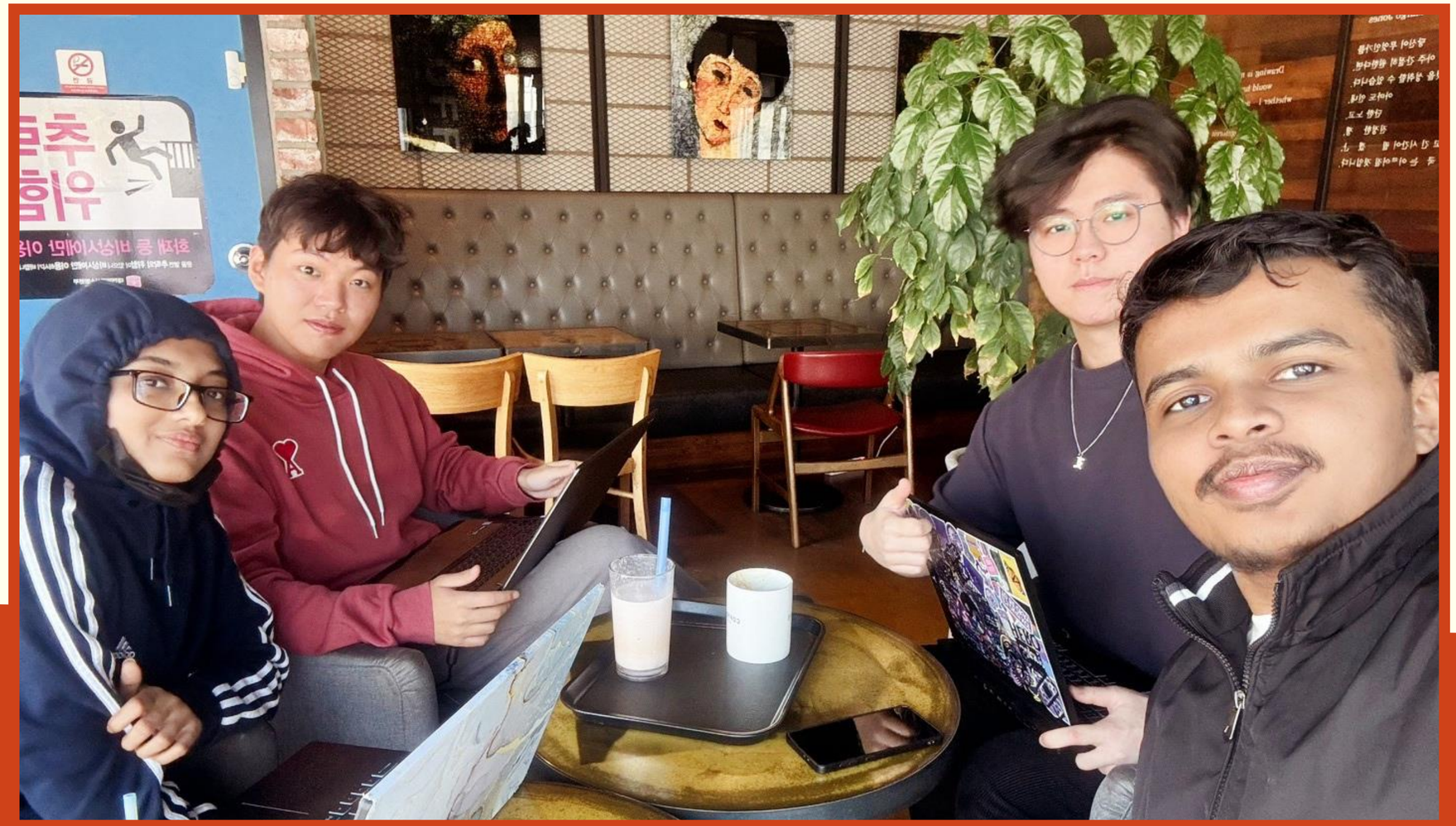
AI-BASED HOUSEHOLD TRASH BAGS CLASSIFICATION SYSTEM

Presentation by EcoSquad

OUR TEAM

WOOSONG UNIVERSITY
우송대학교

Woosong University

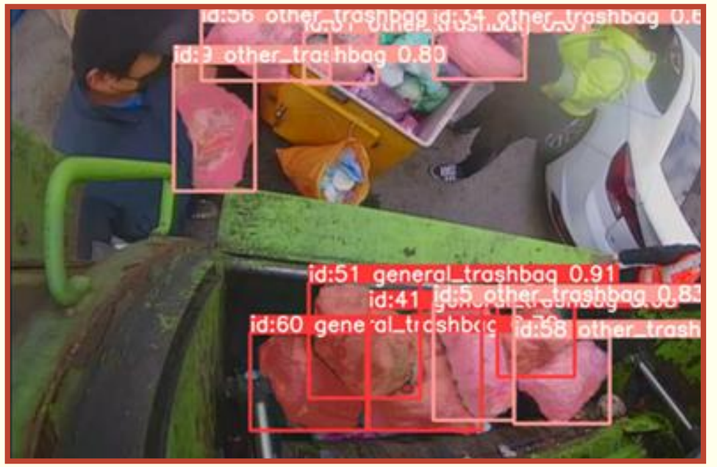


PARTICIPATION COMPANY

대전광역시의 자치구

유성구
儒城區
Yuseong-gu

(주)넷비젼텔레콤



Capstone Competition 2023

Capstone Competition 2024

Development

PROBLEM

Delayed Trash Bags Collection:

- Forgotten trash

Continuous Residential Complaints:

- Smell
- Blocking Path

Lack of Real-Time Monitoring

Development

SOLUTION

AI Based Classification System:

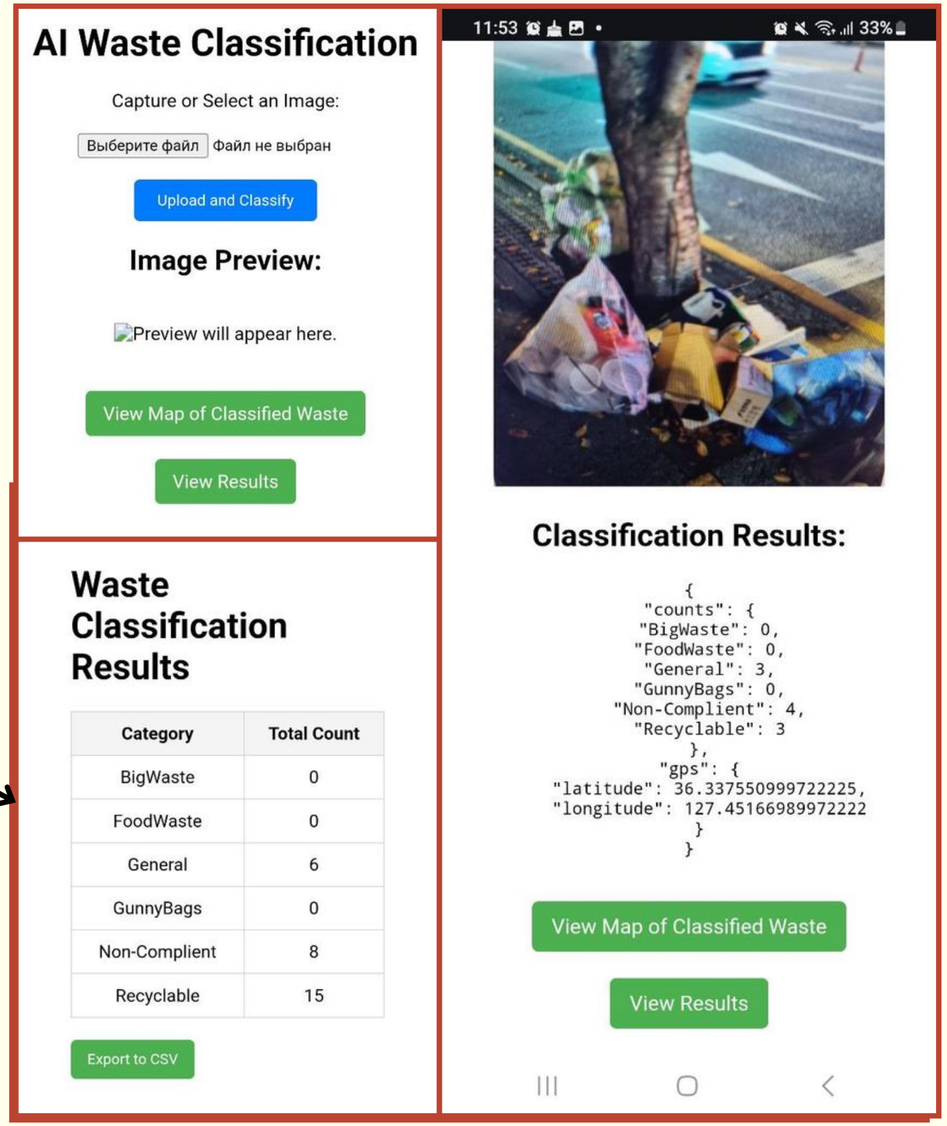
6 waste categories:

- pay-as-you-go bag
- recyclable
- gunny sack
- big waste
- food waste
- non-complient

Data Export to csv

Web & Mobile Application

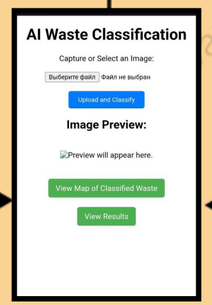
Integrated Map for Real-Time Monitoring



SYSTEM

Waste collector

View



App interface to upload photos and view collection progress

Input

Waste Classification Server for processing, storing and sending data

Input

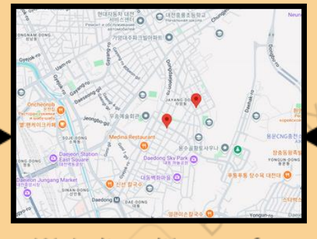
AI

AI Waste Classification Model

Results

DBMS for Storing and Managing classification results

Output



Website with map for trash bag classification information

View

Connect

Take Photo of trash bag

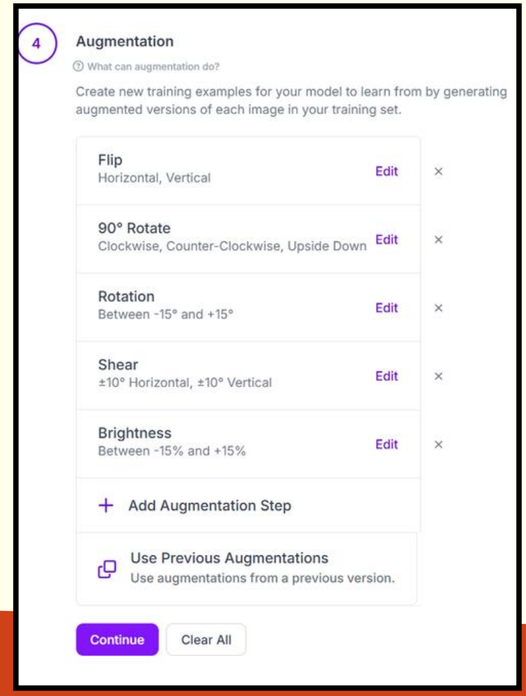
Data for Waste classification (Photos/Videos) and metadata (timestamp data, gps data)

DEVELOPMENT#1

Dataset Collection and Preprocessing

Roboflow

Dataset 8000 images:
Training 92%
Validation 6%
Test 2%

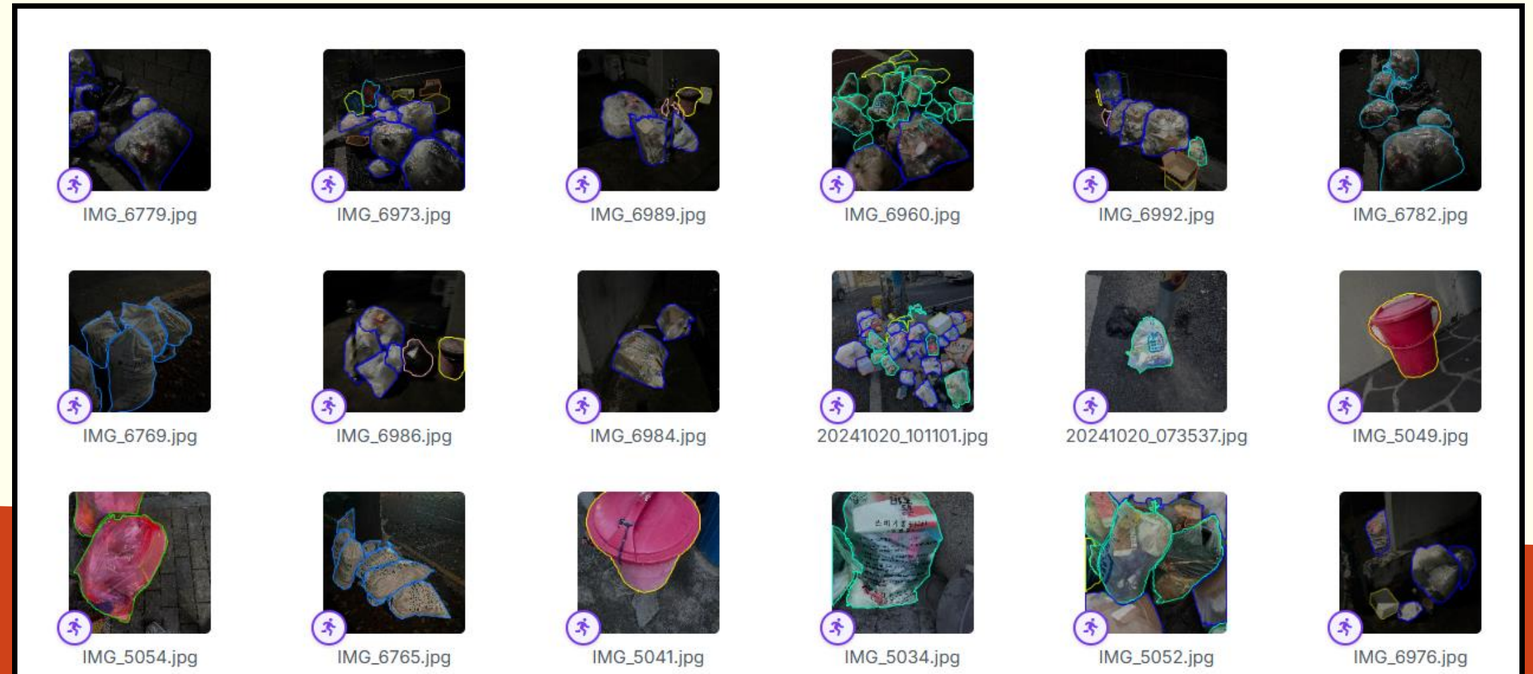


DEVELOPMENT#1

Labeling

Roboflow Labels:

- big-waste
- food-waste
- general-blue
- general-green
- gunny-bag
- gunny-bag-orange
- recyclable-bag-white
- recyclable-cardbox
- recyclable-icebox



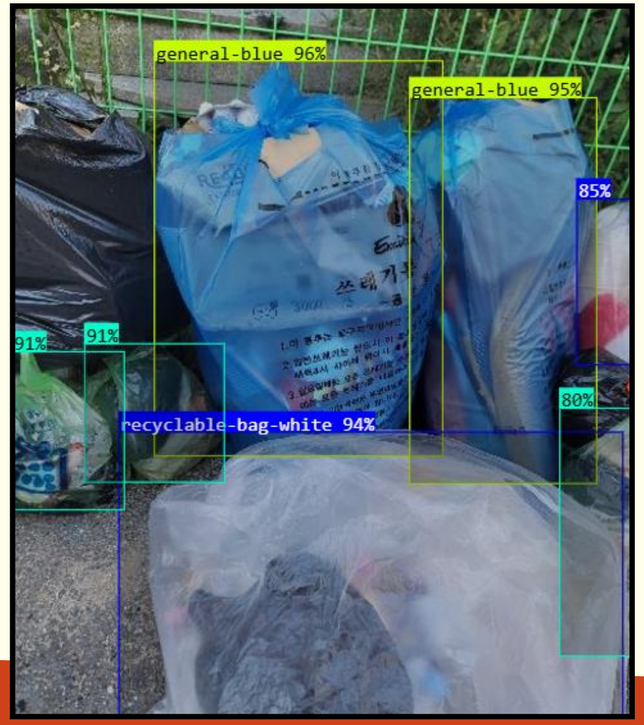
DEVELOPMENT#1

AI model training

Accuracy 85%

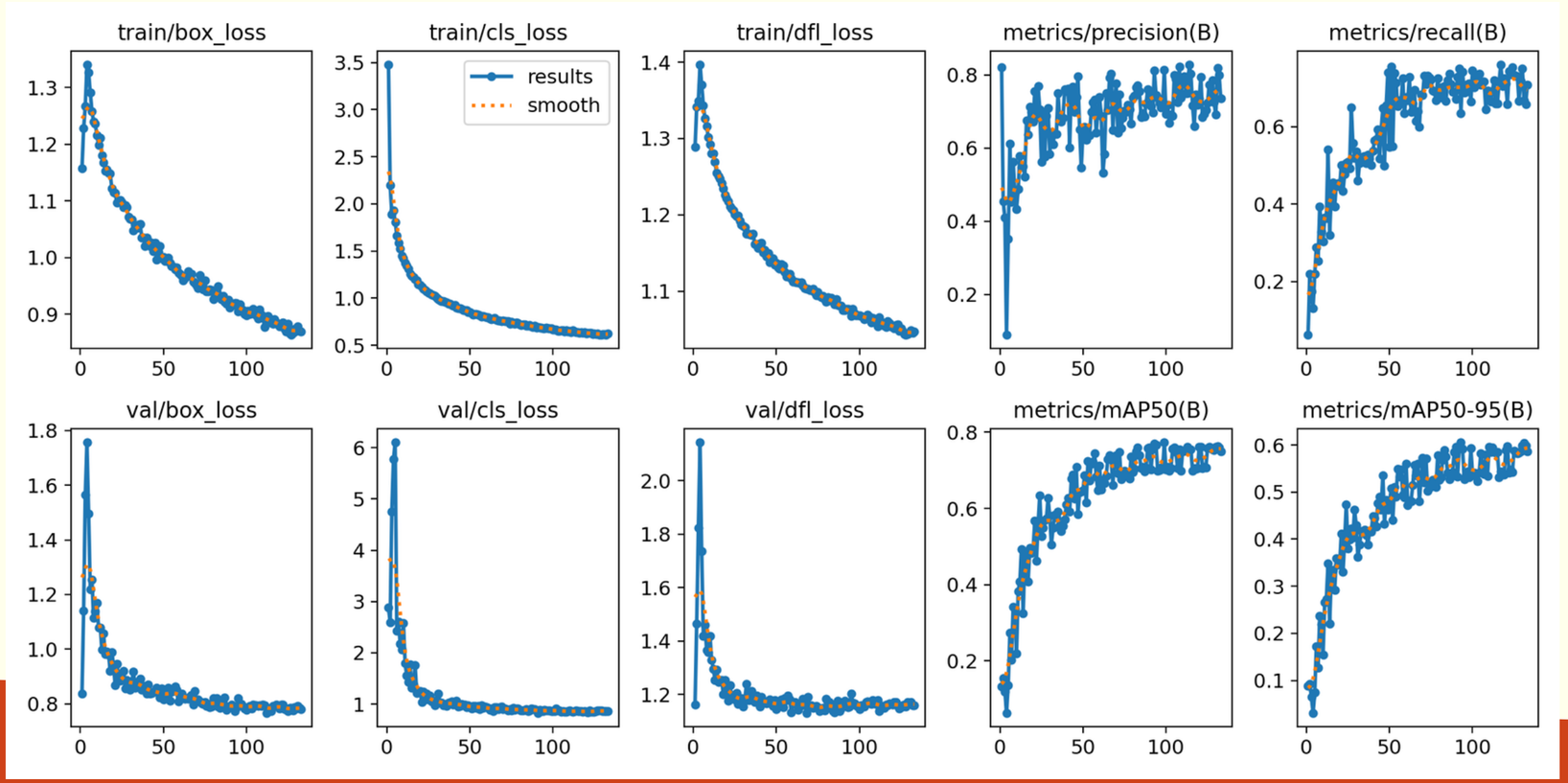
YOLO model:
YOLOv9c
YOLOv5
YOLOv11

IDE for training:
Google Colab
Kaggle



DEVELOPMENT#1

AI model training



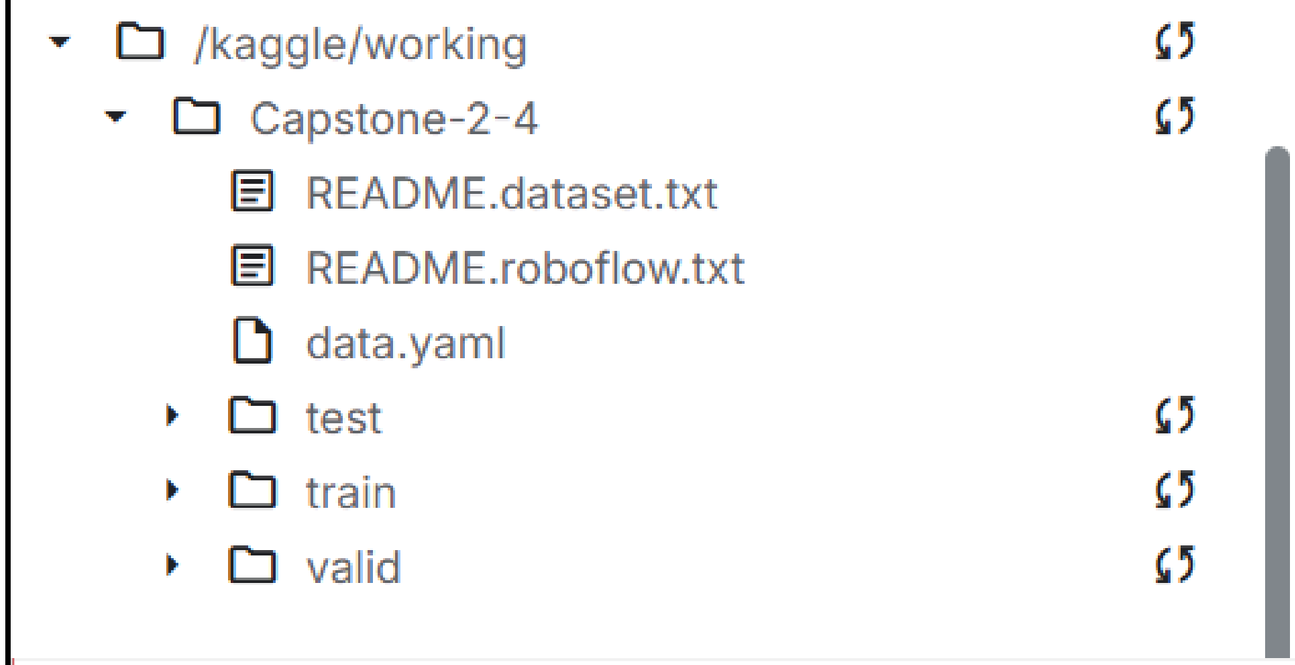
```python
from roboflow import Roboflow
rf = Roboflow(api_key="LzIksZySZh4DbzuxrTeO")
project = rf.workspace("al-y0b6b").project("capstone-2-7aldd")
version = project.version(4)
dataset = version.download("yolov9")
```

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Capstone-2-4 to yolov9:: 100%|████████| 245102/245102 [00:03<00:00, 63201.18
it/s]

Extracting Dataset Version Zip to Capstone-2-4 in yolov9:: 100%|████████| 7618/7618 [00:01<00:00, 5836.36it/s]
```

[3]:

```python
data_yaml_path = '/kaggle/working/Capstone-2-4/data.yaml'

# Step 1: Read the current data.yaml file
with open(data_yaml_path, 'r') as file:
    data_yaml_content = file.readlines()

# Step 2: Modify the content (e.g., change the number of classes or paths)
for i, line in enumerate(data_yaml_content):
    if 'train' in line:
        data_yaml_content[i] = 'train: /kaggle/working/Capstone-2-4/train\n'  # Modify train path
    elif 'val' in line:
        data_yaml_content[i] = 'val: /kaggle/working/Capstone-2-4/valid\n'
        # Modify validation path
    elif 'test' in line:
        data_yaml_content[i] = 'test: /kaggle/working/Capstone-2-4/test\n'
# Step 3: Write the updated content back to the file
with open(data_yaml_path, 'w') as file:
    file.writelines(data_yaml_content)
```

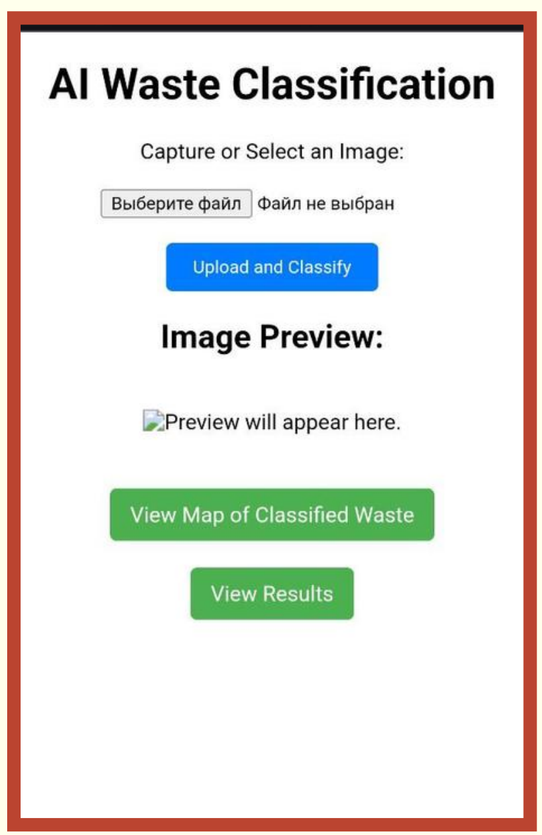
```python
from ultralytics import YOLO

# Load YOLOv9 model
model = YOLO('yolov9c.pt')

# Train the model
results = model.train(
    data='/kaggle/working/Capstone-2-4/data.yaml',
    epochs=150,
    batch=8,
    imgsz=640,
    patience=10,
    name='waste_classifier_yolov9',
    save=True
)
```

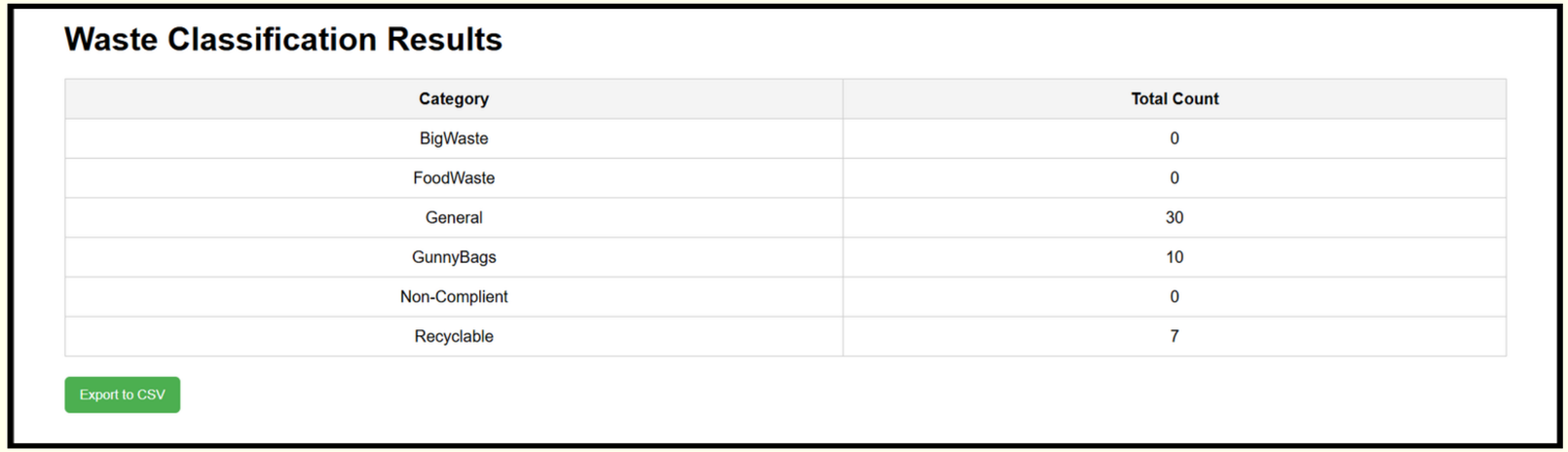



```yaml
names:
- big-waste
- food-waste
- general-blue
- general-green
- gunny-bag
- gunny-bag-orange
- recyclable-bag-white
- recyclable-cardbox
- recyclable-icebox
- non-compliant
nc: 10
roboflow:
  license: CC BY 4.0
  project: capstone-2-7aldd
  url: https://universe.roboflow.com/al-y0b6b/capstone-2-7aldd/dataset/4
  version: 4
  workspace: al-y0b6b
test: /kaggle/working/Capstone-2-4/test
train: /kaggle/working/Capstone-2-4/train
val: /kaggle/working/Capstone-2-4/valid
```

DEVELOPMENT#2

Web Application



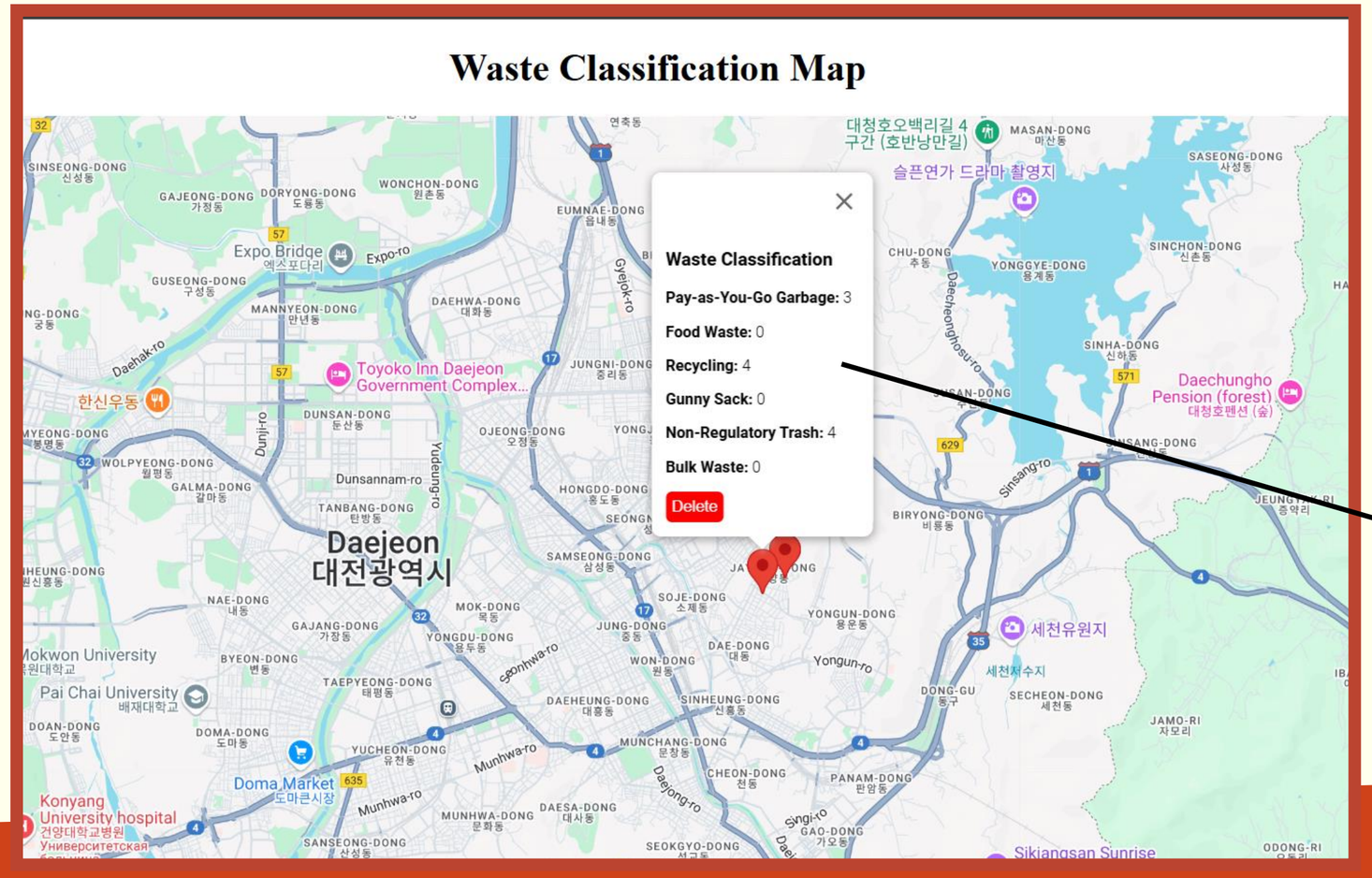


Backend:
Flask Framework
API Request Handling
Metadata Extraction
Database Interaction

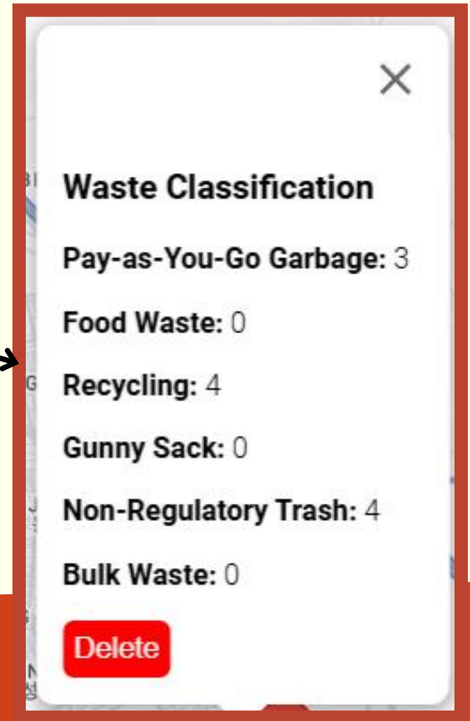
Fronend:
Upload Page
Results Page
Map Page

DEVELOPMENT#2

Google Maps



- Location
- Classification Information



DEVELOPMENT#2

Database(MySql)

waste_classification_db.sql

```sql
CREATE DATABASE ecosquad;

USE ecosquad;

CREATE TABLE WasteCounts (
    id INT AUTO_INCREMENT PRIMARY KEY,
    pay_as_you_go_garbage INT DEFAULT 0,
    food_waste INT DEFAULT 0,
    recycling INT DEFAULT 0,
    gunny_sack INT DEFAULT 0,
    non_regulatory_trash INT DEFAULT 0,
    bulk_waste INT DEFAULT 0
);

CREATE TABLE LocationCoordinates (
    id INT AUTO_INCREMENT PRIMARY KEY,
    latitude DECIMAL(9, 6) NOT NULL,
    longitude DECIMAL(9, 6) NOT NULL
);

select * from WasteCounts;
select * from LocationCoordinates;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conte

id	pay_as_you_go_garbage	food_waste	recycling	gunny_sack	non_regulatory_trash	bulk_waste
4	30	0	1	0	0	0
5	0	0	0	7	0	0
6	0	0	6	3	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

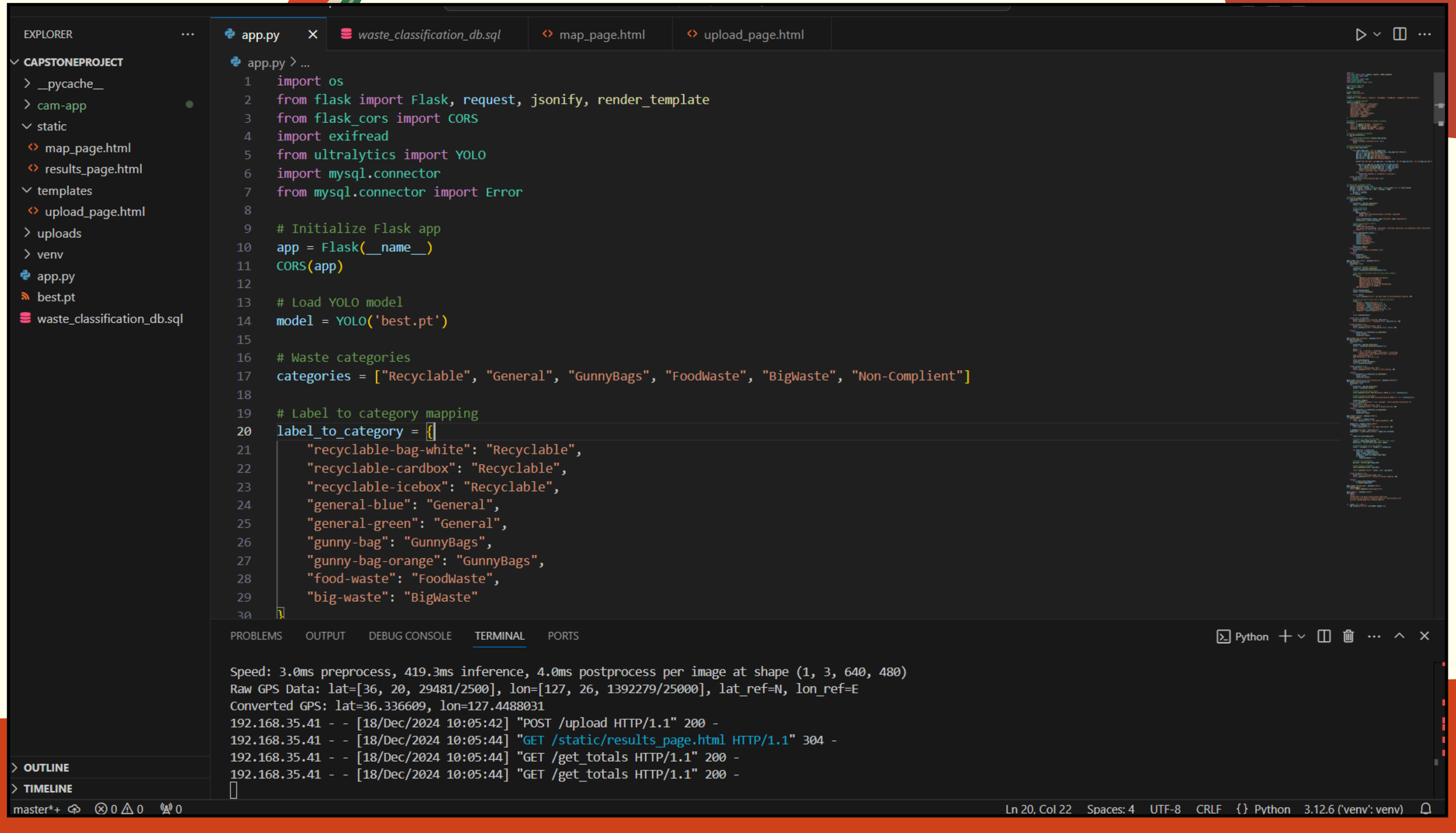
id	latitude	longitude
4	36.334713	127.450823
5	36.335715	127.449064
6	36.336609	127.448803
NULL	NULL	NULL

LocationCoordinates 4

Output

Action Output

#	Time	Action	Message
5	09:46:33	select * from WasteCounts LIMIT 0, 1000	2 row(s) returned
6	09:46:37	select * from LocationCoordinates LIMIT 0, 1000	2 row(s) returned
7	10:10:41	select * from WasteCounts LIMIT 0, 1000	3 row(s) returned
8	10:11:22	select * from LocationCoordinates LIMIT 0, 1000	3 row(s) returned


EXPLORER
app.py
waste_classification_db.sql
map_page.html
upload_page.html
CAPSTONEPROJECT
__pycache__
cam-app
static
map_page.html
results_page.html
templates
upload_page.html
uploads
venv
app.py
best.pt
waste_classification_db.sql
app.py > ...
import os
from flask import Flask, request, jsonify, render_template
from flask_cors import CORS
import exifread
from ultralytics import YOLO
import mysql.connector
from mysql.connector import Error
# Initialize Flask app
app = Flask(__name__)
CORS(app)
# Load YOLO model
model = YOLO('best.pt')
# Waste categories
categories = ["Recyclable", "General", "GunnyBags", "FoodWaste", "BigWaste", "Non-Complient"]
# Label to category mapping
label_to_category = {
"recyclable-bag-white": "Recyclable",
"recyclable-cardbox": "Recyclable",
"recyclable-icebox": "Recyclable",
"general-blue": "General",
"general-green": "General",
"gunny-bag": "GunnyBags",
"gunny-bag-orange": "GunnyBags",
"food-waste": "FoodWaste",
"big-waste": "BigWaste"
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
Speed: 3.0ms preprocess, 419.3ms inference, 4.0ms postprocess per image at shape (1, 3, 640, 480)
Raw GPS Data: lat=[36, 20, 29481/2500], lon=[127, 26, 1392279/25000], lat_ref=N, lon_ref=E
Converted GPS: lat=36.336609, lon=127.4488031
192.168.35.41 - - [18/Dec/2024 10:05:42] "POST /upload HTTP/1.1" 200 -
192.168.35.41 - - [18/Dec/2024 10:05:44] "GET /static/results_page.html HTTP/1.1" 304 -
192.168.35.41 - - [18/Dec/2024 10:05:44] "GET /get_totals HTTP/1.1" 200 -
192.168.35.41 - - [18/Dec/2024 10:05:44] "GET /get_totals HTTP/1.1" 200 -
OUTLINE
TIMELINE
master*+
Ln 20, Col 22
Spaces: 4
UTF-8
CRLF
Python
3.12.6 ('venv': venv)

RESULTS

Fully Functional AI System:

- Fast Trash Bag Classification.
- Accurate Object Detection.

Integrated Web Map.

- Fast Response.
- Comfortable Monitoring.

Web and Mobile Application:

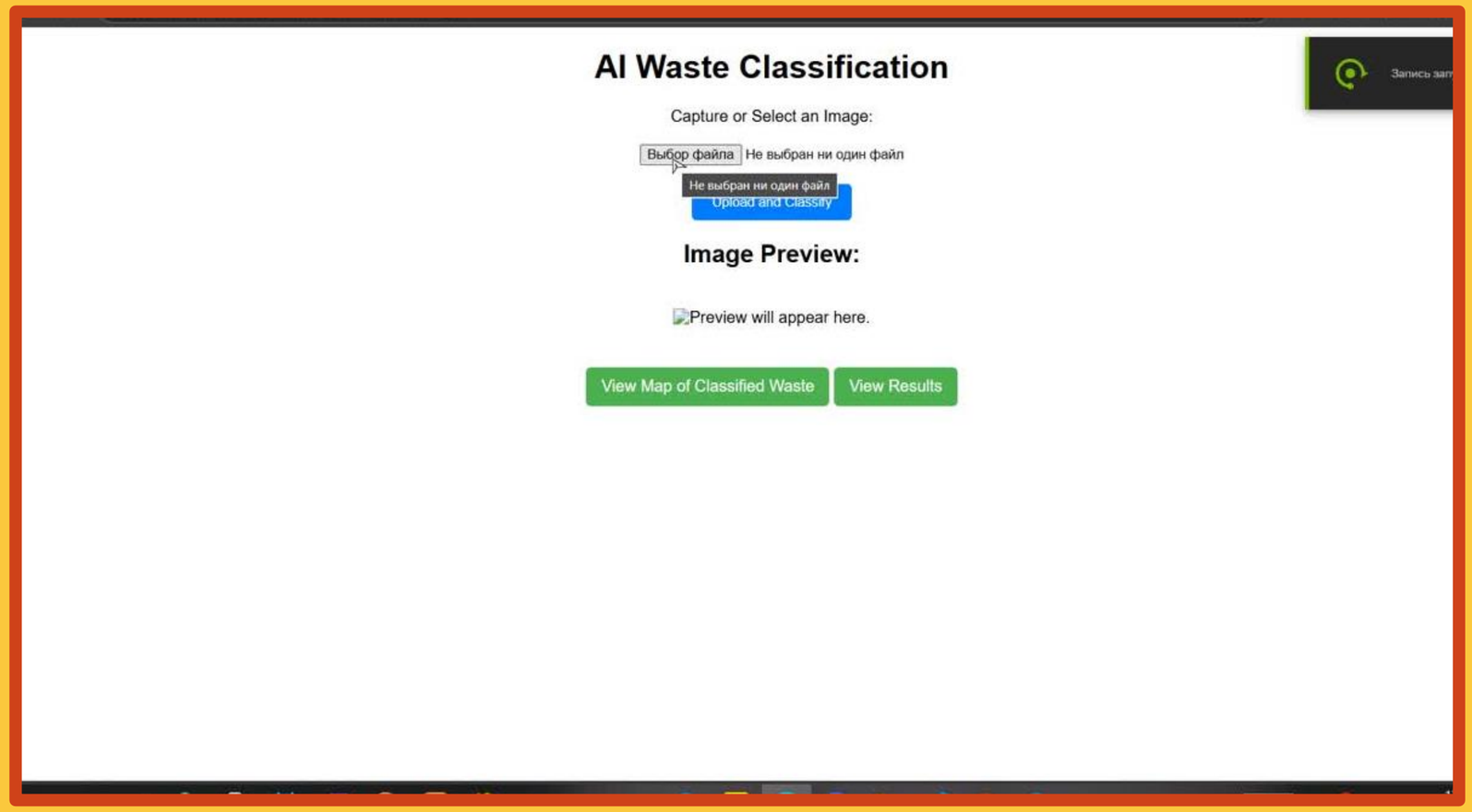
- Efficiency Improvement.
- Data Result Export

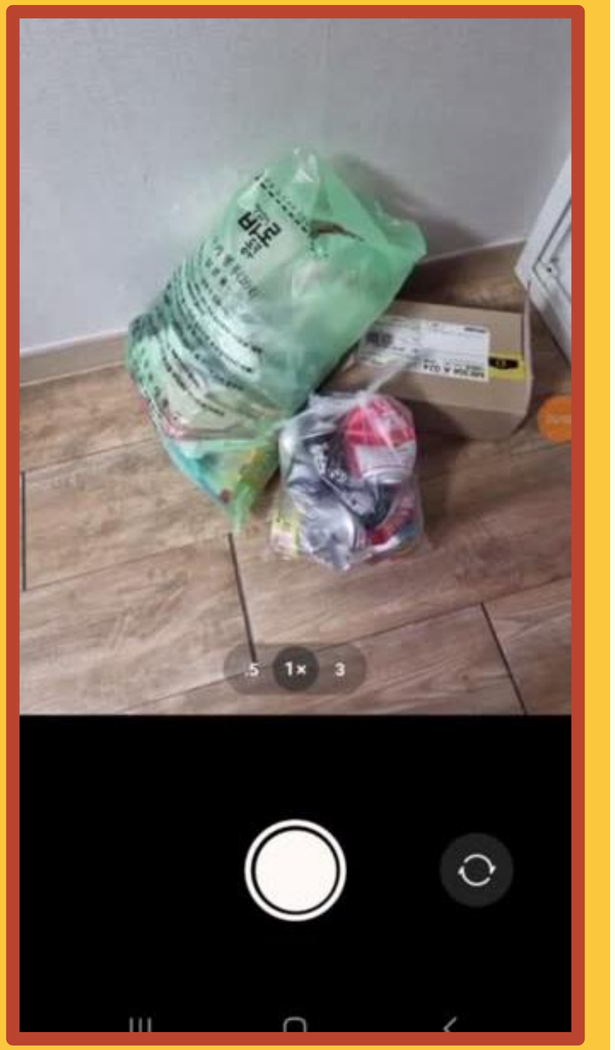
FUTURE PLAN

2025 -2026

- Improve AI Model.
- Expand System Features.(route estimation)
- Real World Testing.

Demo


AI Waste Classification
Capture or Select an Image:
Выбор файла
Не выбран ни один файл
Не выбран ни один файл
Upload and Classify
Image Preview:
Preview will appear here.
View Map of Classified Waste
View Results


.5
1×
3

THANK YOU