

손상 도로 환경에서의 자율주행 의사결정

캡스톤 디자인 II

지도교수

조민호 교수님

About me

최재혁

과제 개요

과제 선정 배경

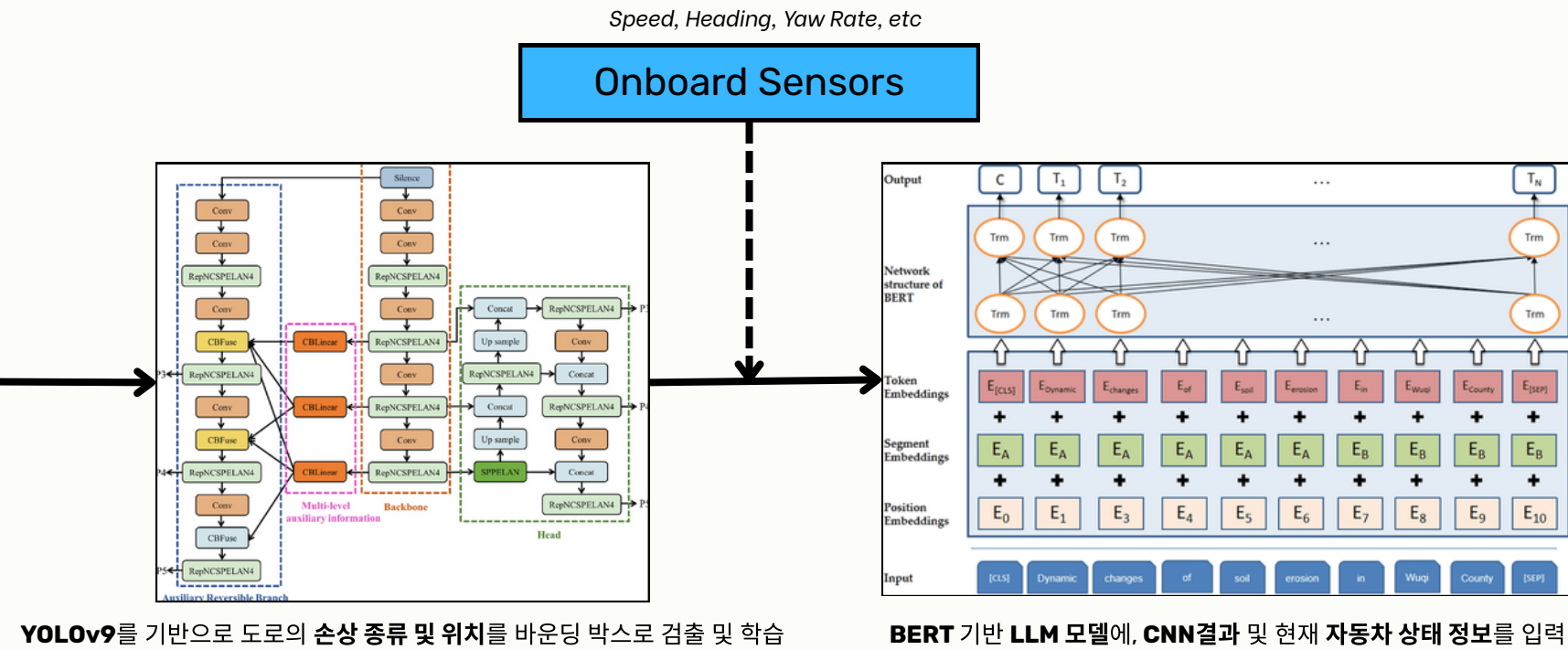
- 최근 자율주행 기술은 눈부신 발전을 이루고 있으며 다양한 실도로 상황에 대응 할 수 있는 알고리즘이 연구되고 있음
- 실도로 환경에서는 포트홀, 균열, 차선 미비, 비포장 구간 등 손상된 요소들이 자주 등장
- 기존의 자율주행 알고리즘은 도로 환경을 전제로 학습된 경우가 많아, 예외적인 손상 상황에 대한 대응 능력이 떨어지고 사고 위험이 증가
- 노후화된 도로, 예측 불가능한 도로 손상(예: 포트홀, 침하, 차선 지워짐 등)은 차량의 센서 오류나 경로 판단 실패를 유발
- 사고 위험 증가와 주행 효율 저하로 이어짐

과제의 필요성

- 자율주행 기술은 상용화 단계에 접어들고 있음에도 불구하고, 현실 도로 환경의 불완전성은 여전히 자율주행의 실효성과 안전성 확보에 큰 장애 요소로 작용
- 기존의 자율주행 시스템은 주로 정형화된 도로 상황에 최적화되어 있어, 예외적 상황에 대한 상황 인지와 대응이 제한적 따라서 다양한 유형의 도로 손상에 유연하게 대응하고, 실시간으로 경로를 재조정할 수 있는 고도화된 의사결정 시스템이 요구됨
- 멀티모달 거대언어모델(LLM)을 통해 다양한 입력 형태(영상, 텍스트, 센서 정보 등)를 융합하여 맥락 기반의 추론을 수행

과제 구성

차량 전방 및 도로의 상태가 드러나는 이미지



YOLOv9를 기반으로 도로의 손상 종류 및 위치를 바운딩 박스로 검출 및 학습

BERT 기반 LLM 모델에, CNN결과 및 현재 자동차 상태 정보를 입력

LLM은 대응 방식을 출력하고, 출력된 대응을 실제 차량에 적용

구성

- 탐지 결과는 온보드 센서의 결과와 함께 판단 모듈로 넘어가며
- 생성된 대응 문장은 차량 시스템에 전달되어, 실제 주행 전략으로 반영
- End-to-End 자율주행 대응 시스템의 초기 구현 기반을 제시

전처리

AI-hub에서 여러 상태의 도로노면의 데이터셋 수집

- AI-hub - 도로장애물/표면 인지 영상
- 도로의 상태 및 장애물, 표면에 대한 데이터
- 라벨링은 다음과 같음 (형식 - json)
- 바운딩 박스 : 도로 위 장애물, 포트홀 등

각 이미지에 대해 도로 손상의 위치와 유형을 태깅

- 현재 과제에서 요구하는 기존 데이터에 없는 손상에 대해서,
- 바운딩 박스로 클래스 ID, 위치를 포함하는 객체로 태깅



탐지

도로 손상은 다음과 같은 특성을 지님 :

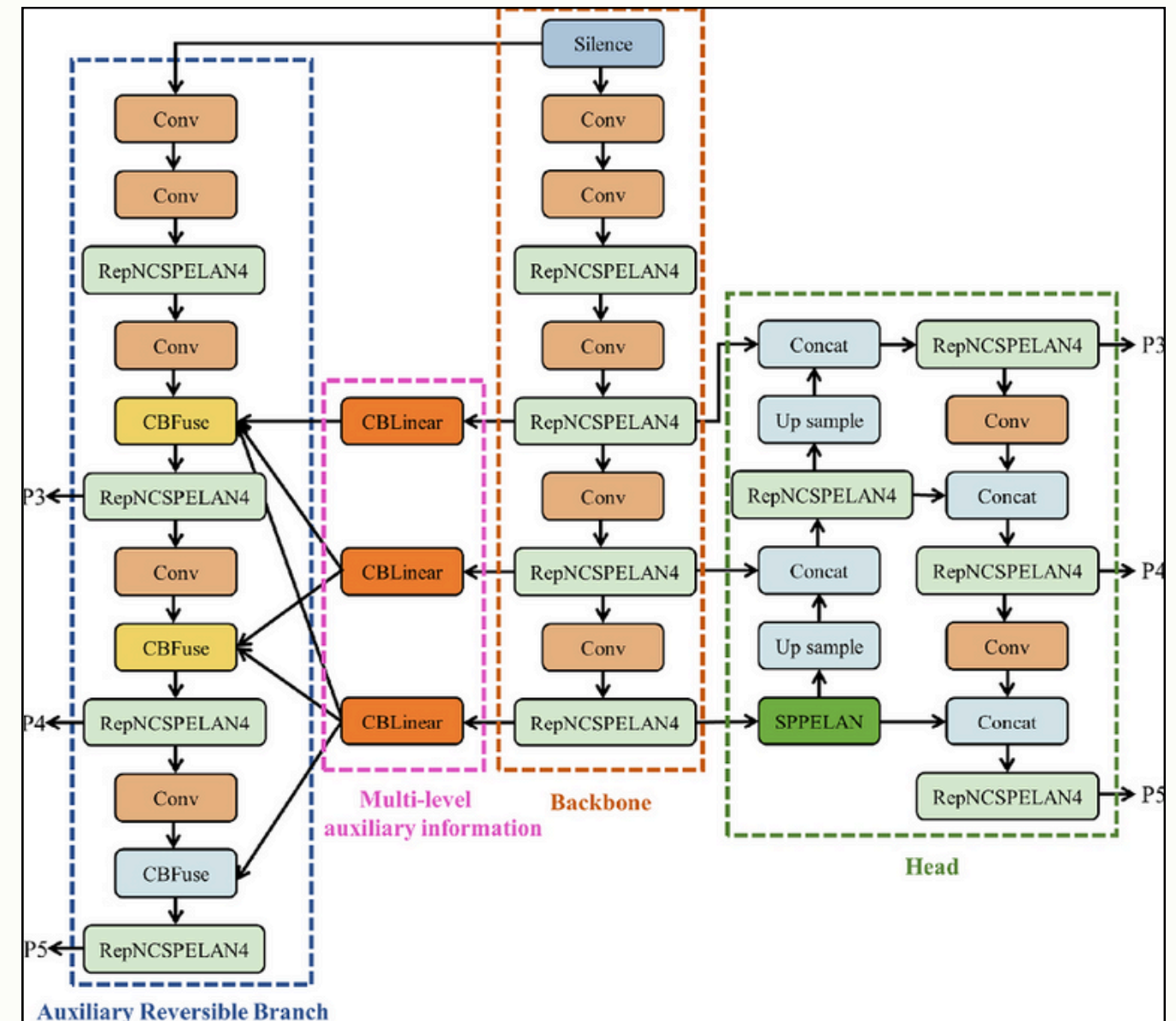
- 크기가 작고 형태가 불규칙적임 (특히 균열)
- 배경과 색상이 유사해 시각적으로 구분이 어려움
- 실시간 탐지가 필수 (운전 중 즉시 반응해야 함)

YOLOv9 기반 손상 도로 탐지 모델 학습

- YOLOv9는 기존 YOLO 계열 모델의 속도 강점은 유지
- 소형 객체 탐지 정확도, 표현력, 구조 유연성이 대폭 향상

학습

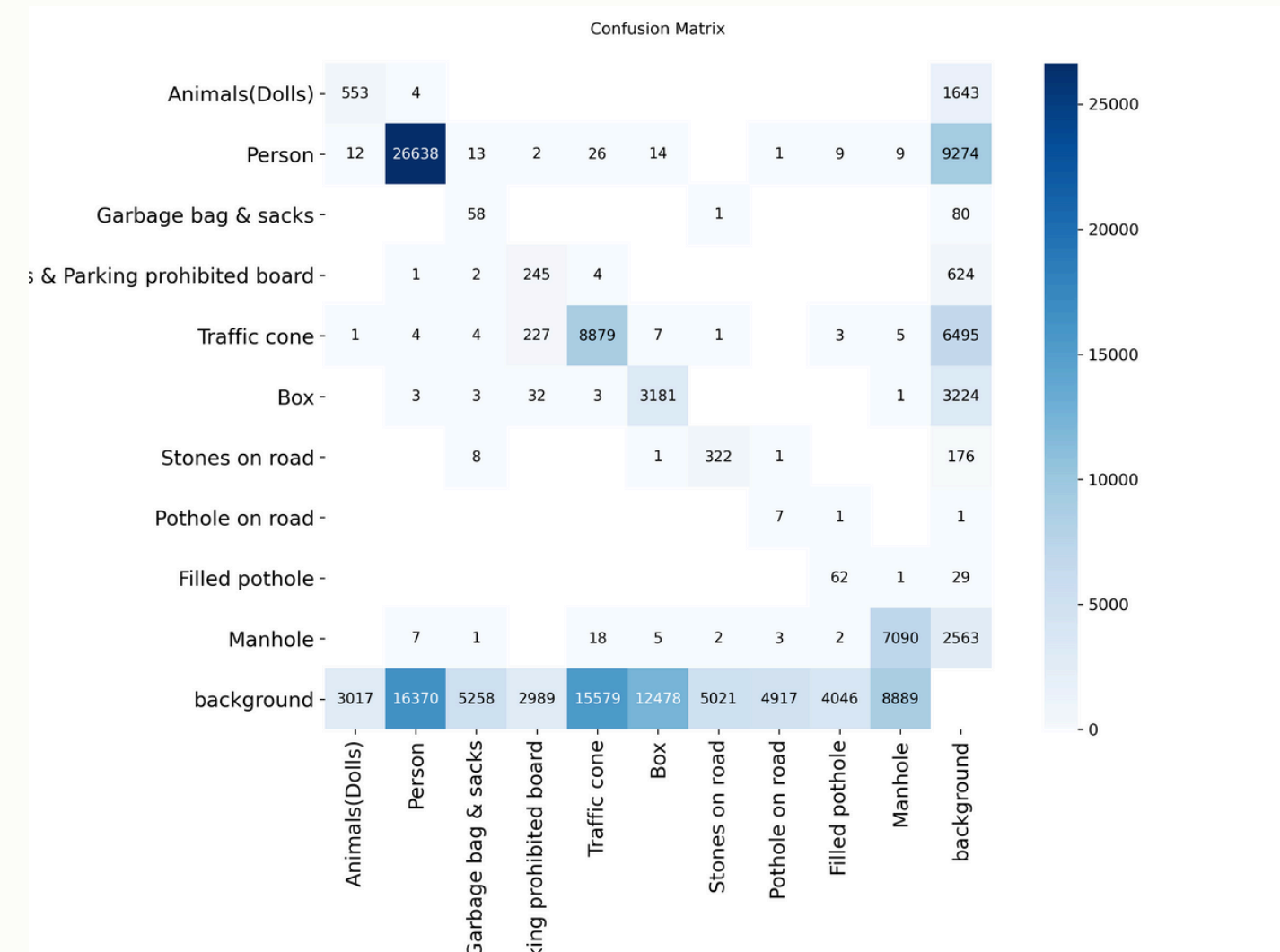
- 인식하고자 하는 클래스는 **총 10개**
- Animal : 동물 / Person : 보행자 / Garbage bag & sacks : 쓰레기 봉투
- Construction signs & Parking prohibited board : 표지판
- Traffic Cone : 트래픽 콘 / Box : 박스 / Stones on road : 도로 위 암석
- Pothole on road : 포트홀 / Filled Pothole : 채워진 포트홀 / Manhole : 맨홀



YOLO 모델 성능 평가

혼동행렬

- 전반적으로 :
 - **background로 흡수되는 비율이 높음** → 검출을 실패함
- 예를 들어, 맨홀(Manhole)의 경우 :
 - 정답 중 상당수가 background로 잘못 분류
 - 일부는 Traffic cone, Box 등과도 혼동이 발생
- Stones on road, Pothole on road, Filled pothole 등 :
 - 도로 손상 계열 클래스들 사이에서도 소량의 상호 오분류 관찰
- YOLO가 객체를 검출하지 못하고 background로 처리하는 경우가 꽤 있었다는 의미 → **Recall 부족** 문제
- 다만 검출만 성공하면 동일 클래스에 맞게 분류되는 비율은 상대적으로 높아 **Precision은 괜찮은 편**

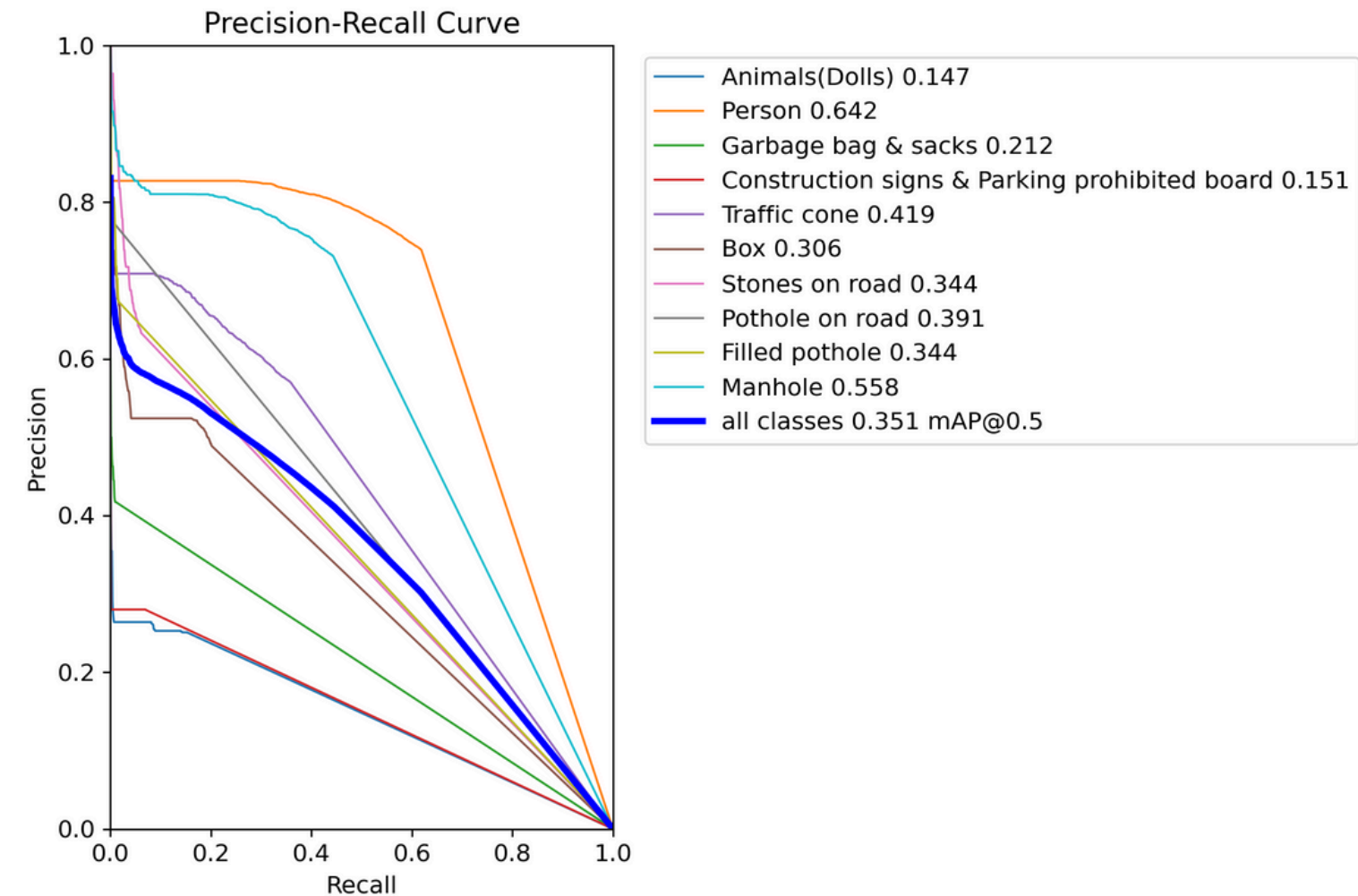


YOLO 모델 성능 평가

Precision-Recall 곡선

- Person, Manhole, Traffic cone, Pothole on road 등 :
 - PR 곡선 아래 면적(AP)이 0.4~0.65 수준으로 상대적으로 높음
 - 실제 주행 영상에서도 이 물체들을 **비교적 높은 확률로 정확히 검출**할 수 있음을 의미함
- Animals(Dolls), Garbage bag & sacks, Construction signs & Parking prohibited board 등 :
 - AP가 0.15~0.25 수준으로 낮아, 이 클래스들은 **검출 성공률이 전반적으로 떨어지는 편**
 - 초반부터 급격히 내려가는 곡선 :
 - Confidence 임계값을 조금만 올려도 Recall이 빠르게 감소하고, 검출을 많이 놓침을 의미함

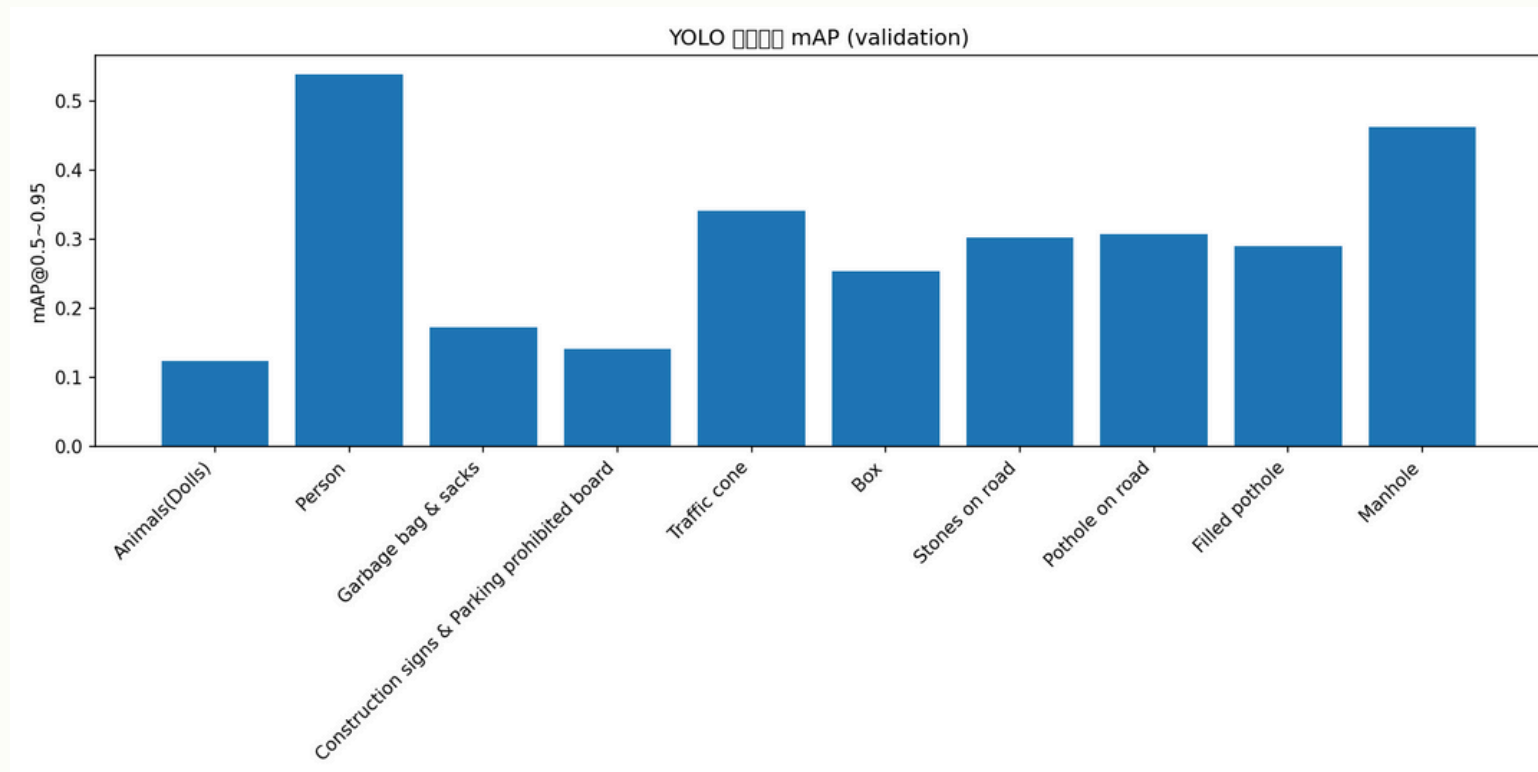
- 형태가 뚜렷하고 도로 위에서 배경과 잘 구분되는 객체는 AP가 높게 나타나며, 실제 시스템에서도 안정적으로 검출될 가능성이 큼
- 크기·형태가 다양하거나 배경과 색·질감이 비슷한 클래스는 AP가 낮게 측정되었고, 이는 실제 주행 환경에서 놓치는 경우가 많을 수 있음을 시사함



YOLO 모델 성능 평가

mAP :

- 전반적인 **mAP@0.5:0.95**는 대략 **0.12~0.53** 범위에 분포
 - 평균적으로 “중간 정도” 정확도의 객체 검출 성능을 보임
- 객체 종류에 따라 성능 편차가 큼
 - Person, Manhole 클래스는 mAP가 0.45 이상으로 비교적 높게 측정됨
 - Animals(Dolls), Garbage bag, Construction signs 등은 mAP가 0.2 이하로 낮게 측정됨



종합적으로 :

- 전체 F1-Confidence 곡선을 보면,
 - confidence threshold가 약 0.2~0.3일 때 F1과 Recall이 모두 상대적으로 높은 구간을 형성
 - 이 구간에서는 탐지 누락(미검출)과 오검출(잘못된 박스) 사이의 균형이 가장 좋음
- 그 이상에서는 Precision은 좋아지지만 Recall이 급감
- 따라서 본 프로젝트에서는
 - 기본 운영 모드에서는 ****threshold ≈ 0.25**** 전후를 기준값으로 사용하여,
 - ‘필요 이상으로 민감하지 않지만, 위험 객체를 놓치지 않는’ 수준의 trade-off를 선택
 - 추후에는 서비스 시나리오에 따라,
 - “안전 우선(보수적 주행)” 모드에서는 threshold를 약간 낮추고
 - “원활한 교통 흐름(공격적 주행)” 모드에서는 threshold를 약간 높이는 전략을 고려

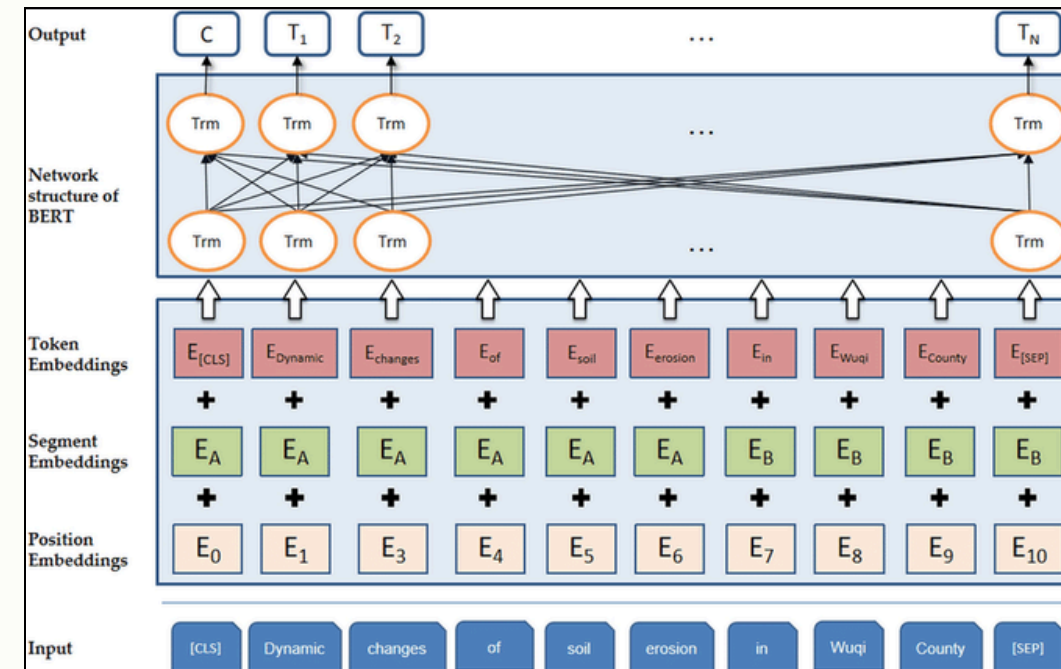
거대언어모델(BERT)을 이용한 자율주행 의사결정 모델

BERT 기반 거대언어모델 장점 및 사용하는 이유

- 단어의 의미를 양방향 문맥을 통해, 정밀하게 해석할 수 있는 Transformer 구조 기반 LLM
- 자율주행에 BERT가 사용될 수 있는 이유는 다음과 같음
 - 복합 입력 해석 :
 - 손상 정보, 속도, 차선, 거리 등 정보를 문맥적으로 통합
 - 자연어 생성 :
 - 상황 판단 결과를 사람이 이해가능한 문장으로 출력 가능
 - 설명 가능한 AI :
 - “왜 그런 판단을 했는가”를 함께 출력 가능
 - Fine-tuning 용이 :
 - 주어진 주행 상황 데이터에 맞춰 쉽게 학습 가능

BERT 기반 거대언어모델 학습

- 입력은 YOLOv9과 세부 손상 분류 모델 에서 나온 결과 및 차량 상태 데이터
- 모델은 json 데이터를 받아 대응 자연어 출력하는 방식으로 Fine-tuning
- Supervised Learning 기반 시나리오 데이터를 수천 건 생성하여 학습
- 응답은 전략 출력 + 판단 이유로 구성됨:
 - 전략 : 감속하며 차선을 유지
 - 판단 이유 : 전방 포트홀이 깊고, 우측 차선이 불안함
- 입력 구조가 정형적이므로 차후 다른 LLM 모델로의 확장도 가능



거대언어모델(BERT)을 이용한 자율주행 의사결정 모델

모델 목표 및 출력

- YOLO로 검출된 도로 위험 객체를 기반으로,
- 현재 프레임에서 차량이 취해야 할 대응 행동을 4가지 중 하나로 분류
 - 감속 : 가속 페달을 떼고 서서히 속도를 줄이는 상황
 - 정지 : 정지선·보행자 등으로 인해 완전 정지가 필요한 상황
 - 우측으로 회피 : 위험 물체가 차선의 좌측에 치우친 경우
 - 좌측으로 회피 : 위험 물체가 차선의 우측에 치우친 경우

입력 특징

- 한 프레임의 bbox들을 이미지 전체에 대한 통계량으로 요약한 뒤, 한국어 문장으로 변환하여 BERT에 입력
 - 이미지 해상도: $W \times H$
 - bbox 개수: n
 - 이미지 하단부에 위치한 객체 비율 : low_frac
 - 좌측/우측 영역에 위치한 객체 비율 : $left_frac, right_frac$
 - bbox의 평균 폭, 평균 높이 : avg_w, avg_h (이미지 크기로 정규화)
 - 임의의 차량 주행 속도 : spd

라벨 생성 방식

- 실제 운전 로그가 아니라 bbox 통계에 기반한 규칙(heuristic),
 - “이 프레임이면 감속/정지/회피를 해야 한다”는 약한 라벨(weak label) 생성
- 상황 대응 :
 - 하단부에 크고 넓은 객체가 많으면 → 정지
 - 하단부에 중간 크기 객체가 있으면 → 감속
 - 좌측에 위험이 치우치면 → 우측으로 회피
 - 우측에 치우치면 → 좌측으로 회피
- 따라서 차량 상태(행동 라벨)와 도로 상태(bbox 통계)가 한 쌍으로 묶인 데이터를 만들어 BERT를 학습시켰음

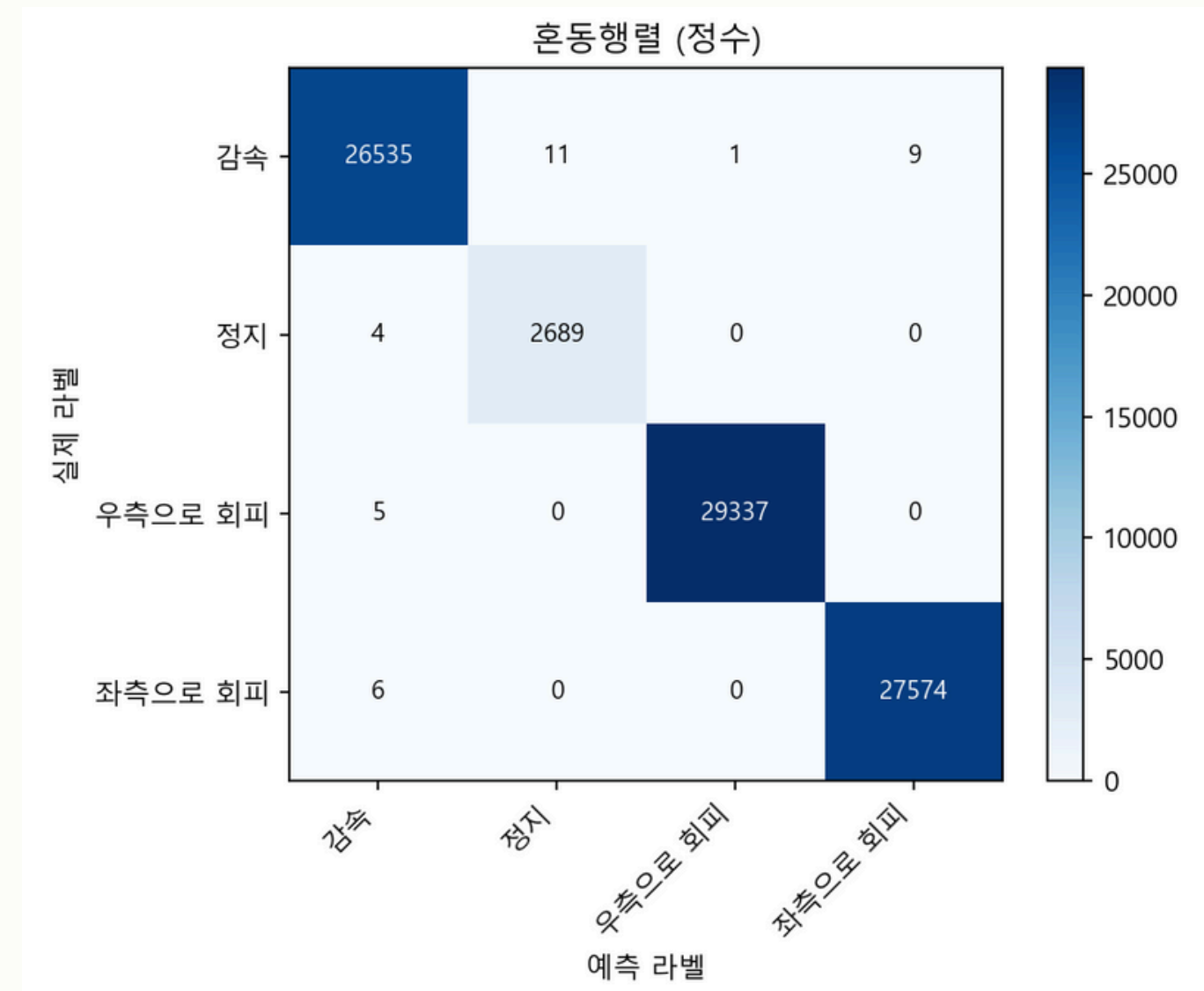
BERT 모델 성능 평가

혼동 행렬

- 네 개 행동 모두에서 대부분의 샘플이 대각선에 몰려 있음
 - 예: 감속은 거의 전부 감속으로, 정지는 정지로, 좌·우 회피도 각각 자기 클래스로 예측
- 비대각선(오분류)에 위치한 값은 극소수에 불과
- 감속 ↔ 정지, 좌측 회피 ↔ 우측 회피 사이에서도 거의 혼동이 발생하지 않음

해석

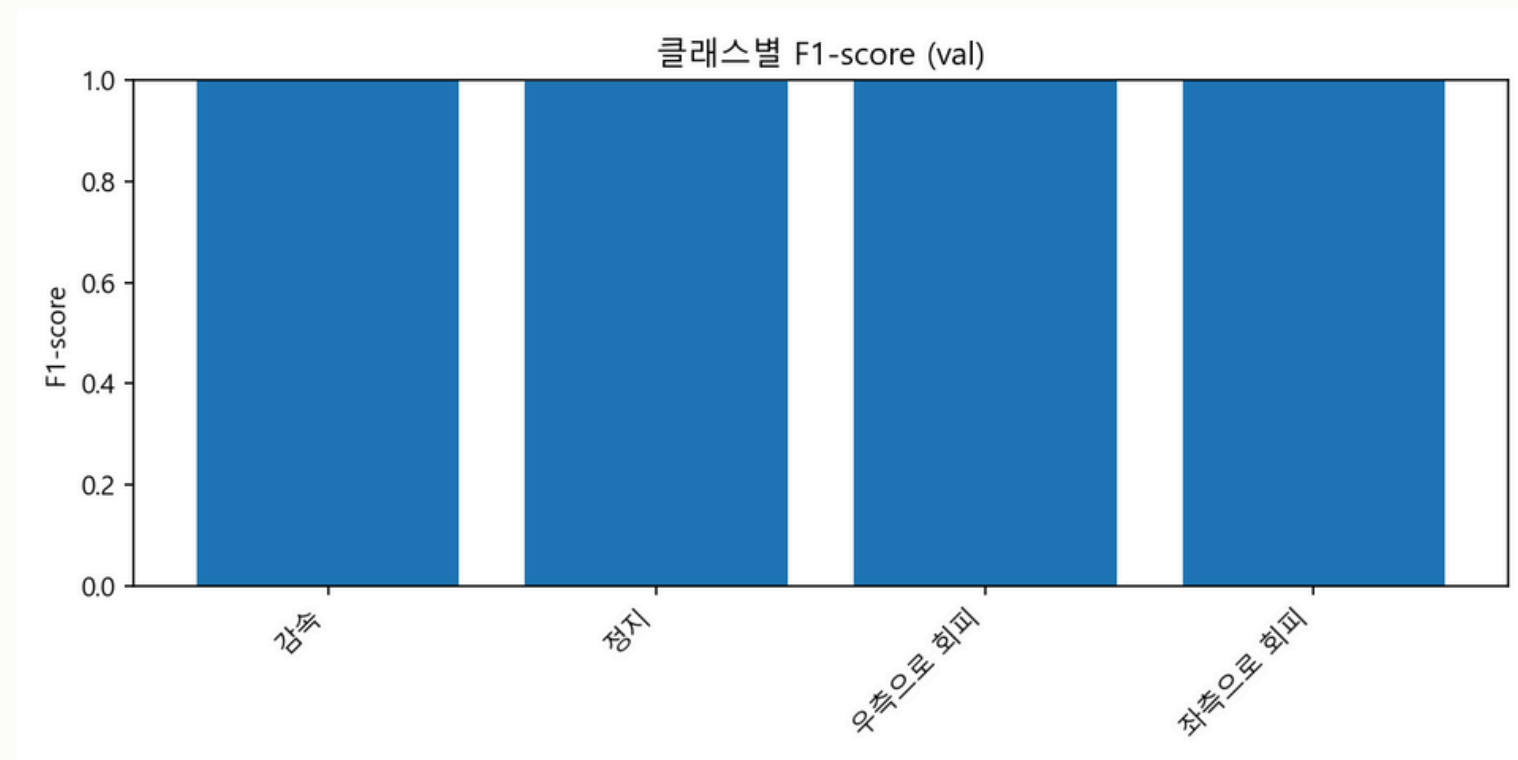
- 주어진 데이터 분포 안에서는
 - BERT가 “bbox 통계 → 행동 라벨” 매핑을 거의 완벽하게 학습했다는 의미
- 즉, 같은 도로 상태 통계가 들어오면 항상 일관된 행동(감속/정지/회피)을 선택함
- YOLO 검출 결과만 안정적으로 들어온다면,
 - 전체 파이프라인에서 의사결정 단계(BERT)에서 발생하는 추가 오차는 매우 적음



BERT 모델 성능 평가

클래스별 F1-score :

- 검증용 데이터에서 4개 행동 클래스 모두 F1-score \approx 0.99 이상으로 매우 높게 나타남
- 이는 :
 - 라벨이 규칙 기반으로 생성되어 일관성이 매우 높고,
 - 입력 특징(하단 비율, 좌/우 비율 등)이 해당 규칙과 직접적으로 연결되어 있기 때문



종합적으로 :

- 전체 정확도는 ****99.96%****로, 주어진 통계 기반 데이터 분포 안에서는
 - “도로 상태 통계 \rightarrow 차량 행동” 매핑을 거의 완벽히 모사
- YOLO 검출 결과만 안정적으로 들어온다면,
 - BERT 단계에서의 추가적인 오차는 거의 발생하지 않음

한계 :

- 라벨이 실제 운전자의 페달/조향 로그가 아니라, bbox 통계 규칙에서 만들어졌기 때문에,
 - BERT가 학습한 것은 “운전자의 진짜 의사 결정”이라기보다
 - 연구자가 정의한 휴리스틱을 근사한 함수에 가까움
- 따라서,
 - 실제 차량 테스트나 복잡한 도로 시나리오에 그대로 일반화된다고 보기 어렵고
- 향후에는 실제 차량 상태 데이터(속도, 조향각, ACC 상태 등)와 도로 객체 정보가 함께 포함된 로그를 이용해 재학습하는 것이 필요함
-

활용방안 및 기대효과

활용방안

- 자율주행 시스템 내 의사결정 보조 모듈로 활용
- 도로 유지보수 및 사고 예방 시스템에 연계 가능
- 다양한 멀티모달 입력을 처리할 수 있는 범용 AI 플랫폼으로 확장 가능

기대효과

- 실도로 환경에서의 실시간 주행 테스트 확대
- 모델 경량화 및 실시간 적용 최적화
- 다양한 도로 상황에 대한 범용성 향상
- 사용자 피드백을 반영한 판단 설명 시스템 고도화

감사합니다

학과

컴퓨터융합소프트웨어

About me

최재혁