|  |  |  |
| --- | --- | --- |
| **Software Challenge** | | |
| 130\_Recruitment | Date: | 18/04/2025 |
| Performed by: | Rad Emilian-Antonio |

A picture containing tray, tableware, dishware

Description automatically generated

# Introduction

The software challenge is a loose replica of one of the many challenges we face daily here at .lumen.

A clear and open mind is recommended, as the challenge might require some creativity. If you're feeling ready, read on.

Recommended time limit: ~ 10 hours

# Challenge Description

Together with this document, you have access to two videos, which were recorded with a .lumen Glasses prototype while walking through Cluj-Napoca, in different scenarios (sidewalk, park, crossing the street, etc.). challenge\_color\_848x480.mp4 is an 848x480 @ 30 FPS RGB stream, and challenge\_depth\_848x480.mp4 is an 848x480 @ 30 FPS Depth stream (grayscale). These are the outputs from the [Intel Realsense D455](https://www.intelrealsense.com/depth-camera-d455/) used in the .lumen Glasses prototype.

Your mission, should you choose to accept it, is to come up with a list of 5 meaningful features that can be developed loosely based on the video(s), and \***start**\* implementing \***one**\* of them. Your solution must use at least one frame from at least one of the videos. That's it.

*Note*: Try coming up with your own ideas before reading the examples, as they might limit your perspective. If you're having a hard time coming up with new ideas, you can freely choose/adapt from the examples below.

Some ideas: track / compute the distance to different objects or points of interest in the frame (signs, cars, humans, traffic lights, etc.), detect potentially dangerous scenarios (crossings, stairs, water, etc.), segmentation of the surfaces in front, visual odometry - record and predict user motion, read text from relevant places/signs, navigation based on approximate human movement, voice commands and/or notifications ("Is the light green?", "Car incoming", etc.)

As this challenge is \***very**\* open-ended, you are free to focus and spend most of your time on what you do best be it research, algorithms / implementation, etc.

## Recommendations for chapters 3 and 4:

- You are free to use any software available online. The internet is your friend.

- Write down ideas, thoughts, issues you had and how you overcame them - be brief and on point, we're all engineers

- What different solutions are available and why some were chosen over others

- Extra points for well documented research & code

- Track your time

- Depending on the task you choose, it might take considerably more than 8 hours to implement. In this case, you are free to choose any part of the implementation you want, but make sure to explain how the other parts are supposed to work together.

At the end, send this document with chapters 3 and 4 filled in, and any other code/video/image/links (using a service such as [WeTransfer](https://wetransfer.com/) if needed) as a single .zip archive to [marina@dotlumen.com](mailto:marina@dotlumen.com) and [diana@dotlumen.com](mailto:diana@dotlumen.com) with the subject "[SW-Challenge2] Firstname\_Lastname".

We don't work with text and code; we work with the person behind the keyboard. The point of this challenge is to get to know that person as much as possible.

**Have fun!**

# Ideas & Research

##### Detect all the street signs and display a text of what it means if it’s applied for pedestrians to (crossroads, school zone, temporary detours, accessible pedestrian route etc) also detect the semaphores and based on their color suggest the user what to do

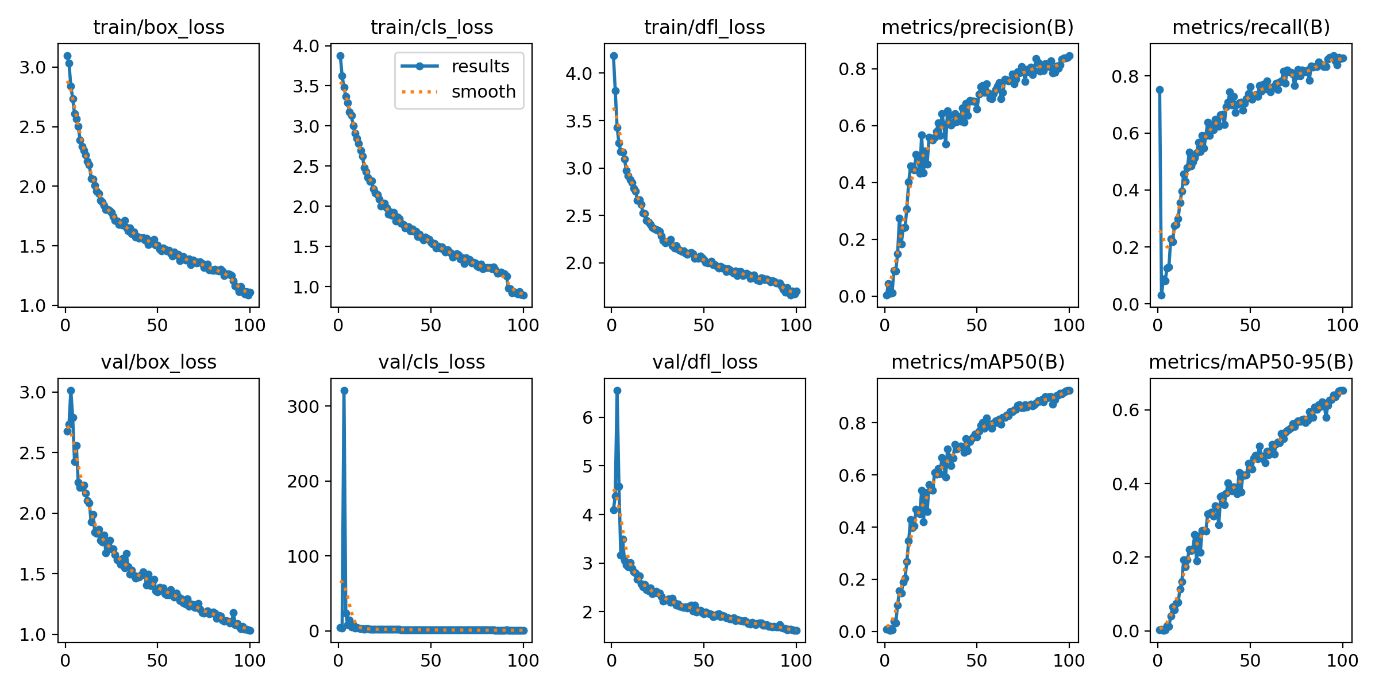
1. Detect trash bins within the video footage and based on the user's movement, calculate the distance between consecutive bins. Rough estimates in minutes until the user reaches the next trash bin.
2. Identify humans and the distance between you and them.
3. Saves to a file or displays on the glasses screen or on the saved video all the visible text in the area of interest if you look at it more than 4-5 seconds (performs a quick OCR)
4. Based on geolocation (I don’t know if it’s implemented in the glasses) it can guide you to different locations giving you instructions
5. Surface segmentation for safe walking, detecting different types of ground(grass, pavement) and provide alerts like (“Grass ahead in x meters”), also detect irregularities(stairs, curbs, elevation changes etc)

For the implementation part of this challenge, I decided to go with the second idea: detecting trash bins in video footage and giving people a rough estimate of how far the next one is, based on their movement. A lot of times, people litter simply because they don’t see a bin nearby and don’t know how long they’ll have to carry their trash. This solution helps by giving them that small but useful piece of info just like a GPS, but for keeping the streets clean. It’s a simple idea that could make a big difference in everyday behavior.

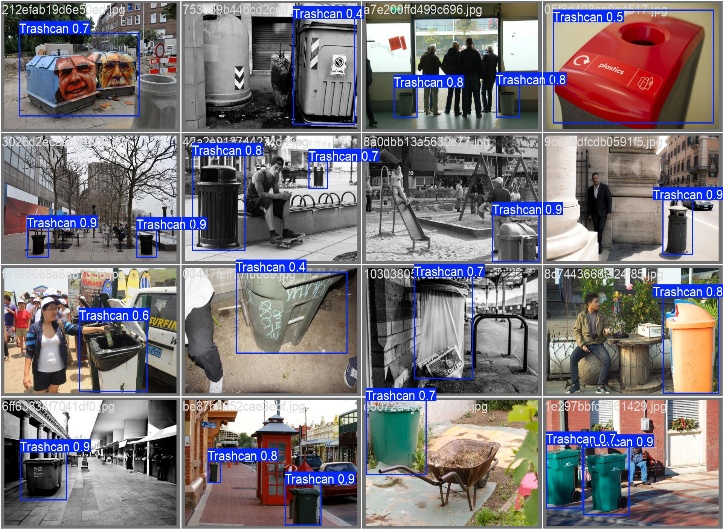
For the research phase of this project, my initial plan was to use the YOLOv8n(2) model to detect trash bins directly. However, after running a few tests, I discovered that the pre-trained model didn’t include a label for trash bins or any closely related categories. As a result, it failed to detect them entirely, making the first step of the idea (identifying trash bins in video footage) impossible with the default setup.

I tried to look up online some datasets already annotated for my needs but I had no luck and it was taking too much time so to work around this, I decided to build a custom solution. I collected around 800 images featuring trash bins from Open Images Dataset V7(3), making sure the photos included a variety of angles, lighting conditions, and urban settings to help the model generalize well. Once I had the images, I used an online annotation tool called CVAT(1) to manually label every trash bin in the dataset, even though the images we’re annotated but they weren’t supported by the model that I wanted to use.

With the annotated dataset ready, I moved on to training. I used the YOLOv8n architecture as the base and trained it for 100 epochs using my custom dataset. This allowed the model to learn to recognize trash bins with a much higher accuracy, even in busy or cluttered environments.

After the 100 epochs that trained the YOLOv8n model these are the result graphics that I got 

And some examples from the labeled batches and the predicted ones (the dataset)

After training the YOLOv8n model on a custom dataset of 800 annotated images featuring trash bins, I tested the AI on the video provided for the challenge. Despite the relatively small dataset, the model performed reasonably well, correctly identifying trash bins in most relevant frames. I also analyzed the results by generating a binary detection array (1 for detection, 0 for none), grouped detections into clusters, and calculated the number of valid trash bin appearances throughout the video. Additionally, I measured the number of ones and zeros (non-detections) between these clusters to better understand trash bin spacing and estimate the time between available disposal opportunities. Overall, the results were promising and show that even with limited training data, a focused detection model can contribute meaningfully to reducing litter by guiding people to the nearest trash bin.





Time spent:

##### Research for the project: 3-4 hours

##### Preparing the YOLOv8n model 2-3 hours

* + - * 1. Implementing the code 2-3 hours

Total time spent: 9-10 hours

# Implementation

## Training the YOLOv8n model

Code to train the model:

A screen shot of a computer code

AI-generated content may be incorrect.

And the config file:

A screenshot of a computer code

AI-generated content may be incorrect.

## Splitting the video in individual frames

## A screenshot of a computer program AI-generated content may be incorrect.

## Using the custom model on the provided video

Splitting the video into individual frames (466 = 7.44 minutes)

All the annotated frames are available on

A screenshot of a computer program

AI-generated content may be incorrect.

## Calculate when the next trash bin should be visible

A screenshot of a computer program

AI-generated content may be incorrect.

# References

1. <https://www.cvat.ai/>
2. <https://yolov8.com/>
3. <https://storage.googleapis.com/openimages/web/index.html>
4. <https://www.youtube.com/watch?v=m9fH9OWn8YM&t=3061s>

# Link To Gihthub Repository