



Confusion Matrix ใช้ประเมินผลโมเดลจำแนกประเภทขยะรีไซเคิล โดยแสดงผลลัพธ์ระหว่างค่าจริง (True label) ที่แกนแนวนอน กับ ค่าที่โมเดลทำนาย (Predicted) ที่แกนแนวตั้ง และในส่วนของตัวอย่าง class plastic_water_bottles มีสีเข้มในตำแหน่งตรงกับ label จริง แปลว่าโมเดลทำนายแม่นยำ แต่ในส่วนของ aluminum_food_cans มีความสับสนกับ steel_food_cans และ aluminum_soda_cans พอสมควร เพราะมีสีอ่อนกระจายหลายจุดประเภทขยะป้องกันอูมิเนียม

4.2 F1 – Confidence Curve



โมเดลนี้โดยรวมถือว่ามีประสิทธิภาพดี ($F1 = 0.86$) และเหมาะสมจะใช้ threshold ประมาณ 0.5 เพื่อให้ได้สมดุลระหว่าง precision และ recall

4.3 Data preparation

```
# แบ่งข้อมูลแบบสุ่ม (80% train, 20% val)
split_ratio = 0.8

for class_name in os.listdir(base_path):
    class_path = os.path.join(base_path, class_name)
    if not os.path.isdir(class_path):
        continue

    all_images = []
    for subfolder in ['default', 'real_world']:
        img_dir = os.path.join(class_path, subfolder)
        if os.path.exists(img_dir):
            all_images += [os.path.join(img_dir, f) for f in os.listdir(img_dir) if f.endswith('.png')]

    random.shuffle(all_images)
    split_idx = int(len(all_images) * split_ratio)

    class_id = class_map.get(class_name)
    if class_id is None:
        print(f"⚠ ไม่พบ class_id สำหรับ: {class_name}")
        continue

    for idx, img_path in enumerate(all_images):
        target_dir = train_dir if idx < split_idx else val_dir
        img_name = f"{to_snake_case(class_name)}_{idx}.png"
        label_name = img_name.replace('.png', '.txt')

        # โฟลเดอร์สำหรับจัดเก็บรูปภาพและ label
        img_output_path = os.path.join(target_dir, 'images', img_name)
        label_output_path = os.path.join(target_dir, 'labels', label_name)

        # คัดลอกภาพ
        shutil.copy(img_path, img_output_path)

        # สร้าง annotation (คำตัวอย่าง: x_center, y_center, width, height)
        with open(label_output_path, 'w') as f:
            f.write(f"{class_id} 0.5 0.5 1.0 1.0\n") # mock data
```

ภาพนี้เป็นโค้ดสำหรับเตรียมข้อมูลรูปภาพเพื่อใช้ในการเทรนโมเดล โดยมีการแบ่งข้อมูลและสร้าง label text file ในแต่ละภาพ แบ่งข้อมูลเป็น 80% train และ 20% สำหรับ validation

4.4 Class

```
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
val: Scanning C:\Users\thana\Downloads\ML\dataset\images\val\labels.cache... 3000 images, 0 backgrounds, 0 corrupt: 100% | 3000/3000 [00:00<?, ?it/s]
Class      Images  Instances  Box(P  R      mAP50  mAP50-95): 100% | 188/188 [00:13<00:00, 14.07it/s]
all         3000      3000      0.872  0.881  0.921  0.92
steel_food_cans    100       100      0.637  0.788  0.703  0.702
styrofoam_cups     100       100      0.941  0.96   0.975  0.975
styrofoam_food_containers 100       100      0.968  0.92   0.981  0.981
tea_bags          100       100      0.936  0.9    0.947  0.947
plastic_water_bottles 100       100      0.876  0.84   0.923  0.923
shoes             100       100      0.918  0.95   0.979  0.979
plastic_shopping_bags 100       100      0.903  0.84   0.924  0.924
plastic_soda_bottles 100       100      0.944  0.8    0.807  0.895
plastic_straws    100       100      0.918  0.898  0.961  0.961
plastic_trash_bags 100       100      0.907  0.875  0.931  0.931
plastic_cup_lids  100       100      0.889  0.93   0.955  0.955
plastic_detergent_bottles 100       100      0.97   0.954  0.98   0.98
plastic_food_containers 100       100      0.893  0.93   0.961  0.961
newspaper         100       100      0.766  0.85   0.899  0.899
office_paper      100       100      0.758  0.817  0.877  0.877
paper_cups        100       100      0.888  0.86   0.929  0.929
glass_cosmetic_containers 100       100      0.955  0.97   0.992  0.992
glass_food_jars   100       100      0.899  0.96   0.979  0.979
magazines         100       100      0.95   0.85   0.935  0.935
eggshells        100       100      0.947  0.895  0.965  0.965
food_waste        100       100      0.941  0.954  0.979  0.979
glass_beverage_bottles 100       100      0.876  0.96   0.971  0.971
clothing          100       100      0.869  0.87   0.932  0.932
coffee_grounds   100       100      0.92   0.94   0.977  0.977
...
aerosol_cans      100       100      0.912  0.97   0.977  0.977
aluminum_food_cans 100       100      0.636  0.62   0.603  0.693
Speed: 0.1ms preprocess, 0.9ms inference, 0.0ms loss, 0.7ms postprocess per image
Results saved to runs\detect\train22
```

4.5 Result

Creating new Ultralytics Settings v0.0.6 file ✓

View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'

Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see <https://docs.ultralytics.com/quickstart/#ultralytics-setting>


WARNING ⚠ Environment does not support cv2.imshow() or PIL Image.show()

Loading /content/drive/MyDrive/ML/best.torchscript for TorchScript inference...

0: 640x640 1 cardboard_boxes, 1 cardboard_packaging, 7.4ms

Speed: 14.5ms preprocess, 7.4ms inference, 382.8ms postprocess per image at shape (1, 3, 640, 640)

cardboard_boxes 0.46 0.40



<Figure size 640x480 with 0 Axes>

[Google Colab Link](#)

สรุปโครงการ

โครงการนี้มีเป้าหมายเพื่อใช้เทคโนโลยี AI เข้ามาช่วยแก้ไขปัญหาสิ่งแวดล้อม

โดยเฉพาะการจัดการขยะ

ด้วยการพัฒนาระบบที่สามารถตรวจจับและแยกประเภทขยะได้อย่างแม่นยำผ่านโมเดล YOLOv8

ซึ่งช่วยเพิ่มประสิทธิภาพในการรีไซเคิล และสนับสนุนการอนุรักษ์สิ่งแวดล้อมอย่างยั่งยืน

นอกจากนี้ยังเป็นการฝึกทักษะทางเทคนิคและการทำงานเป็นทีม

ซึ่งทำให้ได้ความรู้และเป็นประสบการณ์ในการทำโปรเจกต์ด้าน AI และ Data Science เพิ่มขึ้น