National College of Ireland

Smart City Surveillance: Automated Street Waste Detection Using Live Camera

MSc Research Project
Master of Science in AI

Rehman Sarwar
Roll No.x23245875

School of Computing
National College of Ireland

Supervisor: SM Raza Abidi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rehman Sarwar
Student ID:	X23245875
Programme:	Master of Science in AI **Year:** 2024-2025
Module:	Research Master in AI
Supervisor:	SM Raza Abidi
Submission Due Date:	12 Dec 2024
Project Title:	Smart City Surveillance: Automated Street Waste Detection Using Live Camera
Student Name:	Rehman Sarwar

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Rehman Sarwar

Date: 12 Dec 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Smart City Surveillance: Automated Street Waste Detection Using Live Camera

Rehman Sarwar
X23245875@student.ncirl.ie

1. Introduction

This configuration manual provides a detailed guide for setting up and executing the implementation of the research project titled " Smart City Surveillance: Automated Street Waste Detection Using Live Camera". The project aims to Automated Street Waste Detection system based on YOLOv8 models for identifying and classifying urban waste in real time as part of smart city models.

2. System Specification

To ensure the successful execution of the code, your system should meet the following minimum specifications:

- **Environment:** Google Colab
- **Memory:** 53 GB RAM.
- **Storage:** 235 GB free SSD space.
- **GPU:** L4 GPU 24 GB, A100 GPU 24.
- **Python Version:** Python 3.7

3. Softwares Used

The following software and libraries are used in this project:

3.1 Programming Language

- Python: The primary programming language used for implementing the model and processing data.

3.2 Python Libraries

- **Ultralytics:** For importing yolo algorithm.
- **Shutil:** Copy directories and manage file operations like moving files from one location to another.

3.3 Development Environment

- Google Colab.

4. Dataset Source

The dataset used in this project is sourced from TrashNet: A set of annotated images of trash that can be used for object detection Computer Vision Project. It consists of 6049 images and have six different class like cardboard, glass, paper, trash metal and plastic. In the division of data is shown in Table 1. Where 87% of data images are used for training both models of YOLOv8.

Table 1: Data division

Parameter	Description
Total Images	6,046
Train Set	5283 images (87%)
Validation Set	499 images (8%)
Test Set	264 images (4%)

5. Execution of the Code Implementation

The execution process for the given code implementation can be summarized as follows:

1. **Mounting Google Drive**
   The Google Drive is mounted using google.colab.drive to facilitate reading and writing files stored on the drive. This is crucial for accessing the dataset and saving the output of the YOLO model training and prediction.

```
from google.colab import drive
drive.mount('/content/drive')
```

2. **Installing Required Libraries**
   The required libraries, ultralytics and clearml, and shutil are installed using pip. These libraries are essential for working with the YOLOv8 model, for managing ML experiments and saving train files in drive.

```
!pip install ultralytics -q
!pip install clearml -q
import shutil
```

3. **Training the YOLOv8 Model**
   The YOLOv8 model is initialized with a pre-trained weight file (yolov8s.pt same as yolov8l) and trained on a custom dataset specified by data.yaml. The training runs for 100 epochs.

```
from ultralytics import YOLO
model = YOLO('yolov8s.pt') # yolov8l for YOLO Large
model.train(data = "/content/drive/MyDrive/trashnet/data.yaml" , epochs = 100)
```

4. **Making Predictions**
   The trained model (best.pt) is used to predict objects in a test dataset. The predictions are saved as images and text files.

```
infer = YOLO("/content/runs/detect/train/weights/best.pt")
infer.predict("/content/drive/MyDrive/trashnet/test/images", save = True , save_txt =True)
```

5. **Saving Results to Google Drive**
   The results of predictions and training are copied to specific directories in Google Drive using the shutil library. This ensures the outputs are saved for later use and analysis.

```
source = "/content/runs/detect/predict"
destination = "/content/drive/My Drive/trashnet_predict100epochs_ss"

shutil.copytree(source, destination)


source = "/content/runs/detect/train"
destination = "/content/drive/My Drive/trashnet_train100epochs_ss"

shutil.copytree(source, destination)
```

6 Real Time Detection

- capturing video from the webcam and displaying the output
- best_s.pt is my model name of YOLOv8s and we can give directory name of YOLOv8L
- Import cv2 capturing video from the webcam

```
import cv2
from ultralytics import YOLO
model_path = r"/Users/rehmansarwar/Downloads/best.pt"
model = YOLO(model_path)
```

- Initialize Webcam
- A while loop captures frames from the webcam.
- Each frame is processed using the YOLO model by passing the frame to model().
- The results object contains the detection results for each frame.
- If results are a list (indicating multiple results), it iterates through the list and plots detections on the frame
- If results are not a list (a single result), it plots detections directly

```
cap = cv2.VideoCapture(0)

if not cap.isOpened():
    print("Error: Could not open webcam.")
    exit()

while True:
    # Real-Time Frame Processing:
    ret, frame = cap.read()
    if not ret:
        break

    # Perform detection
    results = model(frame)
```

```
# Check if results is a list and handle each result
    if isinstance(results, list):
        for result in results:
            annotated_frame = result.plot()
            # Display the resulting frame
            cv2.imshow('Webcam YOLOv8 Detection', annotated_frame)
    else:
        annotated_frame = results.plot()
        # Display the resulting frame
        cv2.imshow('Webcam YOLOv8 Detection', annotated_frame)
# Press 'q' to exit
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the capture and close windows
cap.release()
cv2.destroyAllWindows()
```

Figure 1 shows the real detection of carboard with 84 percent accuracy on our proposed model.

Figure 1 Real Time detection of Carboard

Figure 2 shows the real detection of glass with 56 percent accuracy on our proposed model.

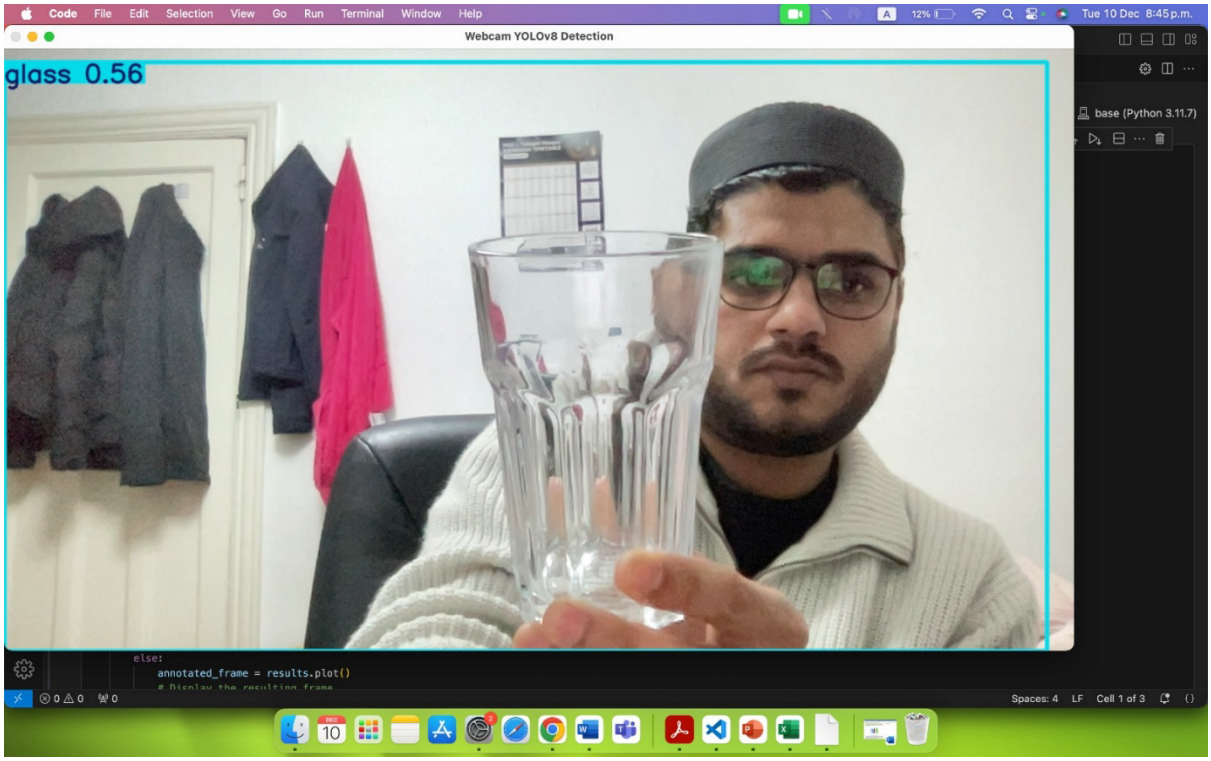

Figure 2 Real Time detection of Glass

Figure 3 shows the real detection of cardboard with 6 , my metal glass with 27 percent accuracy on our proposed model

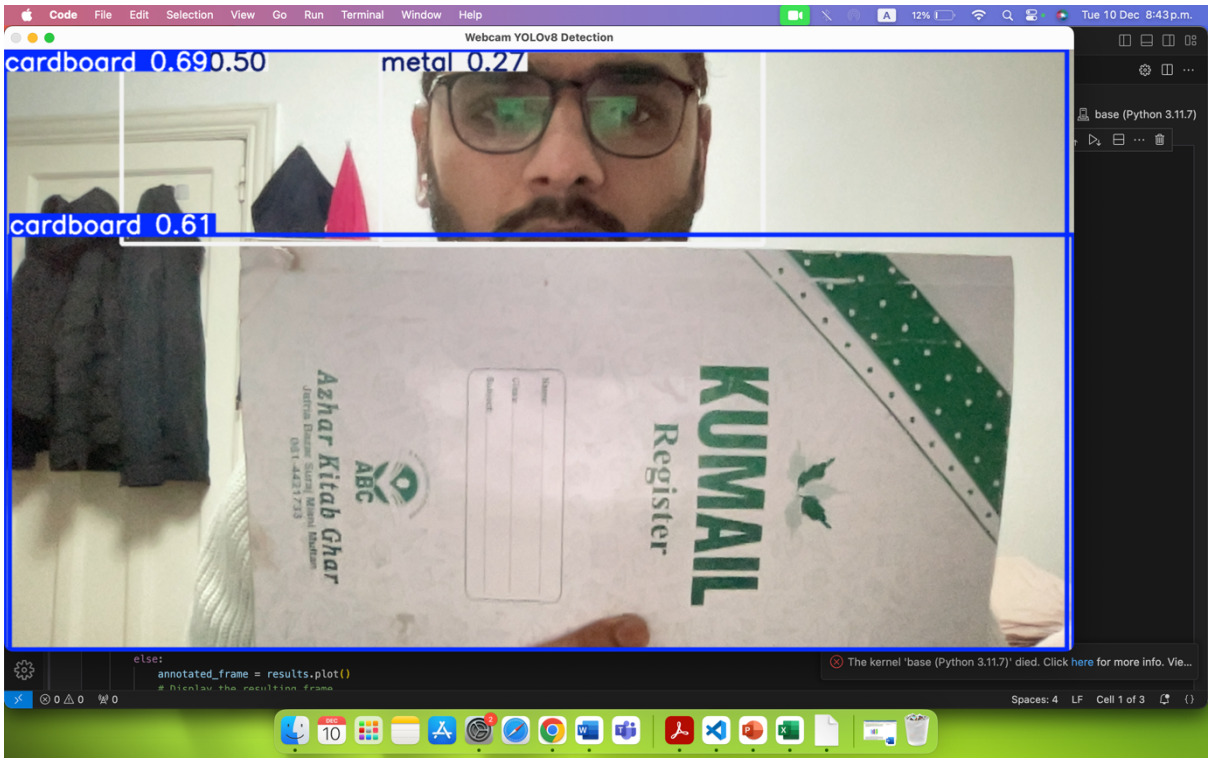

Figure 3 Real Time detection of metal, cardboard in one frame