Detected vehicles and recognize license plates using YOLOv5 for vehicle detection and Tesseract OCR for license plate recognition. Assigned unique vehicle IDs based on license plate and timestamp using DeepSort and Python’s datetime library for efficient tracking. Tracked vehicle movement from entry to toll points with OpenCV, using unique IDs to identify the same vehicle across different camera locations. Calculated the distance between predefined toll spots using OpenCV and NumPy for accurate toll fee computation based on camera locations. Automatically collected toll fees at designated spots based on the distance travelled.

1. **Setup YOLOv5, Deep SORT, and OCR Modules**

**A screen shot of a computer

Description automatically generated**

**Explanation:**

* import torch: Imports PyTorch, a deep learning library used to load the YOLOv5 model.
* import cv2: Imports OpenCV, a library for computer vision tasks like image processing and video streaming.
* import numpy as np: Imports NumPy for efficient mathematical operations, such as calculating distances.
* import pytesseract: Imports the Tesseract OCR library to recognize text from images (e.g., license plates).
* from datetime import datetime: Imports datetime for handling timestamps (to track entry/exit times).
* from deep\_sort\_realtime.deepsort\_tracker import DeepSort: Imports the Deep SORT tracker to assign and maintain unique IDs for vehicles.
* from collections import defaultdict: Imports defaultdict from Python’s collections module, allowing us to store data with default values for each vehicle ID (e.g., plate, timestamps).

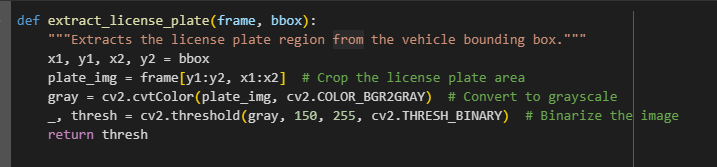
**A computer screen shot of text

Description automatically generated**

**Explanation:**

1. torch.hub.load(...): Loads the YOLOv5 model (pre-trained on the COCO dataset). The yolov5s model is a lightweight version suitable for real-time detection.
2. tracker = DeepSort(...): Initializes a Deep SORT tracker.
   * max\_age=30: The tracker will keep IDs assigned to vehicles even if the vehicle is not detected for 30 frames.
   * n\_init=3: A new object needs to be detected at least 3 times before it is assigned a unique ID.
   * nn\_budget=100: Controls the memory usage for tracking, limiting the history to the 100 most recent detections.
3. pytesseract.pytesseract.tesseract\_cmd: This line points Tesseract to its installation path (required only on Windows). You can omit it if using Linux/macOS.

**2. Define Helper Functions for License Plate Detection and Recognition**



**Explanation:**

* extract\_license\_plate(...): This function extracts the license plate region from a vehicle’s bounding box.

1. x1, y1, x2, y2 = bbox: Extracts coordinates of the bounding box (top-left and bottom-right corners).
2. plate\_img = frame[y1:y2, x1:x2]: Crops the part of the frame containing the license plate.
3. cv2.cvtColor(...): Converts the cropped image to grayscale for better text extraction.
4. cv2.threshold(...): Applies binary thresholding to enhance text visibility for OCR.

A screen shot of a computer code

Description automatically generated

**Explanation:**

* recognize\_plate(...): This function uses Tesseract OCR to extract the license plate text.

1. pytesseract.image\_to\_string(...): Extracts text from the license plate image.
2. --psm 8: Tells Tesseract to treat the image as a single word for better accuracy.
3. ''.join(...): Filters out non-alphanumeric characters to keep only valid plate text.
4. .upper(): Converts the text to uppercase.

**3. Vehicle Tracking and Toll Fee Calculation**

****

**Explanation:**

vehicle\_data: Stores information for each vehicle, including its license plate, entry/exit times, and toll fee. Using defaultdict ensures each new vehicle ID gets a default dictionary with these fields.

**A black screen with white text

Description automatically generated**

**Explanation:**

* calculate\_distance(...): Computes the Euclidean distance between two toll points using NumPy. This will help calculate the toll fee based on how far the vehicle travelled.

**A black screen with green and orange text

Description automatically generated**

**Explanation:**

* compute\_toll\_fee(...): Multiplies the distance travelled by a rate of 0.5 currency units per distance unit.

**A computer screen shot of a program

Description automatically generated**

**Explanation:**

1. formatted\_detections: Formats YOLOv5 detections for Deep SORT.
2. tracker.update\_tracks(...): Updates the tracker with current frame detections.
3. track.is\_confirmed(): Ensures only confirmed tracks are processed.
4. track.to\_ltwh(): Converts bounding boxes to (x, y, width, height) format.
5. extract\_license\_plate(...): Extracts license plate image.
6. cv2.rectangle(...): Draws bounding box.
7. cv2.putText(...): Displays vehicle ID and plate text.
8. **Main Program to Capture Video and Process Frame**

**A computer screen shot of a program code

Description automatically generated**