

# LG U+ Why Not SW Camp

## 모델 정의서

객체 탐지 기반 자율주행 차량 로봇의 포트홀 감지 및 자동 신고 서비스

# Dynamic Object Detection(D오디)

팀원 : 김승엽, 한은선, 김동혁, 김준희, 박장원, 김나현

---

## 목 차

### 1. 프로젝트 개요

- 1.1 프로젝트 주제 및 배경
- 1.2 프로젝트 목적

### 2. 모델 선정

- 2.1 모델 선정 이유

### 3. 데이터 정의 및 처리

- 3.1 데이터셋 구성
- 3.2 데이터 전처리
- 3.3 데이터 라벨링

### 4. 모델 구조 및 기술 사양

- 4.1 네트워크 구조
- 4.2 기술적 사양

### 5. 모델 학습 및 검증

- 5.1 학습 과정
- 5.2 평가 지표

### 6. 모델 학습 결과

### 7. 사용 기술 및 라이브러리

### 8. 향후 계획 및 개선안

### 9. 부록

## 1. 프로젝트 개요

### 1.1 프로젝트 주제 및 배경

- 주제 : 객체 인식 및 아두이노 기반 로봇 자율주행
- 부제 : 객체 탐지 기반 자율주행 차량 로봇의 포트홀 감지 및 자동 신고 서비스
- 프로젝트 의의 : 포트홀 탐지와 자동 신고 시스템을 통해 교통 체증 감소와 도로 관리 효율성을 향상시키는 것을 목표로 함. 객체 인식 기술과 아두이노 기반의 자율주행 로봇을 활용하여 비용 효율적인 해결책을 제시
- 배경 : 객체 탐지 방식 선정 근거



- 일반적으로 포트홀 탐지 방식은 △차량이 겪는 진동 기반 탐지 △도로 노면을 향해 조사된 레이저 계측 기반 탐지 △영상 인식 기반 탐지로 구분할 수 있음...
- 영상 인식 기반 탐지 방식은 합리적인 비용으로 넓은 영역의 노면 상태를 분석할 수 있으며, 최근 인공지능심층신경망을 활용한 객체 인식 기술의 발달로 더욱 주목받고 있음
- 특히 영상 인식 기반의 탐지 방식은 차량 블랙박스나 휴대전화에 탑재된 카메라와 같은 개인 촬영 장비로도 탐지할 수 있어 육안 조사에 의존하고 있는 지자체에 기술을 보급하기 용이하며, 탐지 결과가 사진으로 저장되기 때문에 관련 전문가가 아닌 일반인도 직관적으로 파손 정보를 파악할 수 있다는 장점이 있음
- 이미지 기반의 탐지가 날씨 등 환경의 영향을 많이 받는다는 한계점은 추후 라이다 탐지와 연동하여 보완 가능 (비용 및 시간적 한계)

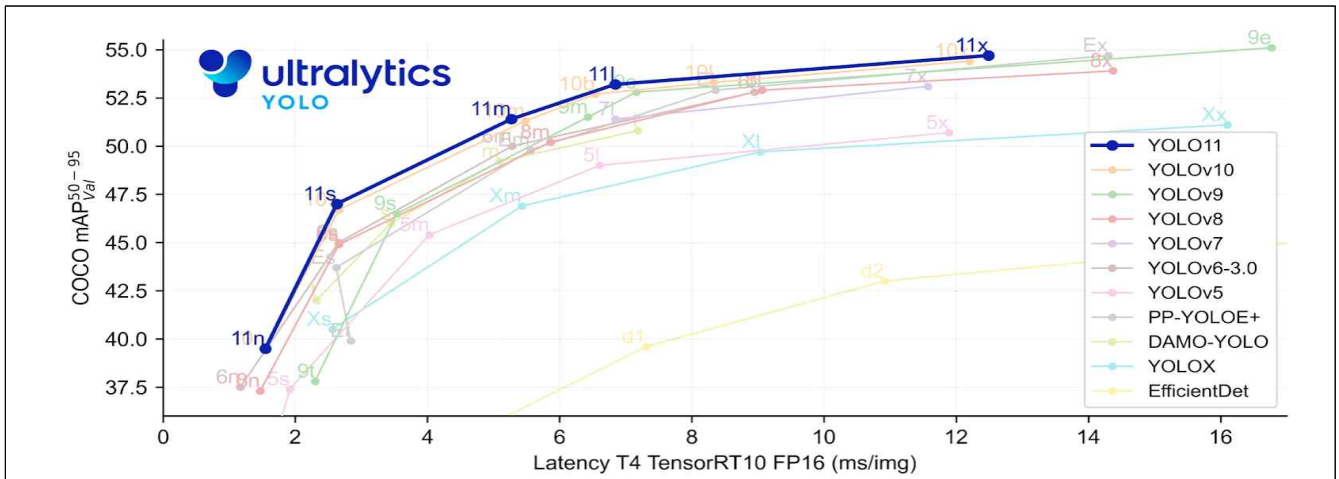
### 1.2 프로젝트 목적

- 프로젝트 객체 리스트
  - 노선 : 강의실에 임의로 설치한 노선을 따라 주행
  - 포트홀 : 저속 주行的 원인
  - 표지판 : 좌회전 및 우회전 등 객체 탐지 기반 자율주행 제어
- 목적
  - YOLO를 이용한 객체 탐지 후 작업 진행 :  
YOLO를 활용하여 다양한 객체를 실시간으로 탐지하고, 해당 객체에 맞는 작업을 진행함. 포트홀, 표지판 등 다양한 도로 환경을 인식하여 자율주행 차량의 행동을 제어할 수 있음
  - 포트홀 발견 시 이미지 및 위치 데이터 웹 서버에 기록 :  
포트홀이 발견되면, YOLO AI 모델이 해당 객체를 인식하여 관련된 이미지와 포트홀 위치 데이터를 실시간으로 웹 서버에 기록함. 이 데이터를 바탕으로 포트홀 위치를 추적하고 관리할 수 있음
  - 포트홀 발견 시 자동 신고 서비스 :  
포트홀을 발견하면, YOLO 분석을 통해 인식된 이미지와 위치 데이터를 웹 서버에 저장하고, 이를 관련 기관이나 여러 기업에 자동으로 신고함. 이 서비스는 포트홀을 신속하게 관리하고, 교통 안전을 향상시키는 데 기여할 수 있음

## 2. 모델 선정

### 2.1 모델 선정 이유

- 초기에는 YOLOv5의 안정성과 검증된 성능을 고려하여 모델을 구성했으나, 프로젝트 진행 중 더 높은 정확도(mAP)와 빠른 응답 시간이 요구됨에 따라 YOLOv11으로 변경 YOLOv11은 최신 기술을 반영해 더 높은 정확도와 낮은 지연 시간을 제공



성능 지표	YOLOv5n	YOLOv11n	개선 사항
mAP (val 50-95)	34.3	39.5	+5.2 (15.2% 향상)
추론 속도 (CPU ONNX, ms)	73.6	56.1	17.5ms 빨라짐 (23.8% 향상)
파라미터 수 (M)	2.6	2.6	동일
FLOPs (B)	7.7	6.5	1.2B 감소 (15.6% 효율 향상)

## 3. 데이터 정의 및 처리

### 3.1 데이터셋 정의

- 학습 대상 : 포트홀, 표지판(왼쪽, 오른쪽, 정지), 차선
- 데이터 출처 : GITHUB, AIHUB, KAGGLE
- 크기 : 640x640 픽셀(YOLO 문서 참조)
- 초기 데이터 용량 : train 160장, validation 20장, test 20장
- 라벨 형식 : 이미지에 해당하는 좌표값 이용

### 3.2 데이터 전처리

- 데이터 리사이징 작업
- 이미지 데이터 크기 : 640x640 픽셀
- 바운딩 박스 크기 : 최소 10x10 픽셀 이상

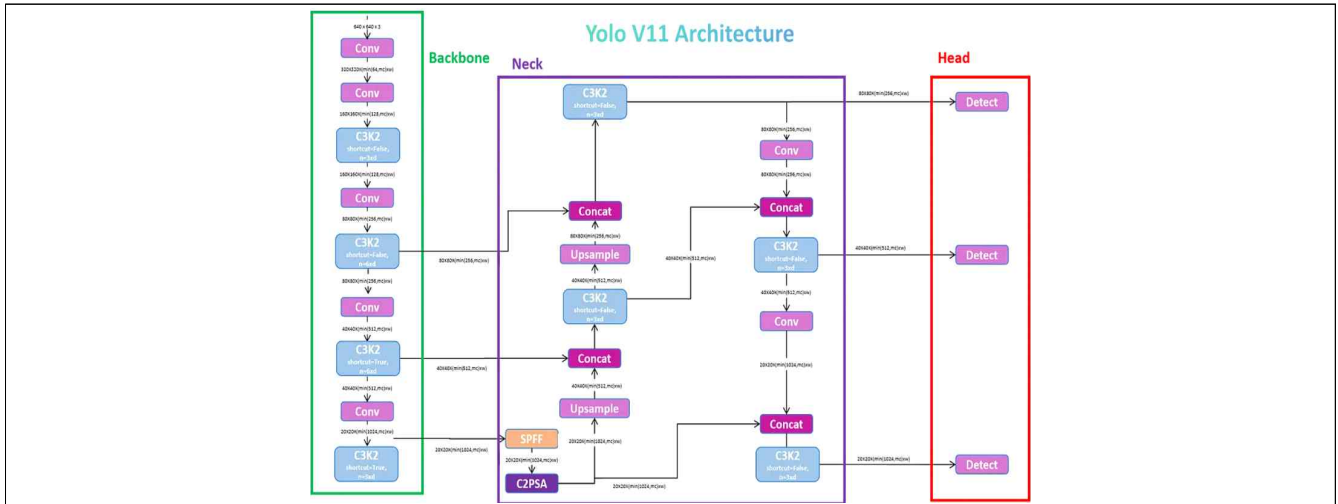
### 3.3 데이터 라벨링

- 이미지 라벨링 개수
- 초기 데이터 셋은 학습용, 검증용, 테스트용으로 학습 80%, 10%, 10%으로 나눔
- 학습 초기에는 클래스당 200장을 준비함(학습 : 160장, 검증 : 20장, 테스트 : 20장)
- 검증 데이터에서 mAP 70% 미만일 경우, 데이터 증강을 활용해 데이터 증가

## 4. 모델 구조 및 기술 사양

### 4.1 네트워크 구조

- 백본과 탐지 헤드의 주요 레이어 비교



구성요소	백본(Backbone)	탐지 헤드(Detection Head)
주요레이어	CSPNet 기반의 CSPDarknet	FPN (Feature Pyramid Network)
핵심 구조	잔차 블록(Residual Blocks), 합성곱 레이어	PAN (Path Aggregation Network)
기능	이미지에서 특징 추출	추출된 특징을 바탕으로 바운딩 박스 및 클래스 예측

### 4.2 기술적 사양

항목	YOLOv11s
입력 크기	640x640
활성화 함수	SiLU (Sigmoid Linear Unit)
Optimizer	AdamW 또는 SGD
Loss 함수	Objectness Loss (BCE), Localization Loss (Ciou), Classification Loss (Cross-Entropy or BCE)
Backbone	CSPDarknet53-Lite
Neck	PANet
Head	YOLO Detect Head

## 5. 모델 학습 및 검증

### 5.1 학습 과정

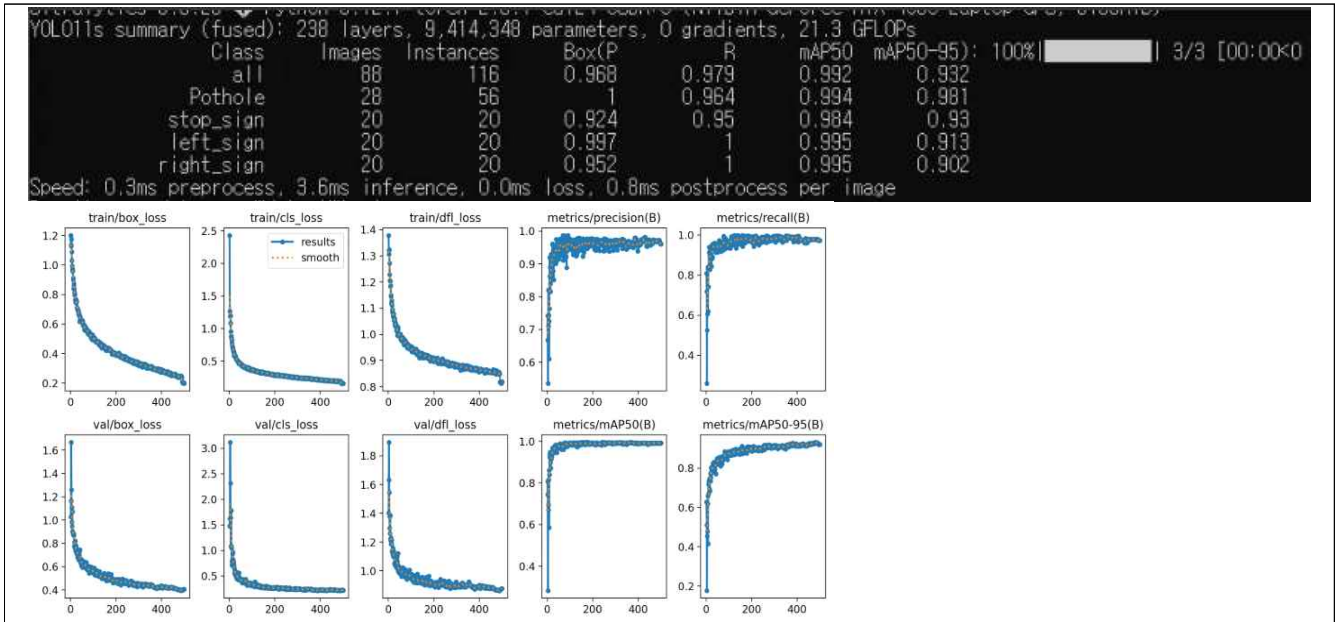
- Epoch 수 : 100번
- 학습 프레임워크 (PyTorch, opencv, labeling, ultralytics, Tensorboard)

## 5.2 평가 지표

- mAP(Mean Average Precision) : Mean Average Precision은 객체 탐지 모델의 성능을 종합적으로 나타내는 지표. 여러 클래스에서 평균적으로 얼마나 정확히 탐지했는지 평가함
- Precision(정밀도) : 예측한 객체 중 실제로 정답인 객체의 비율. FP(오탐지)를 최소화하는 데 중점을 둠
- Recall(재현율) : 실제 객체 중 모델이 올바르게 탐지한 객체의 비율. FN(탐지 누락)을 최소화하는 데 중점을 둠

## 6. 모델 학습 결과

- 모델 성능 mAP: 0.99



## 7. 사용 기술 및 라이브러리

- 기술 : YOLOv11, PyTorch, CUDA 등
- 라이브러리 : NumPy, OpenCV, Tensorboard 등

## 8. 향후 계획 및 개선안

- 추가 데이터 확보 및 학습
- 모델 경량화
- 배포 및 통합 테스트 계획

## 9. 부록

- Backbone :
  - CSPNet (Cross Stage Partial Network) 기반으로 효율성을 높임
- Neck :
  - PANet (Path Aggregation Network): 고해상도 및 저해상도 특징 맵을 결합하여 성능 향상
- Head :
  - YOLOv11S는 YOLOv4부터 유지된 Detect Head 구조를 기반으로 함
  - Anchor-Free로 전환하여 계산량을 줄이고, 더 작은 모델 크기를 유지