LG U+ Why Not SW Camp

성능평가 결과 기획서

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. 프로젝트개요

1.1 목적 및 목표

목적

- 실시간 객체 탐지 정확도를 향상시키고 자율 주행에서 도로, 왼쪽, 오른쪽, 정지, 포트홀의 탐지를 최적화 하기 위함.

목표

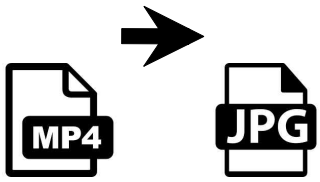
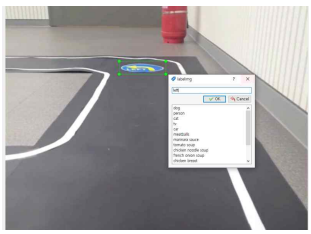
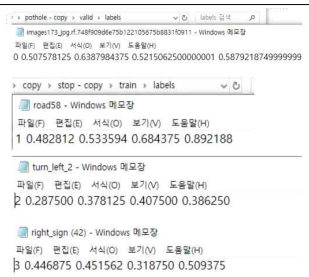
- 최적화된 학습 과정을 통해 검증 데이터셋에서 mAP 50-95(B)가 0.6 이상, Precision(B) 0.8 이상, Recall(B) 0.84 이상을 달성

1.2 데이터

- 직접 촬영한 데이터 셋
- 사용된 데이터셋(훈련,검증,테스트데이터셋)

데이터 종류	설명
훈련 데이터	• 모델 학습에 사용된 데이터셋으로, 손실 값(box loss, class loss, dfl loss)의 감소를 통해 모델의 학습 상태를 평가 • 객체 탐지를 위한 바운딩 박스 정보와 클래스 라벨을 포함하며, 데이터 증강 기법(회전, 크롭, 밝기 조정 등)을 적용하여 다양한 학습 제공
검증 데이터	• 모델의 성능을 중간평가 • 검증 손실 값(box loss, class loss 등)과 mAP(50) 및 mAP(50-95) 지표를 기준으로 모델의 탐지 성능을 평가 • 과적합 여부를 확인
테스트 데이터	• 모델의 최종 성능을 평가하기 위해 사용되는 독립적인 데이터셋 • 모델의 실제 예측 가능성을 평가함.

1.3 데이터 전처리

데이터 전처리 과정			
이미지화	리사이징	바운딩박스	class 통일
			
• 동영상을 프레임 단위로 이미지 생성	• 이미지 사이즈 통일화	• 특정 객체 라벨링	• 클래스 번호를 통한 객체 분류

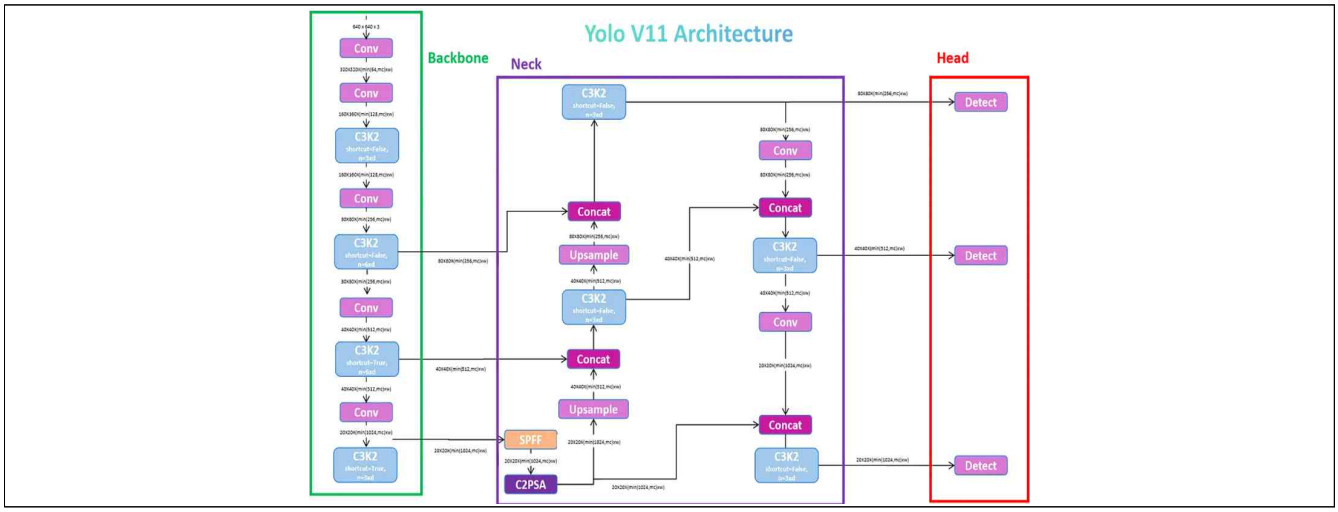
1.4 데이터 분포 및 크기 (현재)

객체		이미지, 라벨링 개수
Pothole	test	54
	train	3003
	valid	28
Stop Sign	test	20
	train	160
	valid	20
Left sign	test	12
	train	516
	valid	47
Right sign	test	20
	train	539
	valid	51

2. 모델 구성

2.1 모델 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

최종 예측 메커니즘으로, 정제된 특징 맵을 바탕으로 객체의 위치와 분류를 출력합니다.
이 단계에서 최종 결과를 생성합니다.

2.2 하이퍼 파라미터 (yolo표준)

이름	현재 값
학습률 (Learning Rate)	lr0: 0.01 lrf: 0.01 warmup_bias_lr: 0.1 fliplr: 0.5
배치 크기 (Batch Size)	Batch Size: 16
에포크 수 (Number of Epochs)	Epochs: 80
모멘텀 (Momentum)	momentum: 0.937 warmup_momentum: 0.8
가중치 감쇠 (Weight Decay):	weight_decay: 0.0005

2.3 하드웨어 요구 사항

CPU (Intel i7-13620H)	총 코어: 10 코어, 16 쓰레드 / 코어성능 : 기본 2.4 GHz, 터보 4.9 GHz
GPU (NIVIDIA RTX 4060)	메모리 크기 : 8GB , 대역폭 : 272.0 GB/s
RAM	32GB
저장공간 (SSD)	414GB

2.4 소프트웨어 요구 사항

운영체제	Window 10
실행환경	Jupyter Notebook
Python	Python 12.0 이상
사용한 패키지	numpy 2.0.1 (고성능 수치 계산)
	pandas 2.2.3 (데이터 조작 및 분석)
	opencv-python l4.10.0.84 (컴퓨터 비전 및 이미지 처리)
	matplotlib 3.9.2 , seaborn 0.13.2 (시각화)
	torch 2.5.1 (모델 구축 및 훈련)
	ultralytics 8.3.38 (YOLO 모델 객체 탐지 및 이미지 분류)

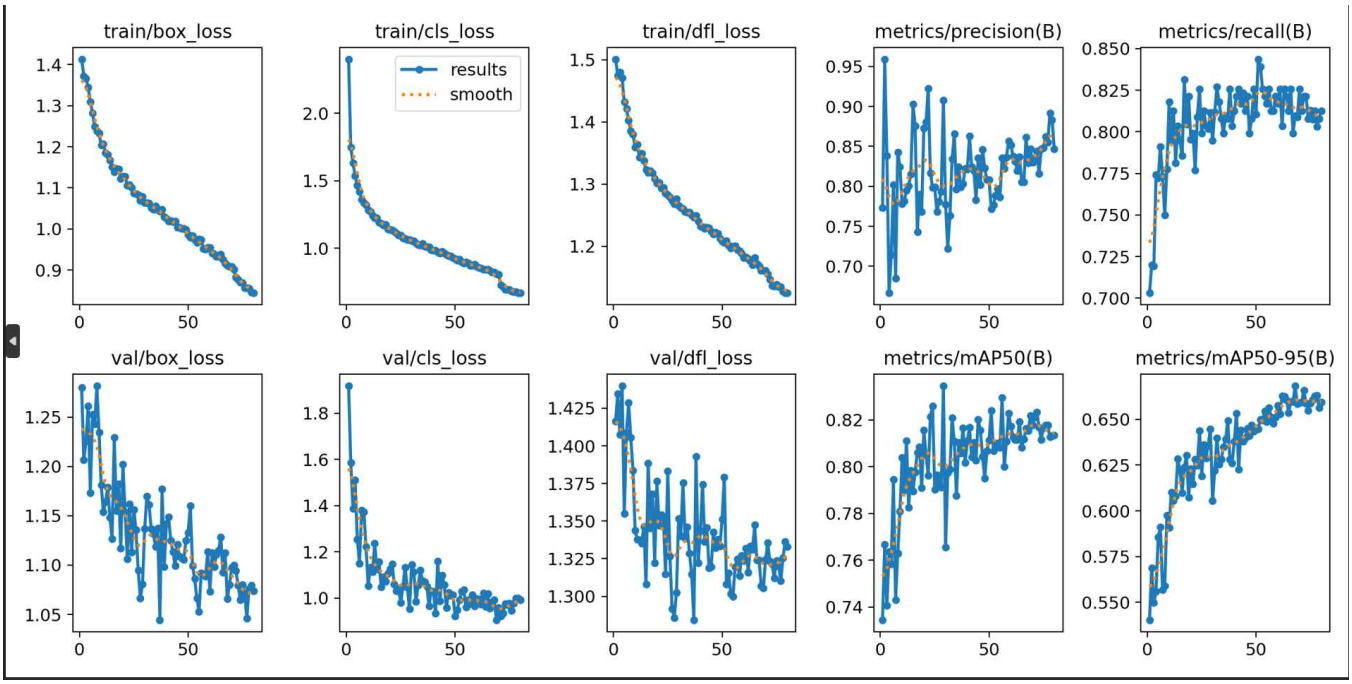
3. 학습결과

3.1 학습 평가

평가 기준값: 50번

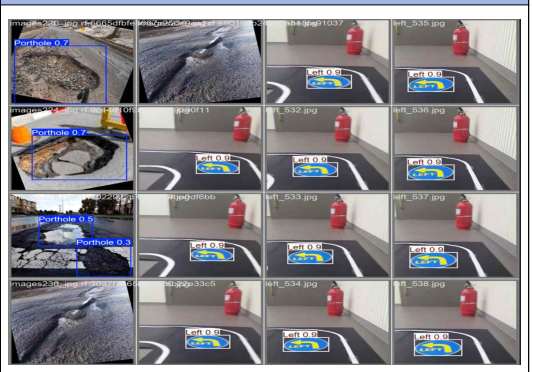
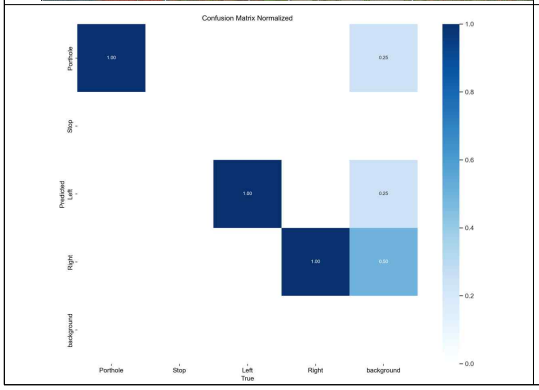
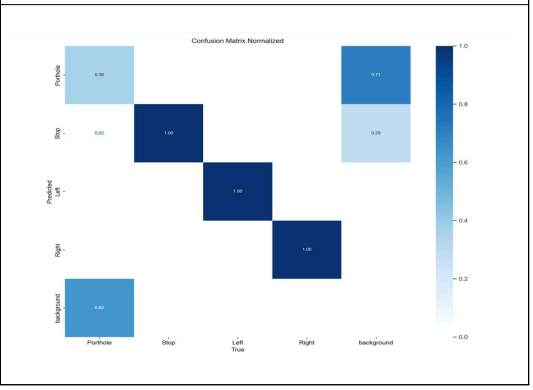
지표		평가	결과 값
손실	훈련	train/box_loss	0.95
		train/cls_loss	0.9
		train/dfl_loss	1.2
	검증	val/box_loss	1.10
		val/cls_loss	1.0
		val/dfl_loss	1.325
정확도		metrics/precision(B)	0.755
		metrics/recall(B)	0.825

3.2 시각화

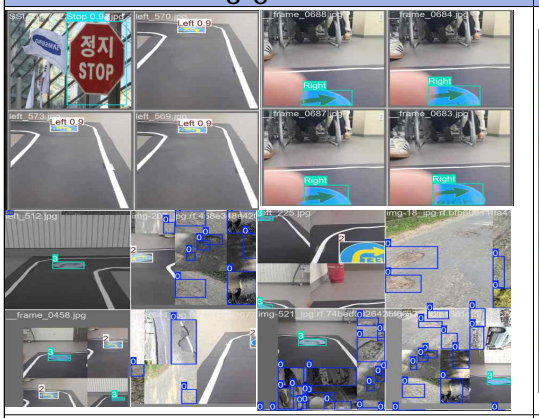
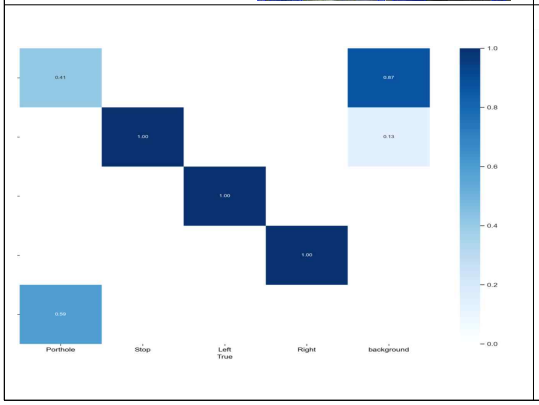
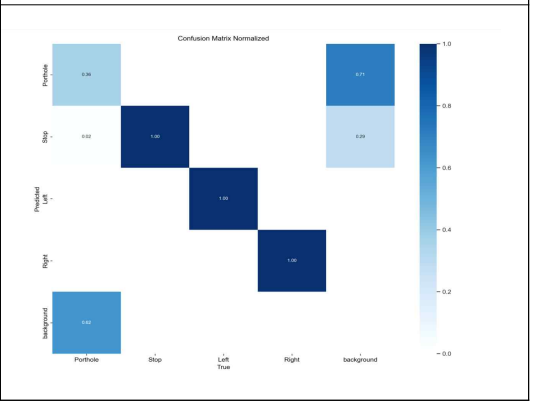


4. 비교분석

4.1 데이터셋

Public Dataset	Custom Dataset
	
	

4.2 데이터 증강

증강 X	증강 O
	
	

6. 결론 및 제언

6.1 결론

본 프로젝트는 객체 탐지 기반 자율주행 차량 로봇에서 객체 인식을 통한 주행과 포트홀 감지 및 자동 신고 서비스를 목표로 하였습니다.

- 검증 데이터셋에서 mAP 50-95(B)가 0.6 이상 목표를 설정하였으나, 성능은 일부 지표에서 목표를 하회하여 추가 개선의 여지가 확인되었습니다.
- Precision(B)은 0.755, Recall(B)은 0.825로 나타나 도로 환경에서의 높은 탐지율을 나타냅니다.

6.2 제언

1. 모델 성능 향상

- 직접 만든 test도로에서의 데이터 추출을 통해 성능을 개선, 포트홀 및 정지 표지판과 같은 특정 클래스에 대한 데이터 다양성을 확보하는 것이 중요합니다.
- 우리가 직접 촬영하고 제작한 데이터셋을 기반으로, 실제 도로 상황과 유사한 추가 데이터를 확보하여 모델의 일반화 성능을 높일 필요가 있습니다.

2. 실시간 시스템 구현

- 현재 ESP32-CAM을 통해 실시간으로 카메라 데이터를 수집하고 있으므로, 이를 활용하여 실시간 객체 탐지 및 분석 시스템을 딜레이에 대한 안정화하는 작업이 필요합니다.

3. 현장 테스트 및 시뮬레이션

- 직접 제작한 도로 및 시나리오를 기반으로 작업한 경험을 확장하여, TEST 도로 환경을 반영한 테스트 시뮬레이션을 추가로 수행해야 합니다.
- 직접 제작한 도로 및 시나리오를 기반으로 다양한 환경과 조건에서 성능을 평가하고, 안전성을 강화하는 방향으로 개선 작업을 이어나갈 필요가 있습니다.

5. 부록

학습률 (Learning Rate)	모델이 가중치를 업데이트하는 속도를 결정합니다. 너무 높으면 최적점을 지나칠 수 있고, 너무 낮으면 학습 속도가 느려질 수 있습니다.
배치 크기 (Batch Size)	한 번의 업데이트에 사용되는 샘플 수입니다. 큰 배치 크기는 더 안정적인 경량 업데이트를 제공하지만, 메모리 사용량이 증가합니다.
에포크 수 (Number of Epochs)	전체 데이터셋을 몇 번 반복하여 학습할지를 결정합니다. 너무 많으면 과적합(overfitting)될 수 있습니다.
모멘텀 (Momentum)	이전 업데이트의 영향을 반영하여 현재 업데이트를 조정하는 데 사용됩니다. 이는 학습 속도를 높이고 진동을 줄이는 데 도움을 줍니다.
가중치 감쇠 (Weight Decay):	과적합을 방지하기 위해 가중치에 패널티를 부여하는 방법입니다. 일반적으로 L2 정규화를 사용합니다.

성능평가 지표

train/box_loss	모델이 예측한 바운딩 박스와 실제 바운딩 박스 간의 차이를 측정하는 손실 값입니다. 일반적으로 L1 손실이나 L2 손실을 사용하여 계산됩니다. 이 값이 낮을수록 모델이 바운딩 박스를 정확하게 예측하고 있다는 것을 의미합니다.
train/cls_loss	클래스 손실로, 모델이 객체의 클래스를 올바르게 분류하는 능력을 평가합니다. 일반적으로 크로스 엔트로피 손실을 사용하여 계산되며, 이 값이 낮을수록 모델의 분류 성능이 좋다는 것을 나타냅니다.
train/dfl_loss	DFL(Distribution Focal Loss) 손실로, 객체 탐지에서 클래스 불균형 문제를 해결하기 위해 사용됩니다. 이 손실은 어려운 예제에 더 많은 가중치를 부여하여 모델이 더 잘 학습하도록 돕습니다.
metrics/precision(B)	정밀도(Precision)는 모델이 예측한 양성 샘플 중 실제로 양성인 샘플의 비율을 나타냅니다. 즉, TP / (TP + FP)로 계산됩니다. 높은 정밀도는 모델이 잘못된 긍정 예측을 적게 한다는 것을 의미합니다
metrics/recall(B)	재현율(Recall)은 실제 양성 샘플 중 모델이 올바르게 예측한 양성 샘플의 비율을 나타냅니다. 즉, TP / (TP + FN)으로 계산됩니다. 높은 재현율은 모델이 실제 양성을 잘 포착하고 있다는 것을 의미합니다.
metrics/mAP50(B)	mAP(Mean Average Precision) at IoU=0.5는 다양한 클래스에 대한 평균 정밀도를 측정합니다. IoU(Intersection over Union) 임계값이 0.5일 때, 모델의 성능을 평가하는 데 사용됩니다. 이 값이 높을수록 모델의 전반적인 성능이 좋다는 것을 나타냅니다.
metrics/mAP50-95(B)	mAP at IoU=0.5:0.95는 IoU 임계값을 0.5에서 0.95까지 변화시키면서 계산한 평균 정밀도를 나타냅니

	다. 이 지표는 모델의 성능을 더 엄격하게 평가하며, 다양한 IoU 임계값에서의 성능을 종합적으로 보여줍니다.
val/box_loss	검증 데이터셋에서의 바운딩 박스 손실입니다. 훈련 중 모델의 일반화 성능을 평가하는 데 사용됩니다.
val/cls_loss	검증 데이터셋에서의 클래스 손실입니다. 모델의 분류 성능을 검증하는 데 사용됩니다.
val/dfl_loss	검증 데이터셋에서의 DFL 손실입니다. 모델의 일반화 성능을 평가하는 데 사용됩니다.
lr/pg0, lr/pg1, lr/pg2	각 파라미터 그룹에 대한 학습률(Learning Rate)입니다. 모델의 훈련 과정에서 각 파라미터 그룹에 대해 설정된 학습률을 나타냅니다. 학습률은 모델의 수렴 속도와 성능에 큰 영향을 미칩니다.

하이퍼 파라미터

학습률 (Learning Rate)	모델이 가중치를 업데이트하는 속도를 결정합니다. 너무 높으면 최적점을 지나칠 수 있고, 너무 낮으면 학습 속도가 느려질 수 있습니다.
배치 크기 (Batch Size)	한 번의 업데이트에 사용되는 샘플 수입니다. 큰 배치 크기는 더 안정적인 경량 업데이트를 제공하지만, 메모리 사용량이 증가합니다.
에포크 수 (Number of Epochs)	전체 데이터셋을 몇 번 반복하여 학습할지를 결정합니다. 너무 많으면 과적합(overfitting)될 수 있습니다.
모멘텀 (Momentum)	이전 업데이트의 영향을 반영하여 현재 업데이트를 조정하는 데 사용됩니다. 이는 학습 속도를 높이고 진동을 줄이는 데 도움을 줍니다.
가중치 감쇠 (Weight Decay):	과적합을 방지하기 위해 가중치에 패널티를 부여하는 방법입니다. 일반적으로 L2 정규화를 사용합니다.