Discussions of methodology and edge cases along with suggestions for solutions for implementation of a timer timing vehicles within a certain region of interest.

Problem statement - Detect cars and time them while inside a chosen region of interest (ROI).

Methodology - A pretrained Yolo V4 model has been used to detect vehicles. After choosing an area of interest I have used chrono library to start a timer once any vehicle enters the ROI and stop the timer when that vehicle exits the ROI.

Edge cases -

1. While deploying this system in real-time, we will face issues with accurate detection of vehicles. I first did the assignment with a **background subtraction KNN module of image processing**, but the number of false detections were really high and the number of correct detection were comparatively lower, thus I switched to the **Yolo V4 model** which significantly increased the vehicle detection accuracy.
2. After using the Yolo V4 model the accuracy of vehicle detection increased but the processing time also increased proportionally , thus I switched to **Yolo V4 model tiny** version which has a lesser heavy architecture than Yolo V4 model thus making it more suitable for real time deployments even on edge devices like RPI, mobile phones, etc.
3. In real time systems there can still be detection misses due to significant environment changes throughout the day and under various unpredictable conditions, this is why we can implement a tracking or prediction module which will work alongside the detection model to produce better results , we can use **optical flow with Lucas Kanade** for this use case which will take the detected results of Yolo V4 model as input when it’s available and during model detection misses it will provide with it’s tracking result so that the vehicle still gets spotted smoothly.
4. If there is more than one vehicle present in the same ROI at the same time, two different timers will be needed to be set for both of them, for which tracking algorithms like **optical flow with Lucas Kanade** again can be used which will generate unique ids for multiple cars and thus time them through unique timers based on ids.