

Lidar and camera based caution object & distance detection for Auto drive

이현도

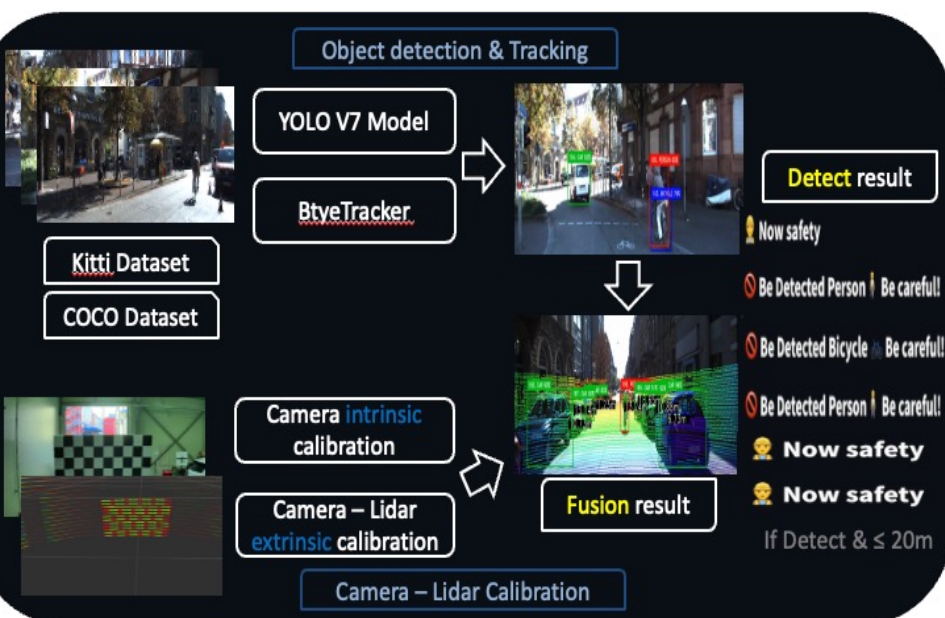
강남대학교 데이터사이언스 전공
E-mail: 201804234@kangnam.ac.kr

Introduction 서론

- 주행 중 주의해야하는 객체 탐지 후 보조
- 실제 주행을 하다 보면, 사람 시각에서 보이지 않는 사람이나 킥보드 이용자, 또는 자전거, 오토바이 이용자와 사고가 나는 사건이 많이 생긴다. 이런 부분을 **camera와 lidar 센서를 통해서** 주의해야하는 객체를 탐지하고 **거리를 계산**하여 해당 객체가 일정 거리내에서 탐지되면 주의하도록 문구를 알리는 서비스 개발을 진행하였다. **Camera를 통해 객체를 탐지**하고 탐지된 객체에 속하는 **lidar point cloud를 통해 거리 값을 얻는다**. 이를 통해 현재 탐지된 객체의 거리를 보여주고 위험 상황을 알려주도록 진행하는 것을 목표로 하였다. 이 과정 속에 필요한 **camera – lidar calibration과 tracking 기술을 접목**하는 소목표를 두고 프로젝트를 진행하였다.

Method 방법론

• Work Flow



✓ Object detection (YOLO v7)

- DataSet
 - * Kitti Dataset <- Auto Drive 관련 opensource dataset ,7418 images
 - * COCO Dataset <- Computer Vision 관련 대표 dataset
- YOLO Model
 - * 1 stage detection model (빠른 판단에 적합한 모델)
 - * Transfer Learning
 - batch-size : 32 , image size : 640 x 640 , Gpu : a600 x 2, epoch :100
 - yolov7_tiny.pt and yolov7_base.pt를 통해 Transfer Learning 진행.

✓ Tracking (ByteTracker)

- Kalman Filter를 활용한 ByteTracker 알고리즘 사용
- Yolo model에서 Box를 가져와, 각 Box를 Tracking하도록 코드 병합

✓ Camera – Lidar Calibration

- 사용할 Camera와 lidar 에서 Checkboard를 활용하여 Metrix 생성.
- $Y(2D) = P \times R0 \times Rlt \times X(3D) \Rightarrow$ lidar 3D to 2D 전환 & image 병합

✓ Show on Wesite (Streamlit)

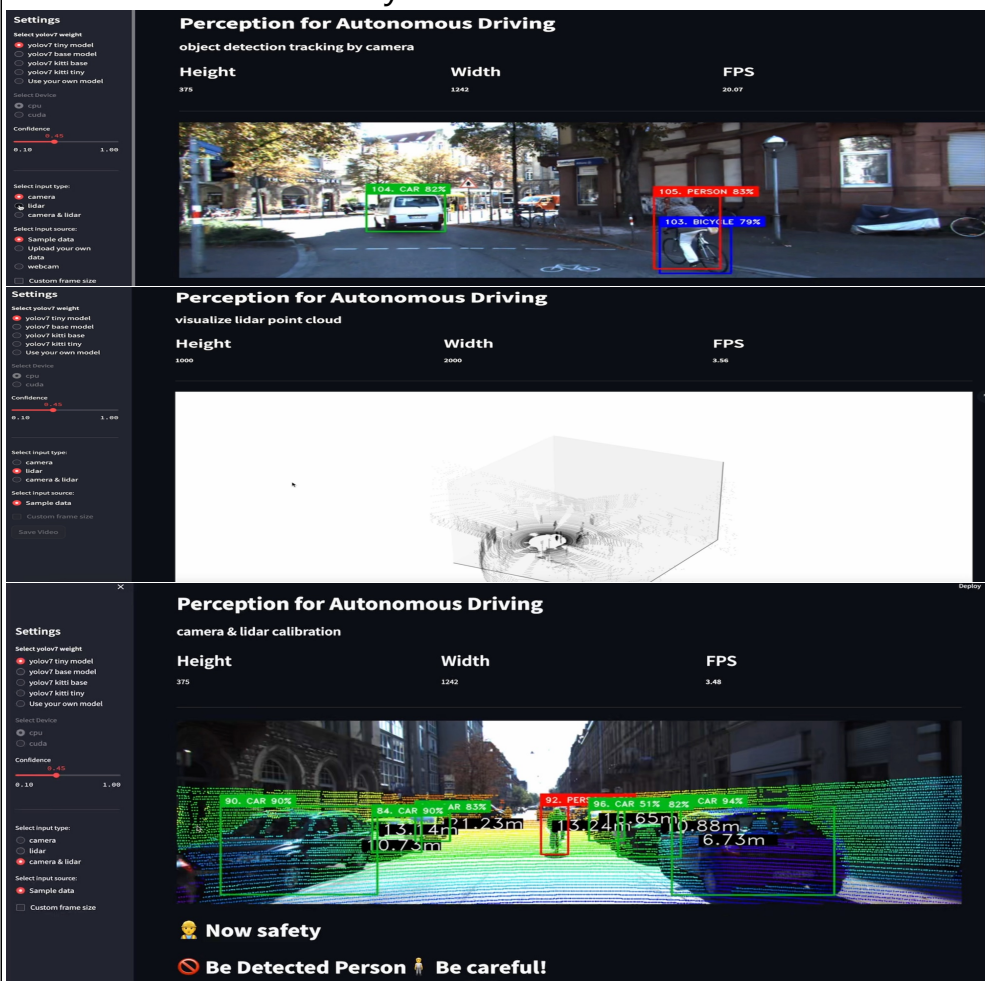
- 웹 기능 소개
 - * lidar visualization
 - * image resize & confidence 조정 가능
 - * Obect Detection : webcam , user video
 - * Object Distance 시각화
 - * Camera – Lidar calibration : **위험 객체 20m이내 탐지 시 문구 안내**

Result 결과

• YOLO V7 Result on Kitti Dataset

Model	mAP @0.5	Precision	Recall	FPS
YOLOv7 base	0.941	0.916	0.95	2.3
YOLOv7 tiny	0.935	0.904	0.945	4.5

• Show on Website by Streamlit



Future Work 추후 개발 계획 및 보완점

“ FPS의 속도 향상을 위해 다른 언어를 사용 계획 ”

→ C++ 또는 TensorRT, onnx와 같은 경량 연산에 특화된 도구 활용 시도

“ Lidar 관련 Tracking 진행 계획 ”

→ Lidar를 활용한 Tracking과 detection 기술 활용 시도.

“ Camera 3D object Detection 계획 ”

→ Model based Distance Detection을 활용하여 3d 객체 탐지 시도.

“ GPU 활용하여 결과 도출 계획 ”

→ CUDA 활용가능한 환경에서 웹 구동 시도. => FPS 향상 예상.

References 참고

- YOLO Streamlit Github 참고 <https://github.com/moaztaha/Yolo-Interface-using-Streamlit/tree/main>
- YOLO & ByteTracker 참고 <https://github.com/theos-ai/easy-yolov7/tree/main>
- Linear Camera Calibration 공부 <https://www.youtube.com/@firstprinciplesofcomputerv3258D>
- Linear Camera Calibration 관련 블로그 <https://gaussian37.github.io/vision-concept-fisheye-camera/>
- ByteTracker 논문 <https://arxiv.org/pdf/2110.06864.pdf>
- Yolo v7 논문 <https://arxiv.org/pdf/2207.02696.pdf>
- Sensor Fusion 공부 <https://www.youtube.com/watch?v=6qV3YfEppuc&list=PLn8PRpmsu08nYoBpEKzoMQveStyS-h4>