AI TRASH DETECTION SYSTEM

우송대학교
WOOSONG UNIVERSITY
Woosong university

Netvision telecom (주)넷비젼텔레콤

Jenn

Class

Capstone Design
in Companys
Project 2

우송대학교
WOOSONG UNIVERSITY

Woosong university

Professor

Edward Kim Young Ill
김영일

Woosong Vision

Cavan Schutte 202112069
Heang Seavleu 202230251
길유진 202110638
Koyiljon Valiev 202112096

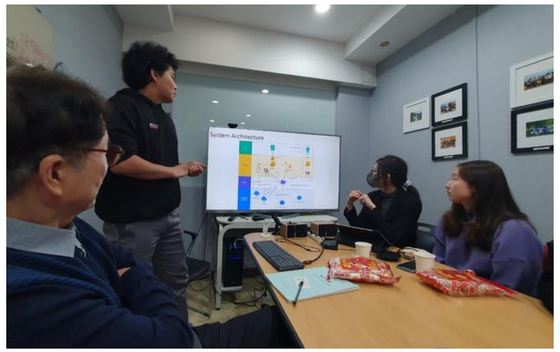
우송대학교
Woosong university

Koyil

Participating Company

(주)넷비젼텔레콤

Company office --->



MOU

Memorandum of Understanding

유성구
YUSEONG DISTRICT

우송대학교
WOOSONG UNIVERSITY

Woosong university

(주)넷비젼텔레콤

대전 유성구,
‘생활폐기물 수거개선 실증’
업무협약 체결

https://daily.hankooki.com/news/articleView.html?idxno=1028480

AGENDA

Introduction	Team	Netvision & MOU
Problem definition	Possible solutions	Collecting Data
AI results	AI process	Conclusion

1st:

NETVISION PROJECT

THE PROJECT

Data driven household waste collection and transportation business innovation

데이터 기반 생활 폐기물 수집 운반 업무 혁신

- Collect more trash everyday
- Reduce complaints from residents

Problem

Large number of complaints from Residents

SMELL

- Large piles of trash gather up - not collected soon enough
- Forgotten trash - smelly bags, or begin to smell

FORGOTTEN TRASH

- Trash not collected

BLOCKING PATH

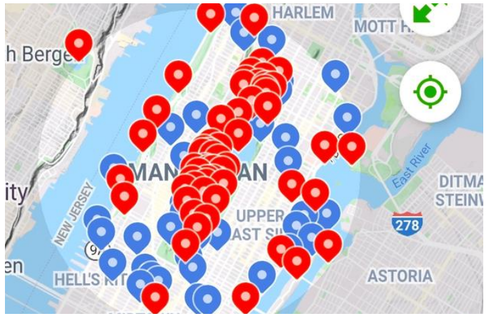
- Large piles of trash block walkways

HOW TO SOLVE TRASH PROBLEM?

1

Identify garbage usage patterns

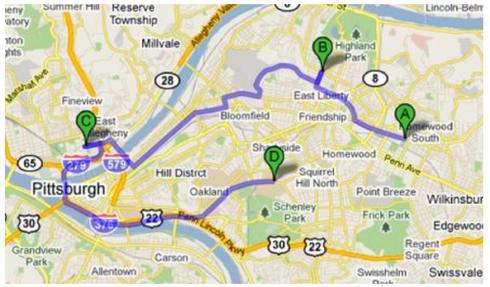
- Know areas of high and low trash usage



2

Optimal driving paths

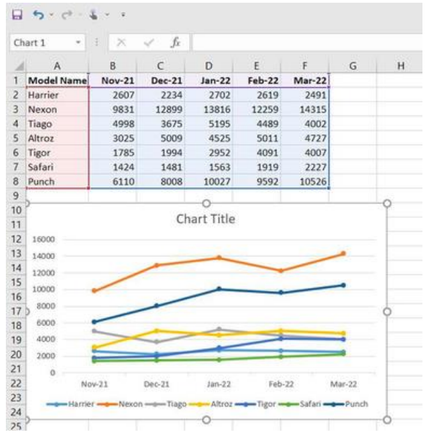
- Effective route for most collection per day



3

Accumulate data and create operational create plan

- Effective operations for greater trash collection



Requires DATA

Requires
High - Low
Usage **DATA**

Requires
High - Low
Usage
And Route **DATA**

Jenn

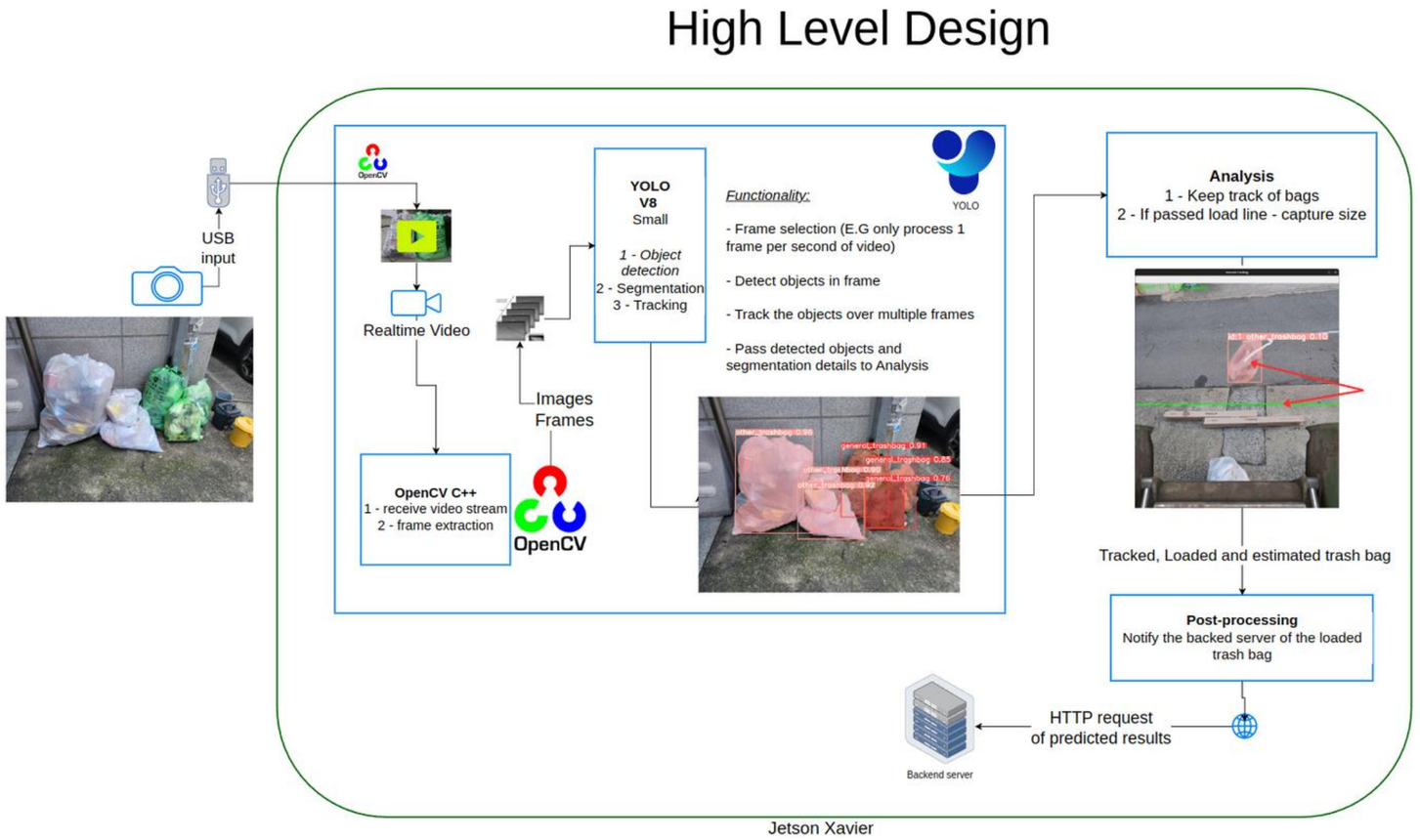
COLLECT DATA

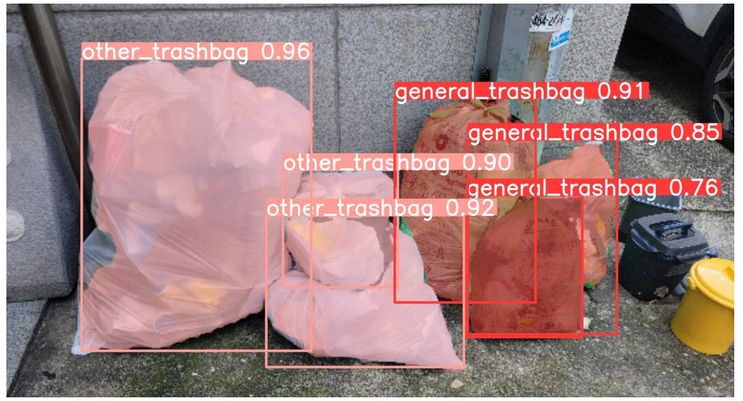
COLLECT DATA

AI powered trash detection

AI

System overview





AI RESULTS

1 Bag detection + segmentation

2 Size estimation

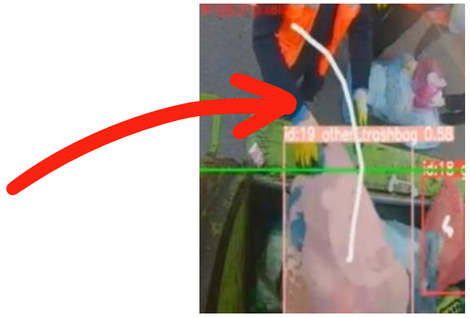
```python
def estimate_bag_size(boxes_with_masks):
    bag_sizes = [
        ('1LT', 900),
        ('2LT', 1500),
        ('5LT', 2500),
        ('10LT', 3500),
        ('25LT', 5000),
        ('50LT', 7000),
        ('75LT', 9000),
        ('100LT', 13000),
    ]

    for box in boxes_with_masks:
        total_pixels = box['mask'].sum()
        box['total_pixels'] = total_pixels

        # Iterate through bag size thresholds
        # Start at the smallest bag size (900), and work up to the largest bag size (13000)
        # If the bag(1200) is bigger than the size(900), then set it to that size
        # Stop when the size(1500) is bigger than the bag(1200)
        for bag_size, min_size in bag_sizes:
            if total_pixels >= min_size:
                box['estimated_size'] = bag_size
                break  # Stop searching once the bag is larger than the next threshold

    return boxes_with_masks
```

- 1 Litre
- 2 Litre
- 5 Litre
- 10 Litre
- 25 Litre
- 50 Litre
- 75 Litre
- 100 Litre

3 Classification

- general trash
- other trash

4 Tracking (frame-by-frame)



5 REST communication with backend server

DEMO

Model: *YOLO V8 Small*

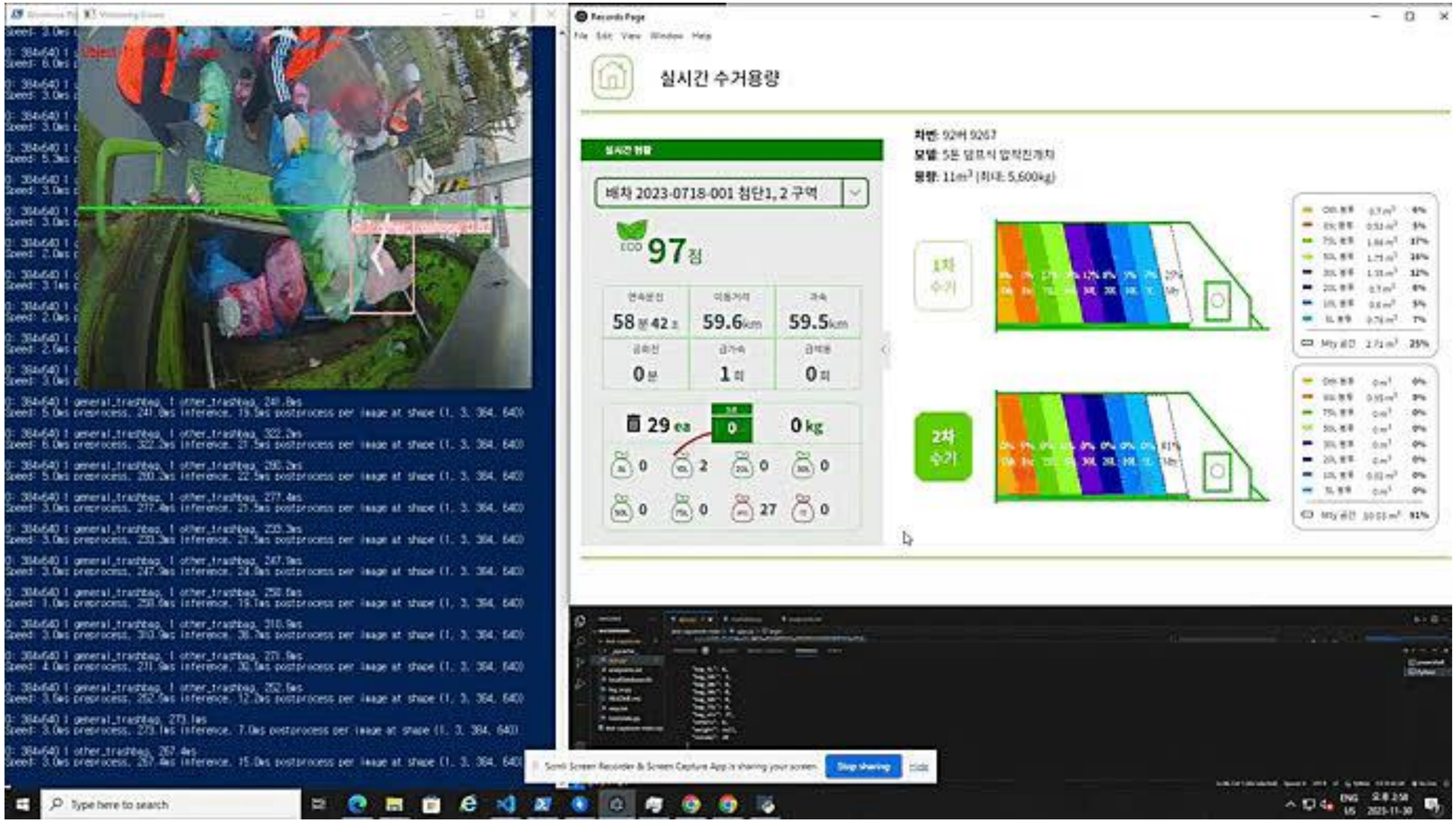
Dataset: *1284 images (3448 aug)*

Training: *1 Hour GPU GTX 1660*

DEMO

WHOLE SYSTEM

inaccurate - (146 images)



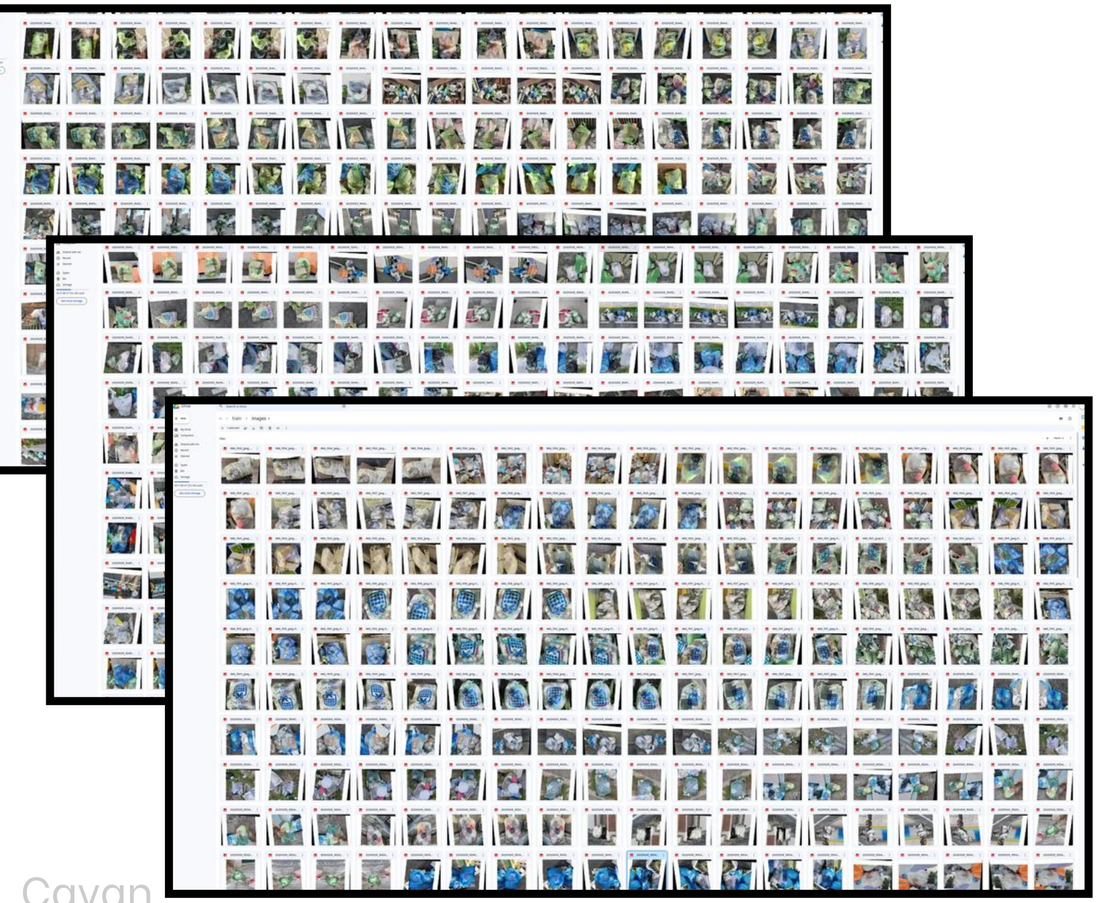
AI DEVELOPMENT PROCESS

DATASETS

V1

Train: 605

Valid: 19



V2

Train: 3448

Valid: 190

Dataset
how we labeled

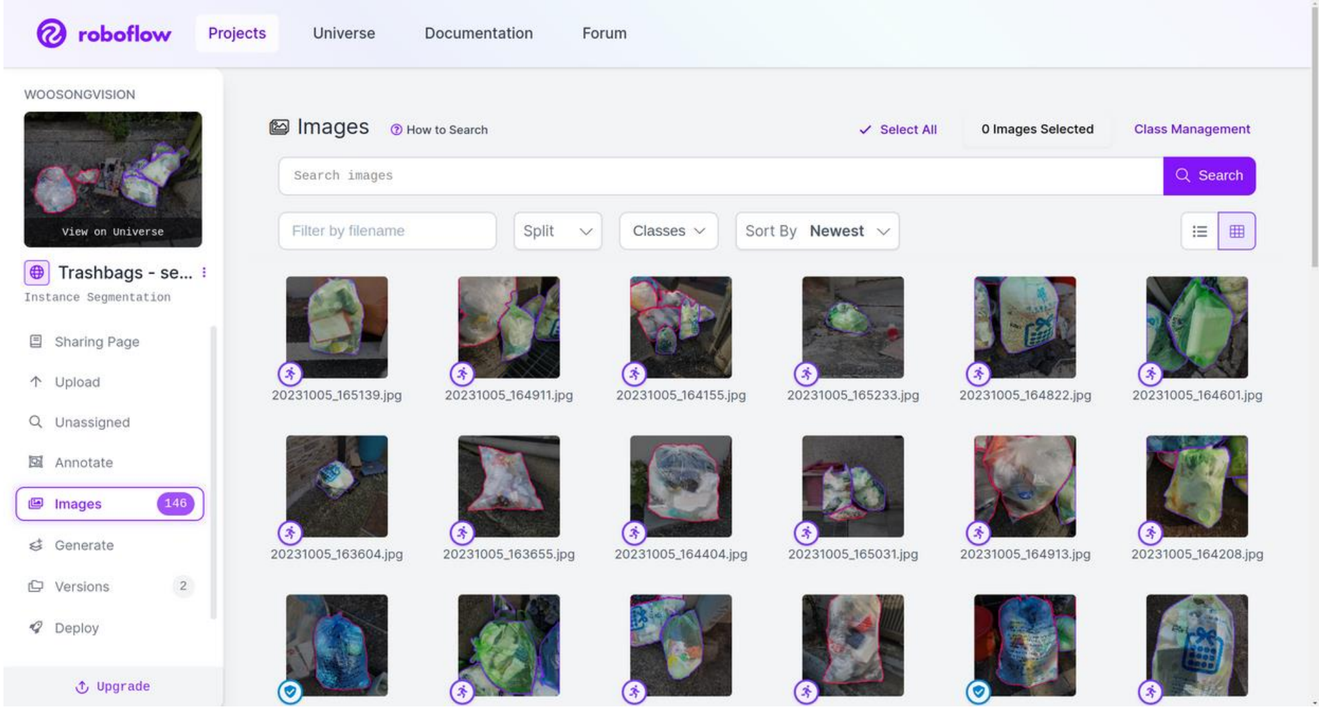
Roboflow

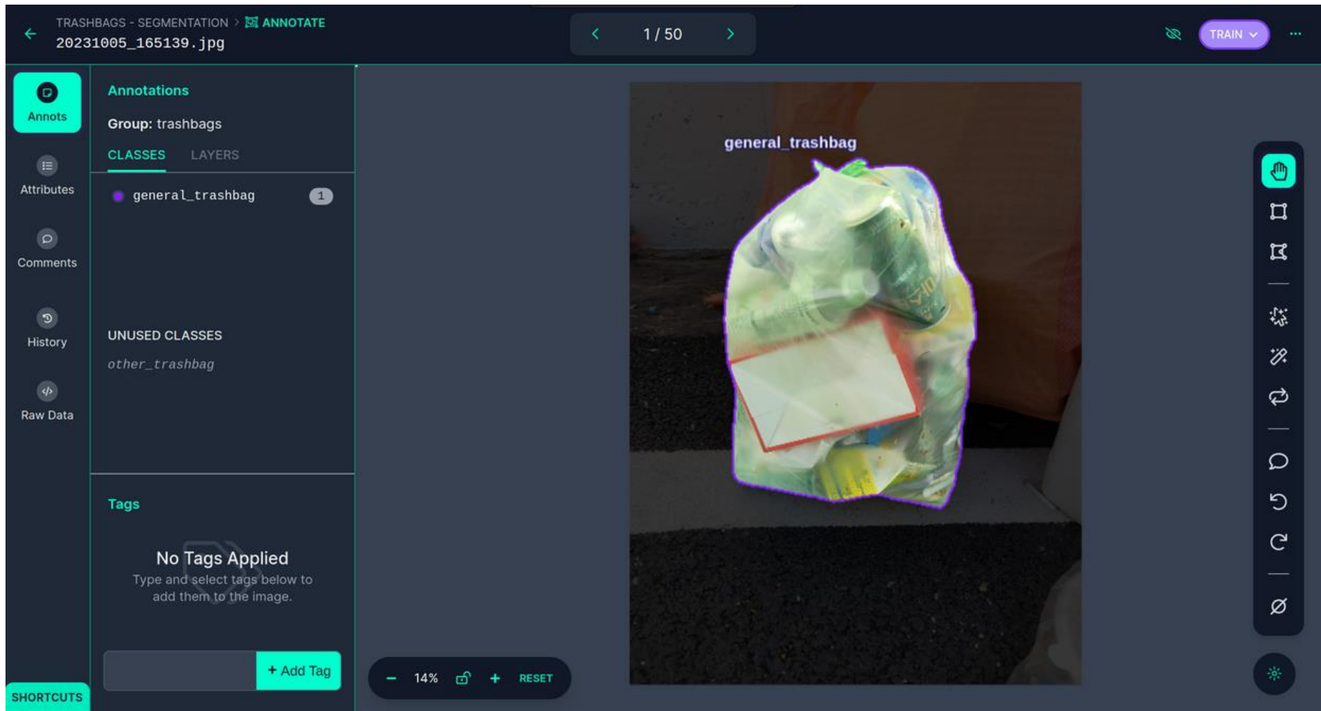
Advantages:

- Free
- Collaborate with Team members (assign images)
- Work is saved and can stop and carry on anytime
- Super AI makes segmentation label 1000x faster
- Can export any version of Data labels (v5, pytorch etc)
- Applies augmentation for you (146 --> 630)

Disadvantage:

- Open source - any datasets is exposed to the world
- Very very expensive. ($260)





YOLO

V5 vs V8

Performance Comparison of YOLOv8 vs YOLOv5

Model Size	Detection#	Segmentation#	Classification*
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

#Image Size = 640 *Image Size = 224

Why

YOLO is the fastest object detection AI

YOLO v5

Advantage: has most community support. v8 is new so not much examples

YOLO v8

Advantage: Has tracking. Greatly simplifying the system

Advantage: v8 is slightly more accurate according to benchmarks

TRAINING

Train

```
In 3  # CD into the training directory
      !cd './training_models'

      # Setup/download YOLO
      %pip install ultralytics
      import ultralytics
      ultralytics.checks()

      from ultralytics import YOLO

      ''' NEW      - train a new model from yolo's pretrained coco models (n ~ xl) '''
      # model = YOLO('yolov8s-seg.pt')
      ''' RESUME/EVAL  - by loading a previous model as the weights initialization '''
      model = YOLO('/home/work/PycharmProjects/pythonProject/WoosongVision/training_model/runs/segment/train6/weights/best.pt')

      # TRAIN
      # results = model.train(data='./trashbag_data_v8.yaml', epochs=30, batch=4, amp=False)  # train the model

      # EVALUATE
      # results = model.val()  # evaluate model performance on the validation set - produces results charts

      # PREDICT
      results = model('/home/work/PycharmProjects/pythonProject/WoosongVision/datasets/real_truck_demo/test1.webm')  # predict on an image

      Ultralytics YOLOv8.0.200 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (NVIDIA GeForce GTX 1660 SUPER, 5934MiB)
      Setup complete ✅ (12 CPUs, 31.2 GB RAM, 444.6/464.3 GB disk)


      WARNING ⚠️ inference results will accumulate in RAM unless `stream=True` is passed, causing potential out-of-memory
      errors for large sources or long-running streams and videos. See https://docs.ultralytics.com/modes/predict/ for help.

      Example:
          results = model(source=..., stream=True)  # generator of Results objects
          for r in results:
              boxes = r.boxes  # Boxes object for bbox outputs
              masks = r.masks  # Masks object for segment masks outputs
              probs = r.probs  # Class probabilities for classification outputs

      video 1/1 (1/21046) /home/work/PycharmProjects/pythonProject/WoosongVision/datasets/real_truck_demo/test1.webm: 352x640 1 general_trashbag
      video 1/1 (2/21046) /home/work/PycharmProjects/pythonProject/WoosongVision/datasets/real_truck_demo/test1.webm: 352x640 1 general_trashbag
      video 1/1 (3/21046) /home/work/PycharmProjects/pythonProject/WoosongVision/datasets/real_truck_demo/test1.webm: 352x640 1 general_trashbag
```

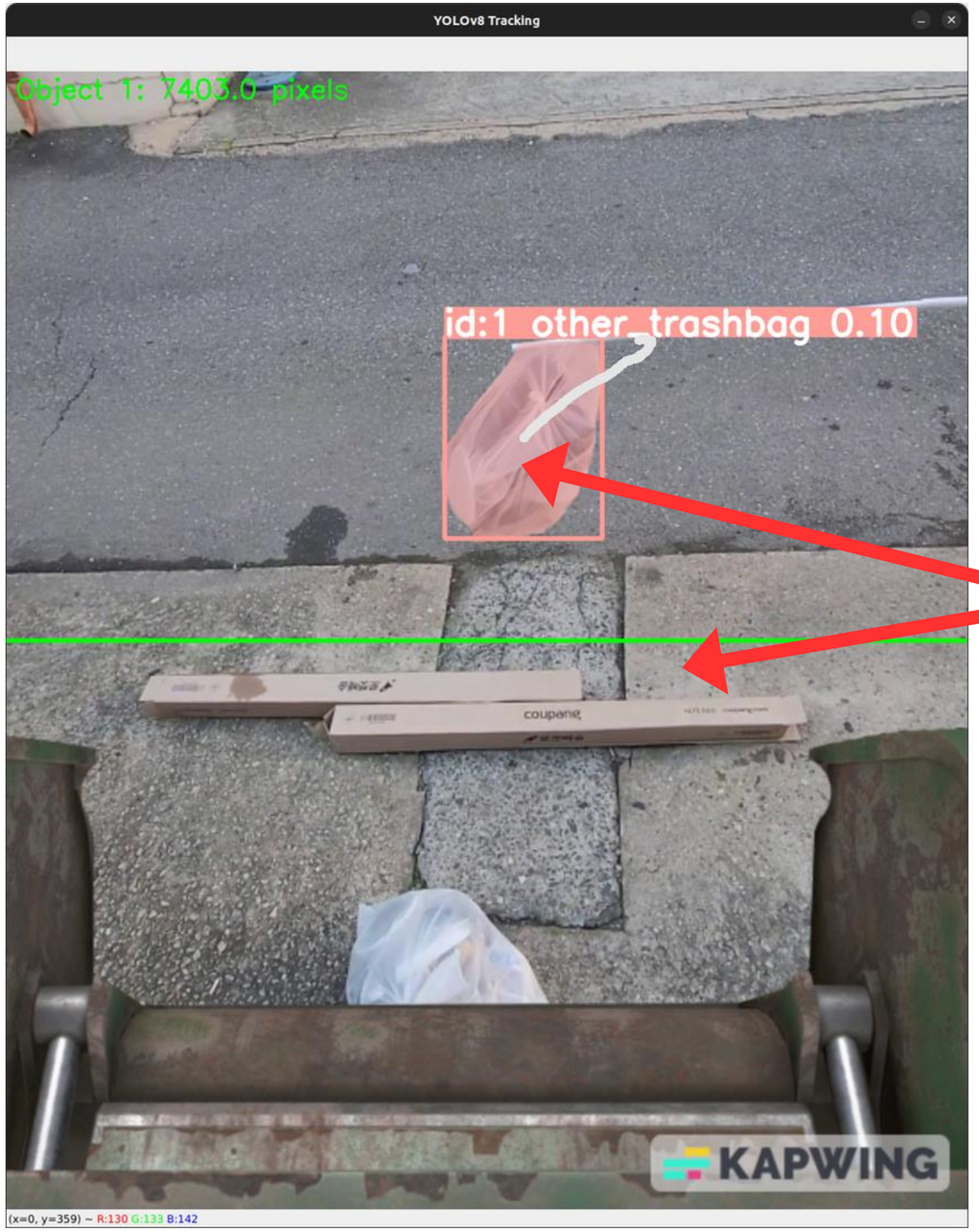
```
Wed Dec 13 22:06:24 2023
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.129.03             Driver Version: 535.129.03   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA GeForce GTX 1660 ...    Off | 00000000:08:00.0  On |                  N/A |
| 36%   51C    P0              34W / 125W |   1698MiB /  6144MiB |      4%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A      4436      G   /usr/lib/xorg/Xorg                          563MiB |
|    0   N/A  N/A      4670      G   /usr/bin/gnome-shell                        117MiB |
|    0   N/A  N/A      6120      G   ...1746269,16441746852891485966,262144      220MiB |
|    0   N/A  N/A     17082      G   ...,WinRetrieveSuggestionsOnlyOnDemand       54MiB |
|    0   N/A  N/A     64919      G   ...ures=SpareRendererForSitePerProcess       24MiB |
|    0   N/A  N/A    142038      C   ...ts/pythonProject/venv_v8/bin/python      702MiB |
+---------------------------------------------------------------------------------------+
```

GPU: GTX 1660 super 6GB

CPU: AMD Ryzen 5 5600X 6-Core

Memory: 32GB DDR4 3200Mhz

Cava



How to:

Detected “loaded” bag

All the trash bags are detected and tracked every frame

If a bag crosses the center “loaded line”,
the bag is marked as “loaded”

Loaded bags are now ignored.

Cava

DEVELOPMENT

PROCESS

- Research
- Data collection V1: Take pictures of trash in the street (145 photos)
- Create Dataset V1: Roboflow - 645 images (augmented)
  - Bounding box []
- YOLO prototyping
- YOLOv5
  - many experiments - Small VS Large
- Create Demo Truck video
- YOLOv8
  - improved results
- Created Dataset V2: Roboflow - 645 images (augmented)
  - Segmented
- Changed to segmentation YOLO
- Implemented tracking
- Final model V1: YOLOv8 Nano - Dataset V2 -
  - Used in Presentation demo
- Finalise Jetson
  - Setup and running (only CPU - Failed to setup Nvidia+CUDA+cudNN
  - ARM architecture is different setup to standard PC
- Create Dataset V2: 1,012 images (augmented into 3000+)
  - Received demo truck videos from NetVision
  - Created new dataset from the trash bags in the videos
- Created model V2:
  - YOLOv8 - Nano - x hours GPU training time

TIPS AND TRICKS

- *Take many notes*
  - *This is a learning process*
  - *Too much to hold in the mind*
  - *Constantly coming back to notes for code commands etc.*

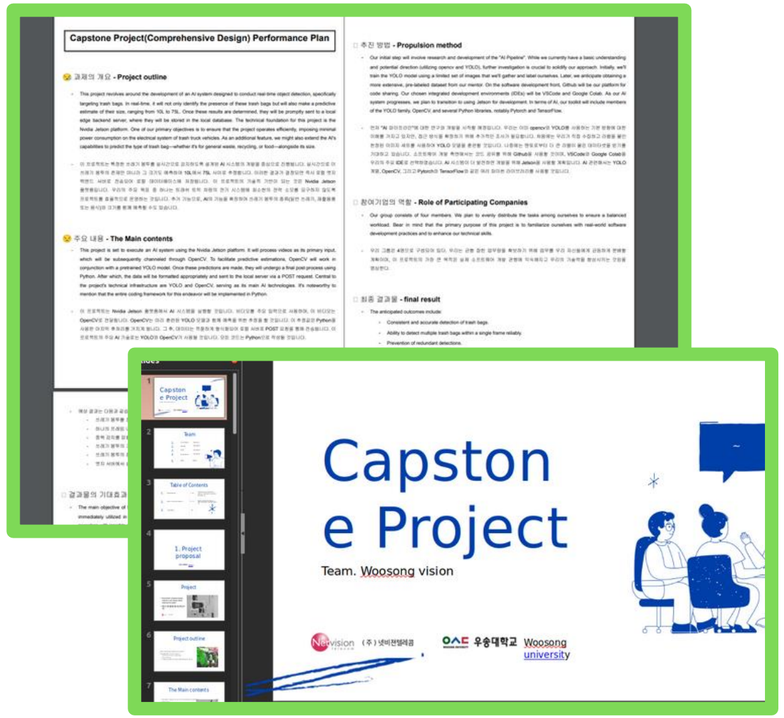
- *Experimenting*
  - *You do not know A –→ Z*
  - *You figure it out by trying step a, then step b, step c…*
  - *Make your code “experiement first, final version later”*
    - *Small test dataset, test code, etc*

- *AI is maturing*
  - *If you want to make a website, you can do so in 1 click nowadays*
  - *But AI feels brand new.*
  - *We faced many issues from start to finish*
  - *E.G setting up GPU was very challenging. And failed on Jetson*
    - *GPU Memory was not clearing, so manually using Nvidia terminal commands, clear the GPU attempt*

EXTRA BACKGROUND

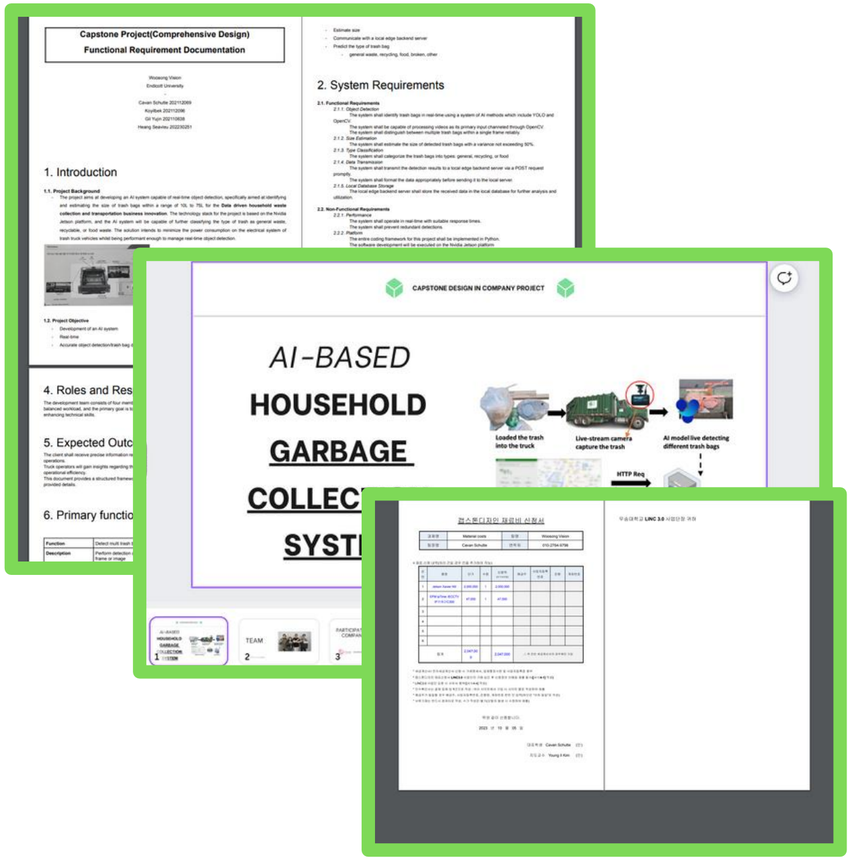
FULL SOFTWARE DEVELOPMENT PROCESS

This project followed the full SW process



Major documents:

- Project Application
- Project Proposal
- Functional Requirements
- System Architecture
- High Level Design
- Low Level Design
- Final Presentation and Demo
- Final Report



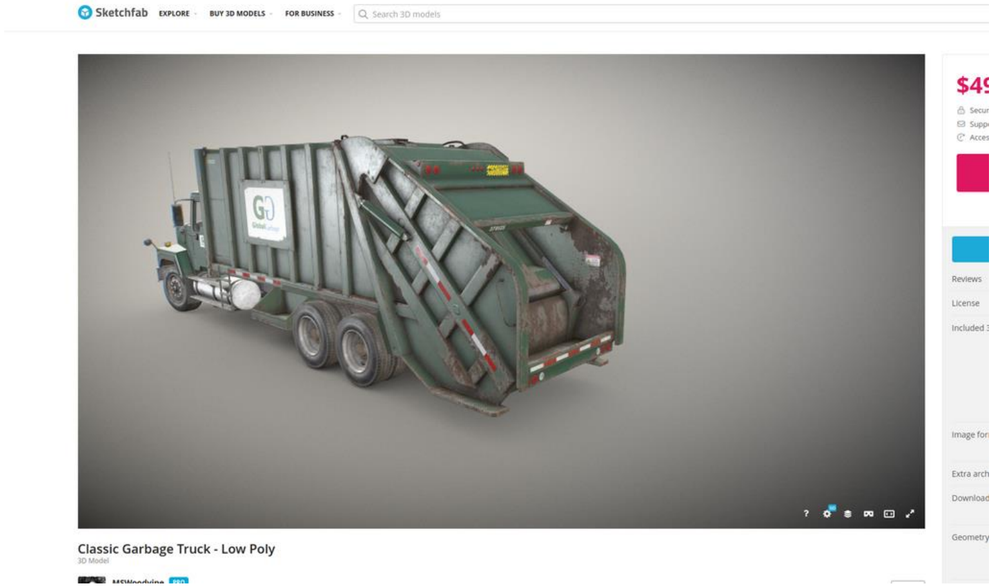
Jenn

Demo video

Need something to develop and test with

- made a mock video
- All AI Pipeline testing will be done on this video

The video simulates trash being loaded into back of truck



1ST PLACE

CAPSTONE DESIGN COMPETITION

Jenn

SPECIAL THANK YOU TO PROFESSOR 김영일

THANK YOU

⭐ ***Woosong Vision***