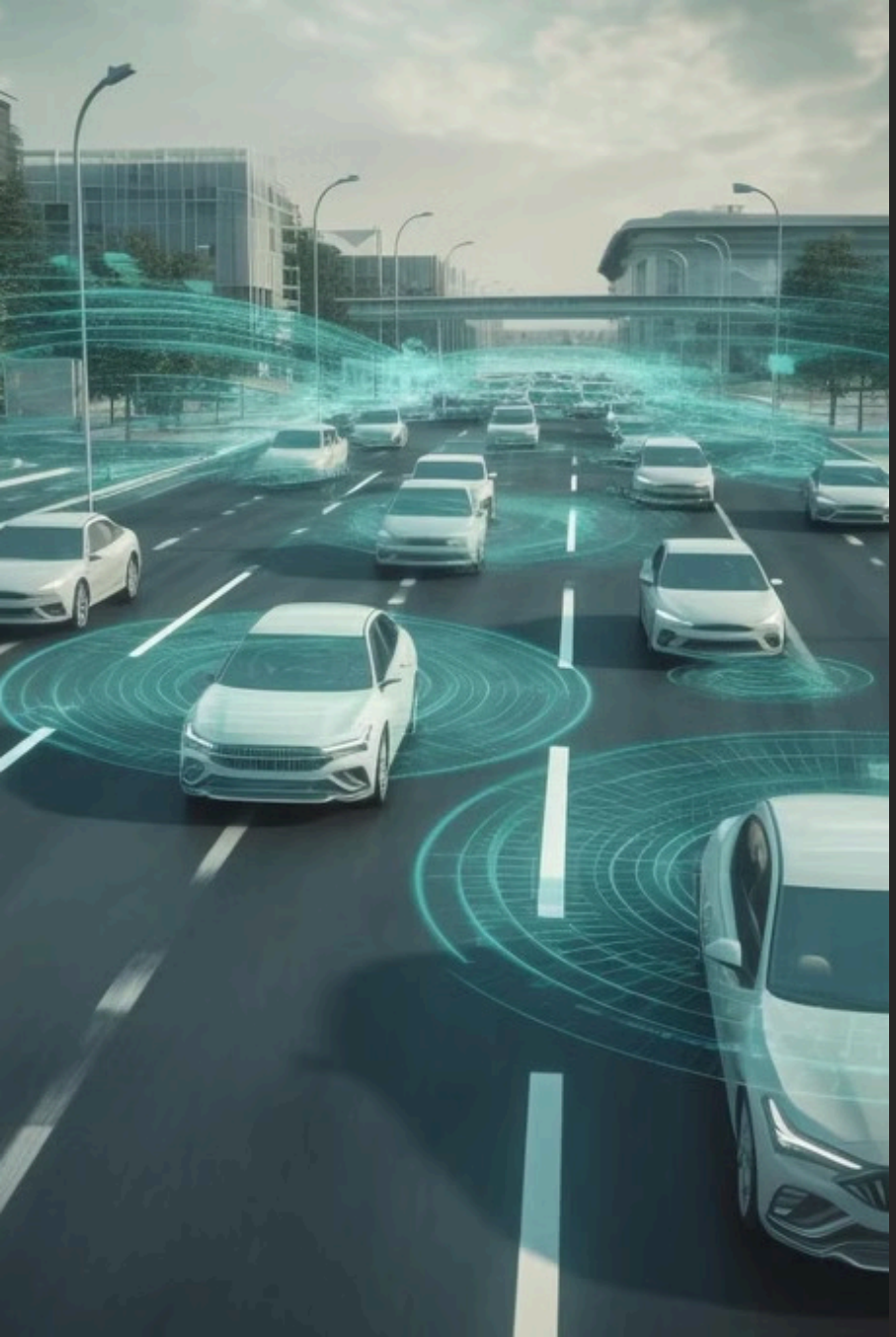


자율주행

자율주행 차량의 핵심 기술인 차선 인식과 ROS를 소개합니다.

안전하고 효율적인 자율주행을 위한 기술적 기반을 탐구합니다.

차선 인식 프로젝트 개요

1 **프로젝트 목표**

실시간 차선 인식으로 안전한 주행 경로 파악

2 **사용 도구**

ROS, Python, YOLO, OpenCV 활용

3 **컴퓨터 비전**

자율주행 차량의 '눈' 역할 수행

4 **기대 효과**

정확하고 신뢰성 있는 차선 인식 시스템 구현

차선 인식을 위한 센서

카메라

고해상도 이미지 및 비디오 수집

다양한 각도에서 도로 상황 캡처

LIDAR

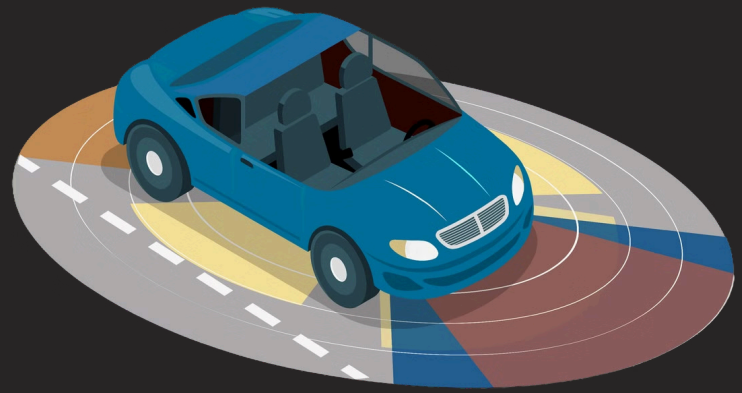
정밀한 3D 깊이 데이터 수집

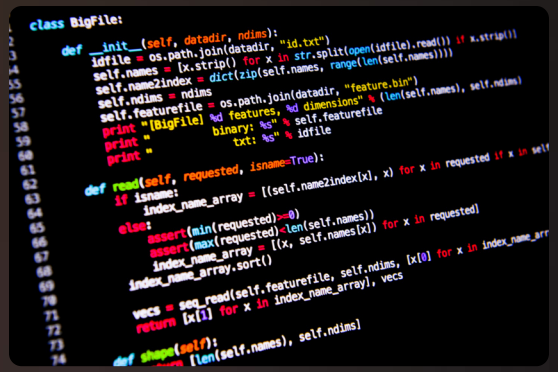
야간 및 악천후 조건에서 유용

데이터 처리

ROS와 OpenCV로 센서 데이터 통합 및 분석

실시간 차선 정보 추출 및 해석





차선 인식을 위한 Python 라이브러리

OpenCV

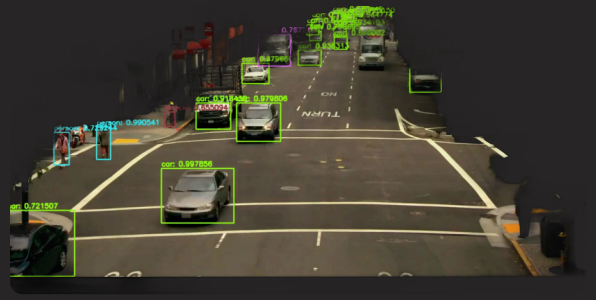
이미지 처리 및 컴퓨터 비전 작업 수행

NumPy

고성능 수치 계산 및 배열 처리

Python 기본 문법

효율적인 코드 구조화 및 로직 구현



YOLO 알고리즘 개요

실시간 객체 인식

빠르고 정확한 차선 및 도로 요소 감지

단일 네트워크 처리

이미지 전체를 한 번에 분석하여 효율성 향상

차선 객체화

차선을 개별 객체로 인식하여 정밀한 추적

다양한 환경 적응

복잡한 도로 상황에서도 안정적인 성능 발휘

차선 감지 워크플로우

1

이미지 입력

고품질 카메라로 실시간 도로 이미지 획득

2

전처리

노이즈 제거 및 그레이스케일 변환으로 이미지 최적화

3

엣지 감지

Canny Edge Detection으로 차선 윤곽 추출

4

YOLO 적용

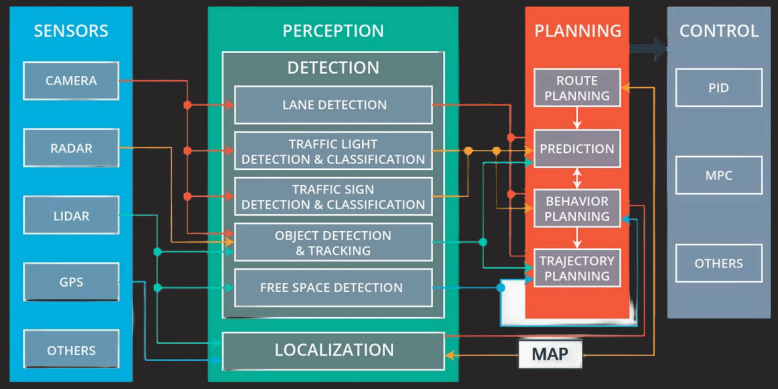
차선 객체 인식 및 분류

5

결과 출력

인식된 차선 정보를 실시간으로 시각화

ROS와의 통합



1

ROS 노드 설계

차선 인식 알고리즘을 독립 노드로 구현

2

데이터 통신

퍼블리셔/서브스크라이버 모델로 실시간 정보 교환

3

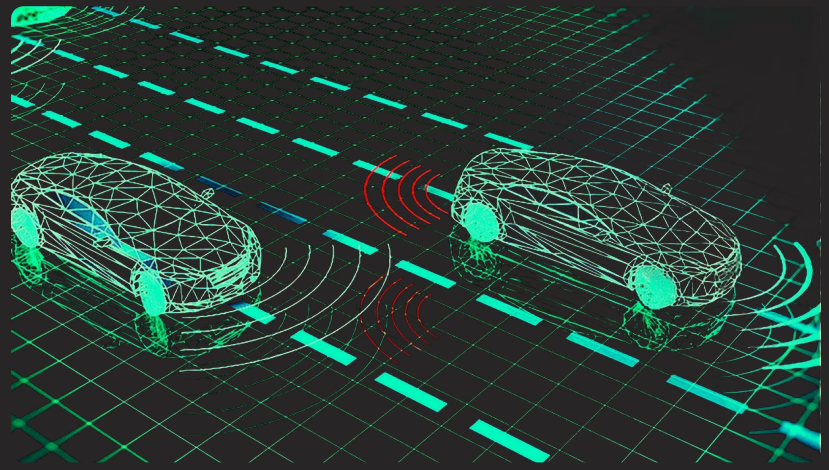
시스템 통합

ROS 프레임워크와 Python 코드 연동

4

성능 최적화

ROS 파라미터 조정으로 처리 속도 향상



테스트 및 평가 전략

테스트 환경	평가 기준	도구
도심	정확도	ROS 시뮬레이터 (Gazebo)
야간	속도	실제 도로 테스트
악천후	안정성	데이터 로깅 시스템

도전 과제와 향후 개선 사항

현재 도전 과제

- 극단적인 조명 변화 대응
- 악천후 시 인식 정확도 유지
- 복잡한 도로 환경에서의 차선 구분

개선 방향

- 다중 센서 데이터 융합 기술 개발
- AI 기반 적응형 알고리즘 구현
- 엣지 컴퓨팅 도입으로 처리 속도 향상

미래 연구 계획

- 딥러닝 모델 고도화
- 실시간 학습 시스템 개발
- 차량간 통신을 통한 협력적 인식

결론 및 다음 단계

1 **차선 인식의 중요성**

자율주행의 안전성과 효율성 확보를 위한 핵심 기술

2 **기술적 성과**

ROS, YOLO, OpenCV 통합으로 고성능 시스템 구현

3 **향후 계획**

실도로 테스트 확대 및 다양한 환경에서의 성능 검증

4 **지속적인 발전**

최신 AI 기술 적용으로 시스템 지능화 추진