

## 자율주행 소프트웨어

➔ 규칙기반 자율주행 SW + 학습기반 자율주행 SW

### - 자율주행 방식

- ➔ 머신러닝 YOLO활용 : 전방 상황을 보고 현재 어떤 미션을 수행해야 할지 파악
- ➔ 강화학습 기반 : 강화학습 모델인 DQN적용
- ➔ End-to-End : 주행영상 + 조향각(핸들각도) 데이터를 학습하여 자율주행
- ➔ Adaptive Cruise Control : 추돌 방지 자율주행 (앞 차와의 간격 유지)
- ➔ PID 제어 : 차량이 흔들거리면서 주행하지 않도록 제어
- ➔ AR Tag 기반의 주차 : AR Tag를 보고 자동차의 자세를 파악하여 주차
- ➔ 앞차 따라가기 주행 : 앞에서 달리는 차량이 지나간 경로를 따라 주행
- ➔ 라이다 기반 복도 주행
- ➔ 고속 코너링 - 드리프트 주행

### - 차선 따라 주행

- ➔ 카메라 영상을 기반으로 차선 인식
- ➔ 차선을 벗어나지 않고 주행
- ➔ OpenCV 기반의 영상처리를 통해 차선 위치 파악
  - color영상 -> Grayscale변환 -> 이진화 변환 -> 차선 찾기
- ➔ 허프변환(Hough Transform)을 이용한 차선 추출
  - Grayscale -> blur -> canny edge -> ROI -> hough transform -> result
- ➔ 머신러닝 모델 적용한 차선 추출
  - Real-Time Semantic Segmentation -> 딥러닝 모델 -> 영상에서 차선 추출

### - 횡단보도 인식

- ➔ OpenCV함수 사용 : 도형 개수 카운트 하는 함수
- ➔ 횡단 보도 : 여러 개의 블록으로 카운팅
- ➔ 정지선 : 한 개의 블록으로 카운팅

#### - 신호등 인식

- ➔ 화면 상단에 ROI영역 설정
- ➔ 연속된 동그라미 찾기
- ➔ YOLO오브젝트 인식
- ➔ 신호등 상태 정보 : 색보다는 밝기와 3개중 위치를 보고 구분

#### - 언덕 구간

- ➔ 차선 왜곡 : 불룩한 항아리 모양
- ➔ 한쪽 차선만 볼 때 : 곡선 차선으로 오해하고 핸들을 꺾어 도로 벗어날 수 있음
- ➔ IMU 센서로 언덕주행 인식 : 양쪽 차선 모두 확인

#### - 장애물

- ➔ 돌발 장애물 : 라이다를 이용하여 전방 장애물 감지

#### - 로터리 진입

- ➔ 장애물 존재여부 확인
  - 좌측에서 다가오는 차량 확인
  - 우측으로 멀어지는 차량 확인

#### - 차선 변경

- ➔ 자동차가 없는 차선으로 진행 : 장애물 감지, 장애물 없는 차선 선택하여 주행

#### - QR 코드

- ➔ 길 옆의 표지판 아래에 있음
  - 앞으로 상황을 QR코드를 읽고 앞으로의 상황을 예상할 수 있음 (차선, 정지선 등)
- ➔ 스스로 알기 위해서는 머신러닝 기반 인지기능 적용 가능