Counting Vehicle by using YOLOv8 and DeepSORT algorithm

Introduction

The goal of this project is to improve my skills and help me have deeper unstanding of Computer Vision techniques, this project will help us count how much vehicle passed specific road automatically without human efforts.

Table

Used Python Library:

- Numpy, pandas: These librarys are commonly used in data preprocessing or analysis.
- Utralytics: This library is primarily used for YOLO.
- OpenCV: This provides a real-time optimized Computer Vision library.

Appoaches To Model Building

This section covers 2 techniques that I used for counting vehicle: YOLOv8 and DeepSORT algorithm.

YOLOv8

YOLOv8 is an algorithm realeased by Ultralytics company on 10 January 2023, the company that developed YOLOv5. YOLOv8 provided five scaled versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) and YOLOv8x (extra large), in this project I chosed to use YOLOv8n to pratice. YOLOv8 support multiple vision tasks such as object detection, segmentation, pose estimation, tracking, and classification.

Figure 1 shows the detailed architecture of YOLOv8. YOLOv8 uses a similar backbone as YOLOv5 with some changes on the CSPLayer, now called the C2f module. The C2f module (cross-stage partial bottleneck with two convolutions) combines high-level features with contextual information to improve detection accuracy.

YOLOv8 uses an anchor-free model with a decoupled head to independently process objectness, classification, and regression tasks. This design allows each branch to focus on its task and improves the model's overall accuracy. In the output layer of YOLOv8, they used sigmoid function as activation function for the objectness score, representing the probability that the bounding box contains an object. It uses the softmax function for the class probabilities, representing the object's probabilities belong to each possible class.

YOLOv8 uses CioU and DFL loss function for bounding box loss and binary cross-entropy for classification loss. These losses have improved object detection performance, particularly when dealing with smaller objects.

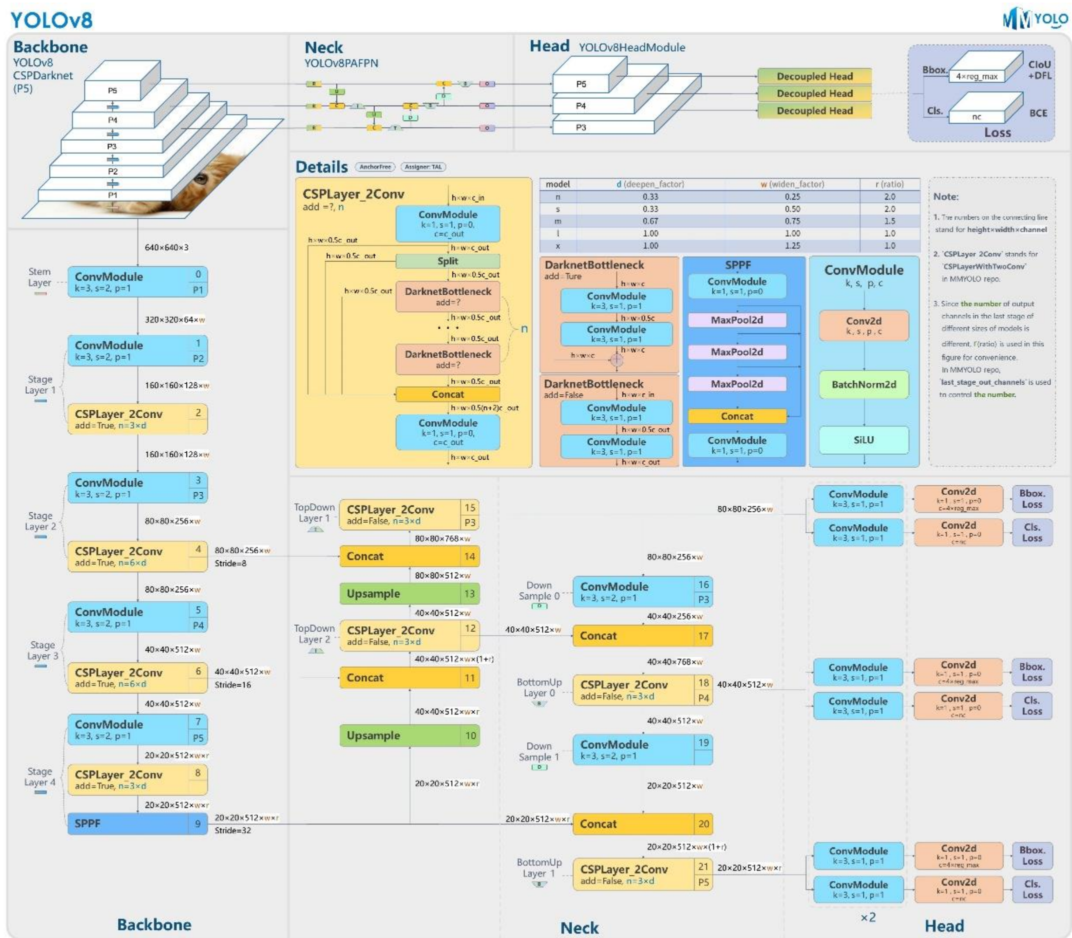

Figure 1: YOLOv8 Architecture

YOLOv8 can be run from command line interface (CLI), or It can also be installed as a PIP package. Evaluated on MS COCO dataset test-dev 2017, YOLOv8n achieved a mAP of 37.3% with an image size of 640 pixels.

DeepSORT

DeepSORT is a Computer Vision algorithm used to track the objects while assigning each of the object a unique. Also, DeepSORT is a Multiple

Object Tracking (MOT) algorithm which is an extension of SORT algorithm. DeepSORT introduces deep learning into the SORT algorithm by adding an appearance descriptor to reduce the identity switches and hence making the tracking more efficient. DeepSORT uses a better association metric that combines both motion and appearance descriptors. DeepSORT can be defined as the tracking algorithm which tracks objects not only based on the velocity and motion of the object but also appearance of the object. Below figure shows detailed how DeepSORT algorithm work.

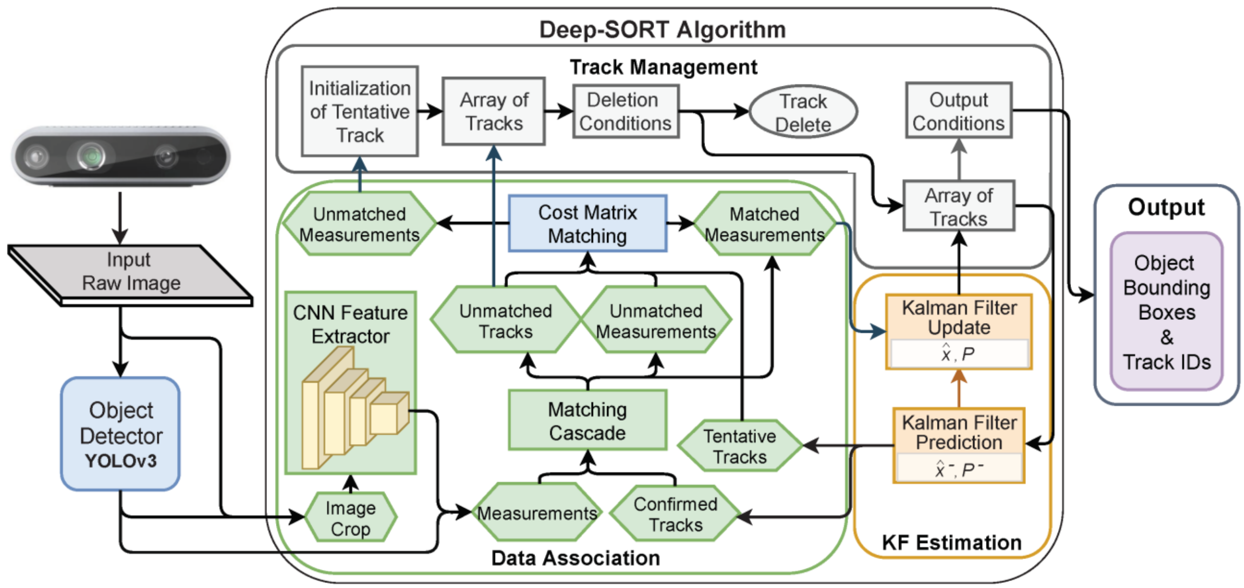

Figure 2: DeepSORT Algorithm

DeepSORT primarily utilizes two algorithms: Hungarian algorithm and Kalman Filter to track the objects. So what are these? We will explore these in the following sections.

Hungarian Algorithm

The Hungarian method is a combinatorial optimization algorithm that solves the assignment problem. In DeepSORT, the hungarian algorithm is used in its matrix format. Example, given a nonnegative n x m matrix, where the element in the i-th row and j-th column represents the cost of assigning the j-th predicted object to the i-th measured object, the Hungarian algorihm finds an assignment of the prediction to measurements, such that each prediction is assigned to one measured object and each measure object is assigned to one prediction, and the total of assignment is minimum.

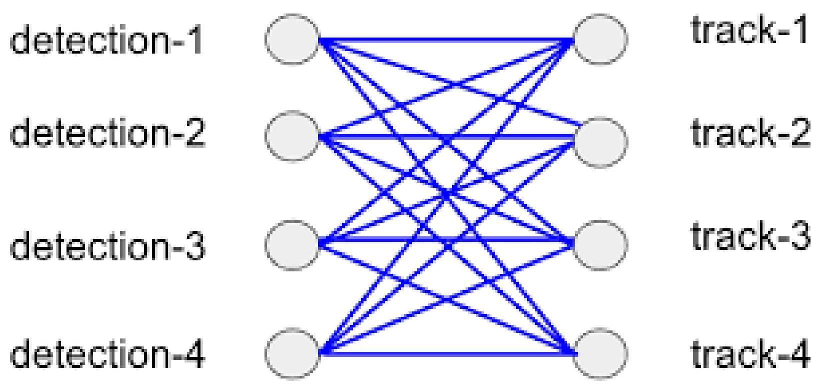

Figure 3 Simple Visualize Hungarian Algorithm

Kalman Filter

Kalman Filter is an algorithm that allows us to estimate the state of a system based on observations or measurements. It is an available tool for various applications, such as object tracking, autonomous navigation systems, economic predict. This is an example illustrates how to use the Kalman Filter for tracking objects and focus on three important feature:

- Prediction of object's future location.
- Reduction of noise introduced by in accurate detections.
- Facilitating the process of association of multiple objects to their tracks.

The basic idea of Kalman Filter is that by using the prior knowledge of the state, the filter makes a forward projection state or predicts the next state.

The Kalman Filter is intended to estimate the state of a system at time k, using the linear stochastic difference equation. It assumes that the state of the system at time k evolved from the prior state at time k-1, expressed in the following form:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

It is always pared with the measurement model z_k that decribes a relation between the state and measurement at the current step k. It is written as:

$$z_k = Hx_k + v_k$$

Where:

- A, a matrix n x m, is the state transition matrix relating the previous time step k -1 to the current state k.
- B, a matrix n x l, is a control input matrix applied to the optional control input u_{k-1} .
- H, a matrix m x n, is a transformation matrix that transforms the state into the measurement domain.
- w_k and v_k represen the process noise vector with the convariance Q and the measurement noise vector with the covariance R, respectively. They are assumed statistically independence Gaussian noise with the normal probability distribution.

$$p(w) \sim N(0, Q)$$
$$p(v) \sim N(0, R)$$

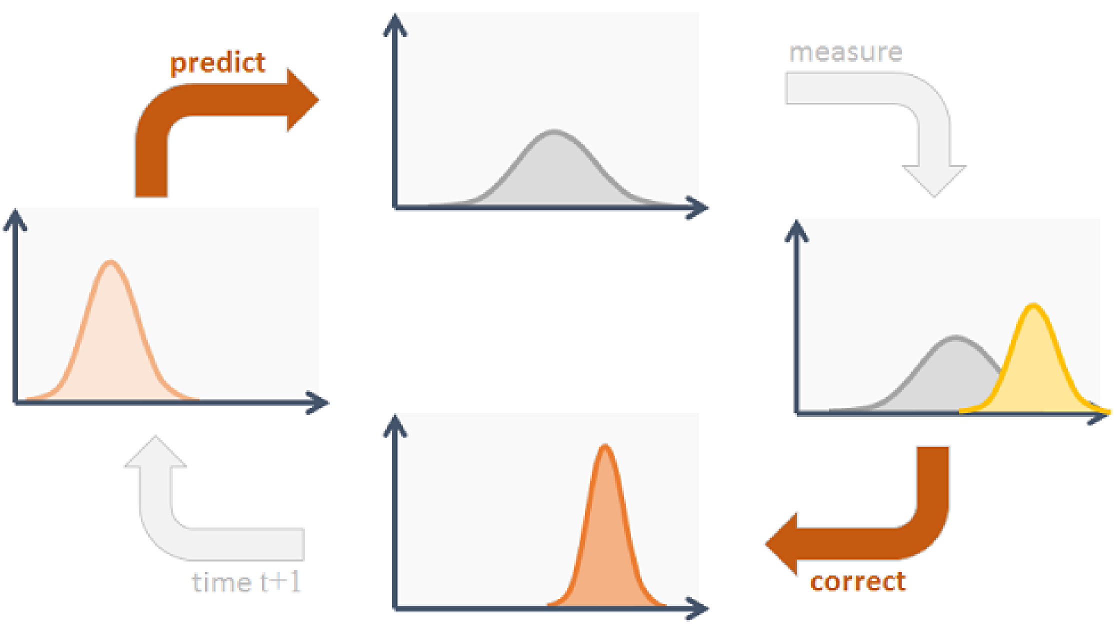

Figure 4: Simple Visualize How Kalman Filter Work

Experiment

In this project, I combine YOLOv8 and DeepSORT to count how many vehicles crossed the line I drawed on the road. Specifically, I use YOLOv8 to detect vehicles on the road, then I use DeepSORT to track the vehicles, each of that will be assigned a unique id, when it passed the line, I will increase amount of passed vehicle and print it on the screen. Below figure is a result of the project.

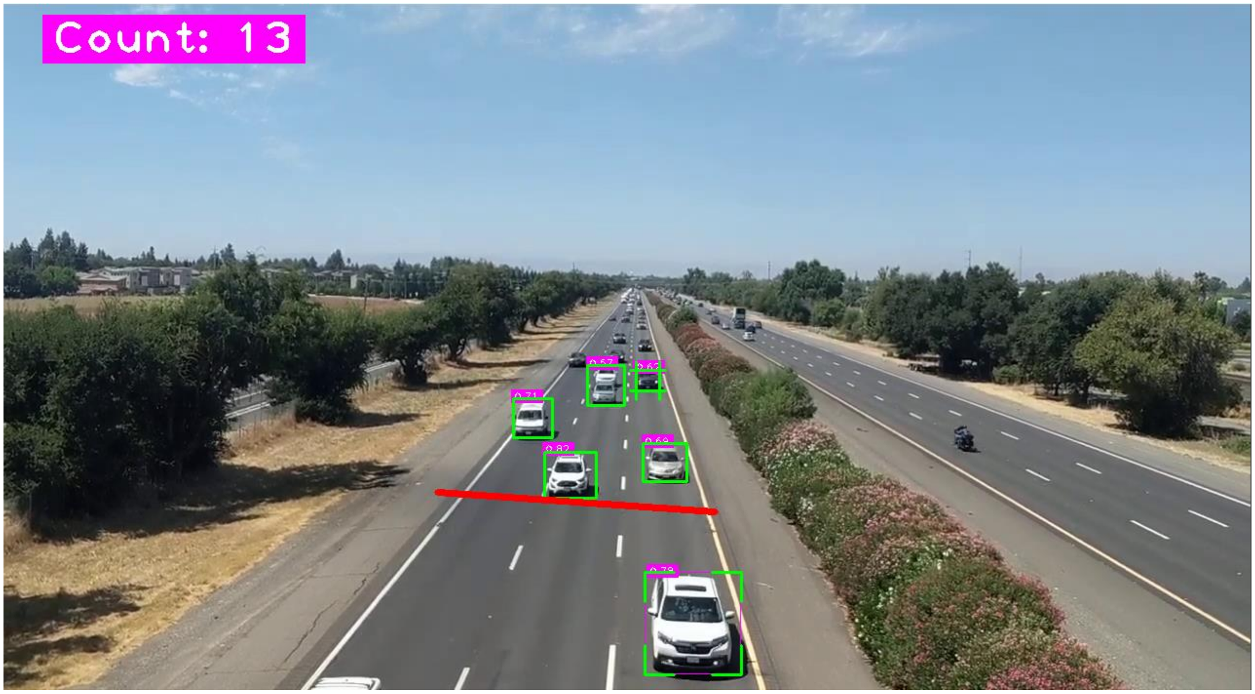

Figure 5: Result

Conlusion

The goal of this project was to learn and improve my skill in Computer Vision. Overall, the project was successful in its showcasing which used DeepSORT helped to reduce amount of ID switches, hence I can count vehicles exactly.