Multi-Object tracking and Trajectory Prediction for Autonomous Vehicles

Vinit Awale
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India
awale.vinit@gmail.com

Prof. Naveen Arulselvan
AI & Robotics Technology Park
ARTPARK, IISC Bangalore
Karnataka, India

***Abstract*—This document specifies the implementation details of the perception module using cameras of a self-driving car. This project was completed as a part of the Summer Internship program at ARTPARK, IISC Bangalore, under the mentorship of Prof. Naveen Arulselvan. The perception module consists of multi-object detection, tracking and trajectory prediction. The multi-object detection is based on the *YOLO v5* algorithm [?]. We have used the *Deep Sort* algorithm for multi-object tracking based on the paper "Simple online and realtime tracking with a deep association metric". The trajectory prediction module is implemented using *PEC Net* based on the paper "It is not the journey but the destination: Endpoint conditioned trajectory prediction". Python and PyTorch framework have been used for the code implementation.**

***Index Terms*—Object detection, multi-object tracking, trajectory prediction, YOLOv5, Deep Sort, PEC Net, Autonomous vehicles**

I. Introduction

Perception is a central problem for any autonomous agent, be it humans, robots or self-driving vehicles. This module helps for a smoother and more reliable control of the car using the path-planning module of the autonomous agent. It can also aid in pose estimation. For our project, we have included the following sub-modules for the perception:

- Multi-object detection using the *YOLOv5* algorithm.
- Multi-object tracking using the *Deep Sort* algorithm.
- Trajectory prediction using the *PEC Net* algorithm.

Object detection in the context of autonomous driving refers to detecting the objects present in the scene (making use of the camera sensors on the autonomous vehicle) by making bounding boxes surrounding the detected objects. This is followed by identifying the class of the objects. The family of YOLO (You Only Look Once) models are the most popular object detection models for autonomous driving. The YOLOv5 model is a state-of-the-art object detection model that is capable of detecting 80 classes of objects. The model is trained on the MS COCO dataset, containing over 1.2 million images and over 20,000 bounding boxes for the 80 classes of objects.
Multi-object tracking refers to the problem of tracking the objects detected across frames. For this project, we are implementing the *Deep Sort* algorithm for tracking the bounding boxes. Simple Online Realtime Tracking *SORT* is an approach of multi-object tracking using simple and effective algorithms such as Kalman Filter. Including an association metric (using deep learning) for the detected object across frames leads to a more robust and accurate multi-object tracking called Deep Sort.
The problem of trajectory prediction (as is already apparent from the name) involves predicting the agents detected by the YOLOv5 model. PEC Net uses an encoder-decoder architecture for predicting the agents detected by the YOLOv5 model. The encoder is a neural network that encodes the input image into a vector of fixed size. The decoder is a fully connected neural network that decodes the vector into a high-dimensional vector. For the implementation of PEC Net, we need an aerial view of the scene. However, we are working with datasets of camera images taken in the ego-centric view in our implementation. Hence for a complete end-to-end perception pipeline, we need to move the detections of our object detection module to birds-eye view. This is a part of future work.
The code for the project is available at

II. Background and Related Works

A. Object Detection

The problem of object detection and object tracking is quite well known in the field of computer vision. Modern Object detectors are composed of two main components:

- The backbone: This is the part of the detector that is responsible for extracting the features of the objects in the image using a deep convolutional neural network.
- The head: This is the part of the detector that is responsible for classifying the objects in the image.

On the basis of head component, we have two main approaches for object detection:

- Two stage Object Detector: This approach leads to high localization and object detection accuracy of the results.
  Example: R-CNN, fast R-CNN, faster R-CNN, R-FCN, Libra R-CNN, etc
- One stage Object Detector: This approach leads to a higher reference speed.
  Example: SSD, YOLO, RetinaNet, etc

We have chosen the One-stage Object Detector approach since we need a higher reference speed for autonomous driving applications. Specifically, we have used YOLO detection for our application.

B. Multi-Object Tracking

Multi-object tracking methods usually include a tracking network and a detection network. The tracking network is responsible for updating the position and orientation of the object in the scene. The detection network is responsible for detecting the object in the scene. The tracking network is responsible for the data association across the frames.
The detections from the object detector can be processed in one of the following ways:

- Batch method: In this approach, the data association problem is solved by storing the image frames with detections in a batch. The batch is then processed one by one.

- Online method: In this approach, the data association problem is solved by processing an image frame with a detection one at a time.

Practically, the batch method performs better than online methods for the tracking network. However, the batch method is computationally expensive, which means that the tracking network needs to process the entire batch of detections before starting the next frame. Hence, this approach is not viable for our project. Therefore, we have chosen the online method for our project.
Famous object tracking methods include:
KCF tracker, SORT tracker, MOSSE, KCF+SORT, MOSSE+KCF, MOSSE+KCF+SORT, MedianFlow, etc.

For our project, we have chosen the Deep SORT, an online multi-object tracking algorithm that incorporates a deep association metric for the tracking network.

C. Trajectory Prediction

III. Datasets

Autonomous driving is an emerging field; hence there are limited datasets that can be used for training and testing which would resemble the real world. The datasets that we have used for our project are:

- Lyft level 5 Prediction dataset [?]
  This is a self-driving dataset for motion prediction, containing over 1,000 hours of data. This was collected by a fleet of 20 autonomous vehicles along a fixed route in Palo Alto, California, over a four-month period. It consists of 170,000 scenes, where each scene is 25 seconds long and captures the perception output of the self-driving system, which encodes the precise positions and motions of nearby vehicles, cyclists, and pedestrians over time.
- KITTI dataset [?]
  The dataset comprises of Raw (unsynced+unrectified) and processed (synced+rectified) color stereo sequences (0.5 Megapixels, stored in png format), captured and synchronized at 10 Hz. This dataset was only used for testing the perception modules of our project.
  Along with camera sequences, the dataset also contains 3D Velodyne point clouds, 3D GPS/IMU data and 3D object tracklet labels. All of these will be useful for the proposed future works of the project.

IV. Procedure, Experiments and Results

A. Object detection

1) YOLOv3:

V. Limitations

VI. Future work

VII. Acknowledgment