Traffic Congestion Detection System

I. APPROACH

The YOLOv8 is a deep learning model developed for real-time object detection. It is already pre trained and can be imported from the Ultralytics library available online. The deep learning model can handle multiple object detection classes at once. This model was used for the traffic congestion detection system to individually detect vehicles on the road and externally control the microcontroller through serial communication. This study aims to apply the pre-trained model of YOLOv8 without any extra training and fine-tuning of weights to test the capabilities of the pre-trained model on multiple applications of traffic systems.

II. METHODOLOGY

The traffic congestion detection system starts with importing the YOLOv8 deep learning model into the project directory. Using the OpenCv pipeline, we can extract frames from the video capture device and individually feed them into the CNN model. The YOLOv8 model outputs all the individual objects detected with their bounding box coordinates. Since the YOLOv8 model, by default, outputs all the objects it recognizes, an additional layer of filtering is required to only allow vehicles such as cars, buses, trucks, and motorcycles. Afterwhich, OpenCV then displays the results in a separate window with the corresponding bounding boxes highlighting the individual objects detected by the CNN model.

After setting up the object detection software, communication between the python script and microcontroller is established using the pyserial library. It is responsible for sending and receiving data with external hardware devices using serial communication. A separate program is built for the microcontroller to respond to specific serial communication commands and turn on specific LEDs in order. Afterwhich, pyserial is then imported into the python script and serial communication with the desired port is established dynamically when running the main.py python code.

After establishing the connections with the python script and microcontroller using serial communication. External logic is then implemented on the traffic detection system to dynamically set the traffic state when a vehicle is detected in a region of interest. By default, the traffic light should be set to STOP when no vehicles are detected. Upon detecting one or more vehicles, the countdown timer initiates before transitioning from STOP to GO. The countdown timer is also inversely proportional to the number of vehicles detected leading to a faster countdown when more vehicles are detected on the road.

III. RESULTS AND DISCUSSION

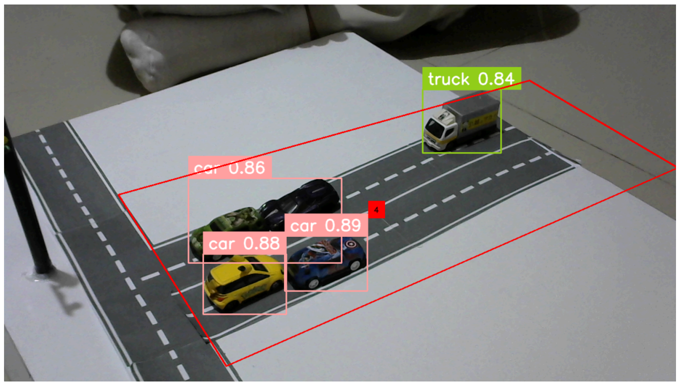

Fig. 1. YOLOv8 Detection System

The implementation of the Traffic Congestion Detection System yielded several key findings that highlight both of it strengths and areas for improvement

Inconsistent Lighting

The model's ability to detect vehicles such as trucks, cars, and motorcycles was hindered by inconsistent lighting conditions. For optimal object detection, it is important that the video footage is well-lit to enhance the contrast and visibility of the vehicles.

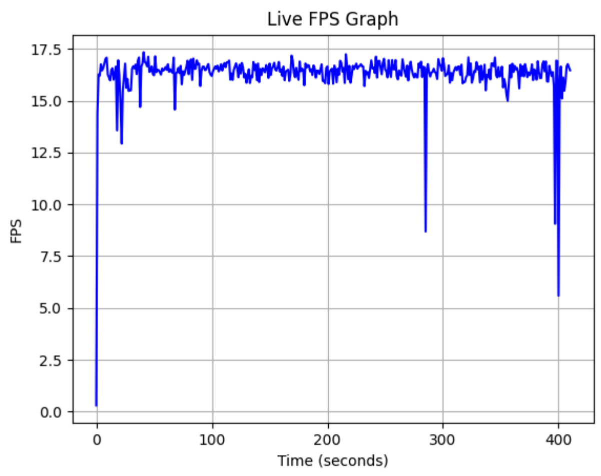

Fig.2. Traffic Detection System FPS

Tracking Inconsistencies

Even after signal smoothing, in rare cases, inconsistencies in detecting vehicles were observed since the object detection is being done frame by frame which can also affect the signal transfer to the arduino, disrupting the control of traffic lights.

Rapid Detection

One significant advantage observed is the models' rapid detection capability. Once the conditions are met, the model quickly identifies cars, trucks and motorcycles within the detection region, ensuring efficient traffic monitoring.

Clustered Vehicles

When the vehicles are too close or clustered together, the model tends to detect them as a single object. This misidentification highlights the need to improve spatial resolution in the detection algorithm to accurately between close packed vehicles.

Detail Recognition Issues:

The model struggled to recognize certain vehicles due to its color specifically green, deep purple, and deep red. This suggests that because of the contrast of the object, most of its details cannot be recognized by the model.