



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2021년 8월

석사학위 논문

정밀도로지도 갱신을 위한
YOLOv3 기반 도로시설물 객체 검출

조선대학교 대학원

토목공학과

이태영

정밀도로지도 갱신을 위한
YOLOv3 기반 도로시설물 객체 검출

Object Detection of Road Facilities using YOLOv3 for
High Definition Map Updates

2021년 8월 27일

조선대학교 대학원

토 목 공 학 과

이 태 영

정밀도로지도 갱신을 위한
YOLOv3 기반 도로시설물 객체 검출

지도교수 정 명 훈

이 논문을 공학석사학위신청 논문으로 제출함

2021년 4월

조선대학교 대학원

토 목 공 학 과

이 태 영

이태영의 공학석사학위 논문을 인준함

위원장 조선대학교 부교수 박 상 준 (인)

위 원 조선대학교 조교수 장 우 식 (인)

위 원 조선대학교 조교수 정 명 훈 (인)

2021년 5월

조선대학교 대학원

목 차

ABSTRACT

제 1장 서론	1
1.1 연구 필요성	1
1.2 연구 목적 및 방법	4
제 2장 연구 배경	5
2.1 선행연구 분석	5
2.1.1 LiDAR 데이터를 이용한 객체검출	5
2.1.2 이미지 데이터를 이용한 객체검출	7
2.2 객체검출 개요	10
제 3장 방법론	12
3.1 객체검출방법	12
3.1.1 YOLOv3	13
3.1.2 IoU(Intersection over Unit)	14
3.1.3 Confusion Matrix	15
3.1.4 Location Prediction	18
3.1.5 Multi-Labels Classification	19
3.1.6 Non-Maximum Suppression	20
3.1.7 Feature Extractor Network	21

3.2 전이학습	23
제 4장 실험	24
4.1 실험 데이터	24
4.2 실험 환경	30
4.3 실험 결과	31
4.3.1 도로시설물 객체검출 성능 평가	32
4.3.2 도로시설물 변화탐지 결과	36
A) 도로시설물 신규 객체 삽입	36
B) 도로시설물 객체 제거	38
C) 도로시설물 객체 클래스 변경	40
제 5장 결론	42
참고문헌	44

표 목 차

표 1.1 레이어축약코드	2
표 4.1 클래스 목록	26
표 4.2 데이터셋 분류	27
표 4.3 AWS EC2 사용 사양	30

그림 목 차

그림 1.1 자율주행 기술레벨	1
그림 1.2 연구 흐름도	4
그림 2.1 객체검출방법 변화	11
그림 3.1 One-stage Detector와 Two-stage Detector 비교	12
그림 3.2 Object Detector 비교	13
그림 3.3 IoU	14
그림 3.4 Confusion Matrix	15
그림 3.5 객체검출에서 Confusion Matrix가 사용되는 예시	16
그림 3.6 Precision-Recall Curve	17
그림 3.7 Bounding Boxes with Location Prediction	18
그림 3.8 YOLOv3 Multi-Labels Classification	20
그림 3.9 NMS 결과 예시	21
그림 3.10 YOLOv3 Architecture	22
그림 4.1 Annotation 예시	24
그림 4.2 도로환경 사진 예시	25
그림 4.3 클래스 객체별 분포	27
그림 4.4 Annotation 수정 예시	28
그림 4.5 학습 이미지 예시1	28
그림 4.6 학습 이미지 예시2	29
그림 4.7 시스템 아키텍처	30

그림 4.8 객체검출결과 31
 그림 4.9 도로시설물 객체검출 성능 32
 그림 4.10 도로주행 영상에서의 객체 검출속도 평가 예시 33
 그림 4.11 도로노면 객체검출 성능 34
 그림 4.12 표지판 객체검출 성능 35
 그림 4.13 신규 도로시설물 객체 검출 예시1 36
 그림 4.14 신규 도로시설물 객체 검출 예시2 37
 그림 4.15 제거된 도로시설물 객체 검출 예시1 38
 그림 4.16 제거된 도로시설물 객체 검출 예시2 39
 그림 4.17 변경된 도로시설물 객체 검출 예시1 40
 그림 4.18 변경된 도로시설물 객체 검출 예시2 41

ABSTRACT

Object Detection of Road Facilities using YOLOv3 for High Definition Map Updates

Lee, Tae Young

Department of Civil Engineering

Graduate School of Chosun University

In recent decades, extensive technological efforts are able to improve autonomous driving. For example, High-Definition(HD) map has been constructed to ensure safety for autonomous driving. More specifically, a vehicle equipped with a Mobile Mapping System (MMS) collects data on road conditions first. Then humans check and modify any technical failure in order to create the final HD map. However, due to a lack of technological development, most jobs are done manually, which costs a tremendous amount of time and money. Therefore, a system, which automatically detects any changes on the road and corrects them synchronously on the map, is highly recommended. This study aims to test methodologies to develop an efficient change detection system in a road environment. To achieve this goal, this study first tried to detect objects in the video using the YOLOv3 algorithm, which is one of the object detection algorithms. Subsequently, via transfer learning, YOLOv3 which detect traffic sign, road sign objects and traffic light in HDmap was constructed. It achieves 56.5 AP_{50} in detecting the road facilities. By comparing the images before and after the road construction, It can detect any changes in road facilities and record road sections that needed corrections. A follow-up study would be conducted to compare the results of this study to results of studies employing state-of-the-art object detection algorithms or open set-based change detection.

제 1장 서론

1.1. 연구 필요성

최근 자율주행 기술이 정밀도로지도(High Definition Map, HDmap)와 센서의 융합된 기술로 발전함에 따라 정밀도로지도의 구축에 대한 관련 기술 연구·개발 지원 필요성이 강조되고 있다(NGII, 2019). 자율주행차 기술이란 차량이 스스로 주행하는 것을 일컫는 말로 상용화를 위해 전 세계적으로 그 기술이 개발되고 있으며, 미국 자동차 기술자 협회(Society of Automotive Engineers, SAE)는 자율주행차 기술을 기술 레벨에 따라 level 1 ~ level 5까지 5단계로 나누고 있다. 이를 그림 1.1에서 설명하고 있다.



그림 1.1 자율주행 기술레벨 (한국교통안전공단, 2020)

그림 1에서 묘사된 것을 요약하자면 level 1부터 level 3까지는 운전자가 주행에 개입해야 하는 단계이고, level 4부터는 차량이 스스로 운전할 수 있는 단계이다. 지금의

기술은 level 3 정도에 해당하고 level 4 진입을 위해 연구되고 있는 시점이기에 아직 자율주행차 센서가 제대로 작동하지 않는 문제 혹은 차량이 사물을 제대로 인식하지 못하는 문제가 존재한다. 이러한 이유로 구축된 정밀도로지도를 통해 차량의 주변 환경에 대한 정보를 인식하는 도움이 필요하며 전 세계적으로 정밀도로지도 구축이 이루어지고 있다. 이에 따라 대한민국 또한 안전한 자율주행을 위한 정밀도로지도를 구축하는 노력을 하고 있다. 이러한 국가정책 및 산업발전에 기여하기 위해 2015년 국토지리정보원에서 국내 최초로 ‘자율주행차 지원 등을 위한 정밀도로지도 시범 구축 연구’를 시작했으며 2017년 본격적인 정밀도로지도 구축을 수행하고 있다(NGII, 2018). 여기서 말하는 정밀도로지도란 기존 전자지도(Digital Map) 기반의 국한된 교통정보 제공 서비스를 넘어 자율주행에 필요한 도로 선형, 차로 구분, 도로 표지판 등 도로환경 정보와 point of interest(POI) 정보를 자동차와 운전자에게 제공하여 안전한 도로 주행을 유도하는 3차원 정밀지도를 말한다(Park et al., 2019).

정밀도로지도 제작 과정을 살펴보면 먼저 작업계획을 수립하여 작업구간을 설정하고 mobile mapping system(MMS) 및 기준점 측량을 실시하여 3차원 점군데이터를 생성하는 표준자료를 제작한다. MMS 장비는 항법센서인 global navigation satellite system(GNSS), distance measurement instrument(DMI), inertial measurement unit(IMU)과 원격탐측센서인 light detection and ranging(LiDAR), 디지털카메라로 구성되고 이러한 장비를 탑재한 차량을 도로에 주행시킴으로써 데이터를 취득하고 3차원 점군데이터를 생성할 수 있다. 그런 다음, 구축된 자료를 기반으로 도로의 정밀한 객체도화 및 구조화 편집 등을 거쳐 최종적으로 정밀도로지도를 구축한다(NGII, 2015; NGII, 2016; NGII, 2019). 완성된 정밀도로지도는 14개의 레이어를 포함하고 있으며 그 구성요소를 표 1.1에서 확인할 수 있다.

표 1.1 레이어축약코드 (국토지리정보원, 정밀도로지도 구축 매뉴얼, 2019, 53P)

No.	축약코드	레이어명	레이어 설명
1	A1	A1_NODE(주행경로노드)	주행경로링크 연결점
2	A2	A2_LINK(주행경로링크)	차로중심선
3	A3	A3_DRIVWAYSECTION(구간)	터널, 교량 등 차도구간
4	A4	A4_SUBSIDIARYSECTION(부속구간)	휴게소 등 부속구간
5	A5	A5_PARKINGLOT(주차슬롯)	부속구간에 존재하는 주차장
6	B1	B1_SAFETYSIGN(안전표지)	교통안전표지

7	B2	B2_SURFACELINEMARK(노면선표시)	중앙선 등 표시 등 노면선
8	B3	B3_SURFACEMARK(노면표시)	유도선 등 노면표시
9	C1	C1_TRAFFICLIGHT(신호등)	신호등
10	C2	C2_KILOPOST(킬로포스트)	킬로포스트
11	C3	C3_VEHICLEPROTECTIONSAFET(차량방호안전시설)	중앙분리대 등 차량방호안전시설
12	C4	C4_SPEEDBUMP(과속방지턱)	과속방지턱
13	C5	C5_HEIGHTBARRIER(높이장애물)	높이 제한 시설
14	C6	C6_POSTPOINT(지주)	신호지주, 교통시설지주 등을 표현

국토지리정보원에서 2017년부터 정밀도로지도를 구축하고 있으나, 이는 국가의 C-ITS 추진 계획과 함께 2020년까지 경인, 서해안, 중부고속도로 등 총 9개 고속도로를 중심으로 한 정밀도로지도를 구축하려고 진행되는 것이며 이와 더불어 네이버, 현대엠엔소프트, 에스케이텔레콤 등 민간기업에서도 각자 정밀도로지도를 구축하려 노력하고 있다.

정밀도로지도는 구축하기에 많은 비용과 시간이 소요되고, 제작 완료 후 신속하고 주기적인 갱신이 뒤따르기 때문에 이를 자동화하기 위한 플랫폼이 필요한 상황이다. 기존 국가 공간정보 갱신체계를 살펴보면 1년 수준의 갱신체계로 운영되고 있고 이러한 연 단위 갱신체계를 적용한다면 자율주행차에서 요구하는 정확한 지도 제공에 한계가 존재하게 된다. 따라서 현재 정밀도로지도의 갱신방법은 국토교통부에서 제공하는 건설사업정보시스템과 도로관리통합시스템을 통해 도로공사 현황을 파악하여 전체적인 변화를 인식하여 새로이 구축 혹은 부분수정 방식으로 지도를 갱신하고 있다. 하지만 이는 소규모 시설공사에 대한 정보는 등재되지 않는 등의 문제들이 있어서 도로변화 정보를 수집하기에 한계가 존재하고 지도 갱신에 어려움이 있다(NGII, 2018).

이에 정밀도로의 안정적인 구축 및 갱신을 위해 주기적인 도로시설의 변화정보 수집을 위해 MMS 차량의 정기적 운행을 통한 변화정보 탐지기술을 확보하려고 노력 중이다. 하지만 대부분의 구축에 있어서 외국 기술에 대한 의존도가 높고 갱신작업은 수작업으로 진행되고 있어 많은 한계점이 존재한다. 이러한 문제를 다소 해결하기 위한 노력으로 국토부는 2018년 자율주행 지원을 위한 도로변화 신속 탐지, 갱신기술 개발 및 실증화 연구사업을 추진하여 관련 국내 기술 개발 중이고, TomTom과 Deepmap 등과 같은 지도 관련 기업들이 차량 내 지도를 업데이트하여 지도 정보를 제공하는 기술을 개발 중이다(Park et al., 2019).

1.2. 연구 목적 및 방법

현재까지 많은 시간과 노력을 들여서 정밀도로지도를 구축하고 있으며, 동시에 새롭고 효율적인 구축 방법을 개발 중이다. 나아가 정밀도로지도는 자율주행 안전을 위해 지도의 무결성 및 정확성이 요구되기에 최근 도로변화에 대한 변화탐지 및 지도 갱신 시스템 개발과 함께 정밀도로지도를 신속하게 갱신하기 위한 연구가 진행 중이다. 하지만 아직 정밀도로지도 갱신에 있어서 바로 적용되기에는 한계점이 존재한다. 그렇기에 현재 시점에서 변화탐지 과정만을 자동화하고 수정은 수작업으로 진행하여 부분변화만 빠르게 수정하여 지도에 갱신한다면 기존의 갱신 과정보다 시간과 비용은 줄이고 정확도는 유지할 수 있다고 판단된다. 본 연구는 이를 실현하기 위해 도로 영상을 이용하여 도로시설물 객체를 검출하는 기법을 이용하고 이전 영상의 객체탐지결과와 비교를 통해 빠르게 변화된 부분을 탐지하여 부분수정이 가능하게 하는 방안을 제시한다. 먼저 영상에서 객체를 검출하기 위해 딥러닝 기반 객체검출기술 중 하나인 YOLOv3(You Only Look Once version 3) 알고리즘을 사용했다. 이 알고리즘은 특정 객체만을 추출하도록 학습되었기에 정밀도로지도 갱신에는 적합하지 않았고, 도로시설물 변화탐지를 위한 모델을 완성하기 위해 전이학습 과정을 통한 새로운 학습을 추가했다. 이 모델로 정밀도로지도 구축 시 사용되었던 예전 사진과 시간이 지난 뒤 촬영된 사진의 객체검출결과를 비교함으로써 도로시설물의 변화가 발생하여 수정이 필요한 구간을 탐지했다. 연구의 전체 흐름은 그림 2.1과 같다.

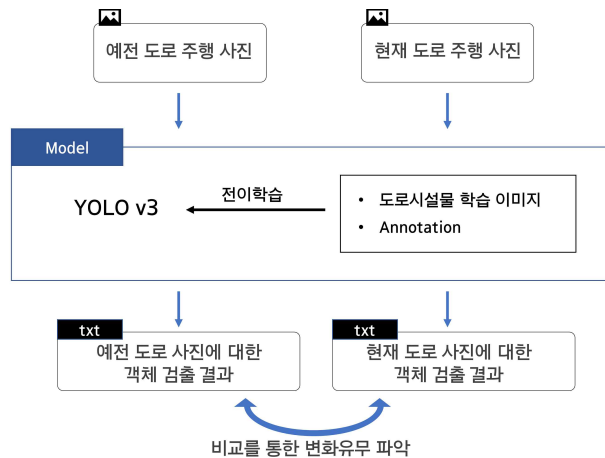


그림 2.1 연구 흐름도

제 2장 연구 배경

2.1 선행연구 분석

앞 장에서 살펴본 바와 같이 정밀도로지도는 LiDAR와 디지털카메라가 포함된 MMS 장비를 탑재한 차량을 이용하여 도로 및 주변 시설물 등의 정보를 취득하고 이들에 대하여 도화 작업을 수행함으로써 구축하고 있다. 정리하자면, 안전한 자율주행 시스템체계를 갖추기 위해서 도로시설물의 최신성을 유지한 정확한 정밀도로지도를 만들어야 하며, 이러한 목적을 달성하기 위해 신속하고 주기적인 갱신이 필요하기에 LiDAR 데이터 혹은 카메라 이미지 데이터를 사용하여 객체를 검출하고 변화를 탐지해 부분 수정작업을 진행하는 방식이 바람직하다. LiDAR와 이미지 데이터를 이용한 객체검출 연구는 많이 수행되고 있으며, 먼저 LiDAR 데이터를 이용한 객체검출연구를 살펴보았다.

2.1.1 LiDAR 데이터를 이용한 객체검출

LiDAR는 고출력 펄스 레이저 신호를 발사하고 목표물에 반사되어 돌아오는 시간을 이용하여 거리를 측정하는 센서이다. 간단히 말하면 레이저 거리측정기라고 할 수 있다. 정의된 센서 좌표계상에서 설계된 방향으로 레이저를 발사하고 거리를 측정하면 방향과 거리를 조합하여 반사지점의 3차원 위치를 결정할 수 있게 된다. 여기서 얻어진 3D Point를 포인트 클라우드(Point Cloud)라고 하며, LiDAR 이동측량 장치는 차량에 장착되어 3차원 공간자료인 포인트 클라우드를 획득함으로써 정밀도로지도를 구축하게 한다.

LiDAR를 이용해 얻어진 포인트 클라우드 데이터를 객체별로 구분하여 인식하는 연구들이 진행되었는데, Hata and Wolf(2014)은 차선과 아스팔트에 대한 LiDAR 포인트 데이터를 분할하여 차선을 검출하는 연구를 진행했지만, 1m 정도의 위치 오차가 존재하는 연구였다. Qin and Gruen(2014)은 입체적으로 복잡한 지상 이미지에서 변화를 탐지할 경우 사진을 촬영한 각도가 다를 경우 이미지 매칭이 어려운 문제를 해결하기 위해 MLS 포인트 클라우드와 이미지 두 개의 서로 다른 데이터를 비교하고 최적화함으

로써 단일 이미지 기반 변화감지, 단일 포인트 클라우드 기반 변화감지의 한계점을 해결하는 연구를 진행했다. 이 연구는 위성영상에 대한 변화탐지 연구만이 주로 이루어졌을 시기에 지상 이미지에 대한 변화탐지 연구의 기반이 되었다. Wang et al.(2018)은 포인트 클라우드에서 인스턴스 분할을 위해 similarity group proposal network(SGPN)이라는 딥러닝 프레임워크를 제시하였다. 이는 3D 객체 분할을 위해 similarity metric learning을 적용시켜 각 포인트 쌍 간의 유사성을 파악하고 그룹화하는 프레임워크이다. 이를 통해 객체 탐지 및 세분화를 가능하게 했다. Jinno et al.(2019)은 LiDAR 센서가 장착된 모바일 로봇을 이용해 3D 변경 감지 방식으로 효과적으로 지도를 업데이트하는 연구를 진행했다. 나아가 차량에 센서를 접목시켜 도로를 주행함으로써 새로 등장하거나 사라진 부분을 탐지하는 연구 또한 보여주었다. Ma et al.(2019)은 카메라와 LiDAR를 사용하여 각각의 방식으로 교통 표지판을 탐지하고 이를 융합하여 센티미터 단위의 오차를 가진 측정 위치를 파악하는 연구를 진행했다. 두 개의 데이터를 융합함으로써 빛이 적어 어두운 결과를 갖는 카메라 영상에서 객체를 잘 탐지할 수 없던 기존연구와 달리 객체를 잘 분별하는 결과를 보여주었다. 이러한 연구들과 같이 자율주행을 위한 지도작성기술 simultaneous localization and mapping(SLAM)이 개발되었다. SLAM은 지도를 생성하고 현재 위치를 파악하여 자율주행이 가능하도록 하는 기술이지만 인식 오류와 같은 문제를 완벽히 보완하지 못했다. 이렇듯 센서와 로봇 연구영역에서 자율주행을 위해 객체를 인식하여 변화를 탐지하고 이를 3D 객체로 구축하는 연구가 진행되고 있지만 아직 그 완성도와 실효성은 떨어지는 시점이다. 2021년 현재로는 MMS 장비로부터 데이터를 취득하고 보정한 데이터를 수작업으로 도화하여 정밀도로지도맵을 구축하고 있는 것이 현실이다. LiDAR와 카메라 데이터를 복합적으로 사용하여 단일 연구 시 발생했던 단점을 보완하는 연구들이 진행되었지만, LiDAR는 고가의 장비이고, 지도를 구축하는 모든 차량에 탑재하기에는 현실적인 문제가 있다. 또한, LiDAR는 거리를 측정할 수 있고 객체의 입체적 형태를 파악하기에 좋다는 장점이 있지만 카메라 영상보다는 해상도가 떨어지고 용량은 크다는 단점이 있다. 요약하자면, LiDAR 장비는 매우 고가로 MMS 장비를 탑재한 차량 수는 제한적이므로 전국 모든 도로를 갱신하기 위한 모든 데이터를 수집하기에는 한계가 존재한다. 결론적으로, 카메라와 같은 저가의 장비를 통해 변화를 탐지하는 것이 더 효율적이라는 관점에서 본 논문은 이미지 데이터를 이용해 객체를 검출하여 변화를 탐지하고자 한다.

2.1.2 이미지 데이터를 이용한 객체검출

디지털 카메라를 이용하여 사물을 촬영한 결과물을 디지털 이미지라 한다. 디지털 이미지에서 사람, 동물, 사물과 같은 특정 class에 대한 객체를 검출하는 object detection은 이미지에서 어떤 객체가 어디에 존재하는지에 대한 정보를 제공하는 것을 목표로 한다. 이러한 관점에서 object detection은 인간이 사물을 보고 인지하는 것처럼 컴퓨터가 이미지를 보고 특정 물체를 검출하는 성능을 높이는 연구를 의미하는 “일반적인 객체검출 영역”과 보행자 추적 및 텍스트 감지와 같은 “객체검출 활용” 2가지 연구 주제로 나뉠 수 있다(Zhengxia et al., 2019). 먼저 객체검출 활용에 해당하는 연구 및 사례에 대하여 살펴봄으로써 정밀도로지도 변화탐지에 활용할 수 있는지 확장성을 확인했다.

첫 번째 예시로는 객체검출 방법을 사용하여 간판 혹은 상품에 존재하는 로고를 인식하는 연구들이 있다. 도로시설물에서 표지판과 같이 비슷한 성격을 갖는 로고는 주변 상황을 이해하고 길을 찾는 데 도움이 되며 차량 번호판 인식과 같이 지능형 교통 체계를 구축하는 역할을 한다(Chun et al., 2013). Chun et al.(2013)은 차량 로고 이미지를 convolutional neural network(CNN) 기반으로 특징을 추출하여 차량의 브랜드를 인식한 결과와 전통적인 검출기법인 scale invariant feature transform(SIFT)를 이용한 결과와 비교하여 딥러닝 기반의 방법이 이미지 인식에 있어서 더 나음을 보여주었다. 이후에는 로고를 검출하는데 그치는 것이 아닌 두 개의 이미지를 비교하여 변화를 탐지하는 연구들이 진행되었다(Su et al., 2017; Tuzko et al., 2018; 김민정 et al., 2020). 이러한 연구들에서는 로고 검출 단계에서 open set이라는 개념을 접목하는데, 이미지에서 로고 검출 시 해당 로고 이름을 직접 알려주는 기존 object detection의 closed set 방식과 달리 open set 방식은 상위 분류 범주인 로고로 객체를 검출한 후 분류 단계를 따로 한번 더 거침으로써 최종적인 로고 이름을 매칭시키는 방식이다. 이러한 방법은 사전에 학습되지 않은 로고 브랜드를 로고로 인식하여 검출할 수 있다는 점과 분류 단계가 따로 존재하기 때문에 새롭게 추가학습이 자유롭다는 장점이 있다. 하지만 로고가 아닌 것도 로고로 탐지되는 경우가 존재하며 객체검출을 위해 두 단계를 거쳐야 한다는 단점이 있다.

다음 예시로는 지도를 갱신하기 위해 street view 이미지를 이용하여 도로변화를 탐지하는 연구들이 있다. Alcantarilla et al.(2018)은 거리 사진에 대하여 SLAM 기법을 사용해 3D 맵을 구성하고, deconvolutional network(DN)에 픽셀 단위 비교를 위해 변화 전과 후 이미지를 학습시켜 마스킹 처리된 결과로 탐지하는 연구를 진행했다. 하지만 마스킹 처리된 결과는 정확히 객체를 포함하고 객체화하기에는 한계가 존재했다. Jo et al.(2018)은 정밀도로지도 구성요소 중에서 교통 표지판에 대한 변경사항을 감지하고 업데이트하기 위해 이미지 데이터를 이용한 SLAM 기법을 사용했고, 이는 Dempster - Shafer theory를 적용해 객체가 원래와 같은지, 삭제되었는지, 새롭게 등장했는지 분별했다. 하지만 위치오차가 1m 정도가 발생하며 앞 절에서 언급한 것처럼 SLAM은 아직 완성도가 떨어지기 때문에 많은 도로시설물을 검출하고 갱신하는 것에 적합하지 않았다. Revaud et al.(2019)은 건물 내에 존재하는 매장이 변화가 발생할 때 수작업으로 갱신하는 기존 작업에서 로고와 같은 간판 이미지 쌍 비교를 통해 자동으로 변화가 발생한 위치(POI)를 알려주는 연구를 진행했다. 이미지 쌍 비교에 있어서 사진끼리 유사도를 학습시키는 deep metric learning 방법을 사용함으로써 새로운 POI가 나타나거나 변경, 제거, 변화 없음과 같은 변화를 바로 갱신할 수 있었다. Heo et al.(2020)은 HDmap 갱신을 위해서 서로 다른 종류의 데이터 형태인 HDmap 객체와 카메라 이미지 데이터를 adversarial learning를 통해 인풋 도메인 문제를 해결하고 이들의 유사도를 학습시켜 변화를 탐지했다. 이 연구는 저비용인 카메라 이미지 데이터만을 사용했다는 것과 HDmap과 다른 점이 발견된다면 즉시 HDmap에 갱신할 수 있다는 장점을 보여주며 앞으로의 갱신체계에 대한 비전을 제시했다. 하지만, 도로시설물 객체의 제거 또는 추가에 있어서는 좋은 결과를 보였지만 클래스 변경은 상대적으로 좋지 않은 결과를 보여주었고 도로 노면에 대한 변화탐지만 수행했기 때문에 정밀도로지도 갱신에 적용하기 위해서 추가적인 연구가 필요하다.

객체검출기법을 통한 정밀도로지도 갱신 및 활용 가능성을 확인하고자 LiDAR 데이터와 카메라 이미지를 이용한 객체검출 기법 적용사례를 살펴보았다. 정리하자면, LiDAR 데이터와 이미지를 같이 사용함으로써 객체 인식 정확도를 높이는 연구들이 있었지만 LiDAR 장비가 비싸다는 단점이 있으며, 향후 전국단위의 정밀도로 지도를 갱신하는데 효율성이 뛰어나지 않다고 판단된다. 이와 같은 이유로, 지도갱신기

술을 연구하는 업체들이 이미지 데이터를 이용하여 변화를 탐지하고 이를 벡터지도로 변환하여 자동 생성하는 연구를 진행하고 있지만, 그 결과는 공개되지 않고 있다. 또한, 카메라 이미지를 통해서 넓은 지역의 변화를 빠르게 탐지하려는 목적을 가지며 정밀도로지도객체에 해당하는 수많은 도로시설물 객체를 검출하고자 하는 연구 사례는 없었다. 이러한 맥락에서 본 연구는 효율적인 정밀도로지도 갱신을 위해서 LiDAR 장비보다 가격이 저렴한 카메라 장비로 구축된 이미지 데이터만을 이용하여 도로시설물 객체검출 및 변화탐지를 진행하고자 한다. 기존 지도갱신을 위한 객체검출활용 연구에 있어서 특정 객체만을 검출하는 한계점을 보완하고자 표지판 인식 및 도로 노면 표시와 같은 다양한 객체를 인식하고자 하였으며, 이를 통해 도로환경을 파악하고 그 변화를 확인하는 연구를 하고자 하였다.

2.2 객체검출 개요

앞 절에서는 object detection 연구 중 활용영역에 대해 살펴봄으로써 정밀도로지도 변화탐지를 위해 이미지 데이터를 활용하는 방법이 효율적인 방법이라고 판단할 수 있었고, 이번 절에서는 활용하기에 앞서 연구에 사용할 객체검출 기법을 선택하기 위해 객체검출 기법 발전 과정에 대해 살펴보고자 한다. 기존에 이미지를 분석하여 객체를 인식하고 구별하기 위해 컴퓨터 비전(computer vision) 영역에서 많은 연구가 진행되고 있으며 이를 위해 전통적인 특징 추출 기법과 딥러닝 기반의 특징 추출 기법이 개발되고 있다. 컴퓨터 비전이 목표하는 과제는 classification, instance segmentation, object detection 등 여러 가지가 있지만, 그중에서 object detection은 이미지 내 찾고자 하는 객체를 박스로 영역 표시하여 객체가 어디에 있는지 위치를 알려주고 그 객체가 무엇인지 파악하기 위해 후보 class를 매칭하여 알려주는 방법이다. 먼저 딥러닝 기법이 객체검출방법에 적용되기 전 초기의 전통적인 객체검출기법으로 viola jones detectors(VJD), histogram of oriented gradients(HOG), deformable part-based model(DPM) 등과 같은 특징추출 알고리즘이 사용되었다(Viola and Jones, 2001; Dalal and Triggs, 2005; Felzenszwalb et al., 2008). 하지만 대상이 가려진 경우 혹은 특징이 명확하지 않은 경우에 있어서 검출 결과가 좋지 않은 한계점이 있었다. 또한, 객체검출까지의 시간이 많이 소요되었다. 한편, 2014년부터 GPU와 같은 컴퓨터 하드웨어와 딥러닝 알고리즘의 발전으로 딥러닝 기반 객체검출이 효과적으로 개발되기 시작하였으며, 해당 연도에 Girshick et al.(2014)이 연구한 regions with convolutional neural network(R-CNN)이 object detection에 CNN을 적용한 첫 논문으로써 주목 받았다. 이는 이미지를 학습할 때 객체가 있을법한 위치를 탐지하는 region proposal을 수행한 후 각 영역마다 CNN에 학습시켜 분류하는 방식으로 객체를 검출하였다. 이를 시작으로 사람들의 관심 속에서 다양한 방식으로 후속연구가 활발하게 진행되고 있다.

Object Detection Milestones

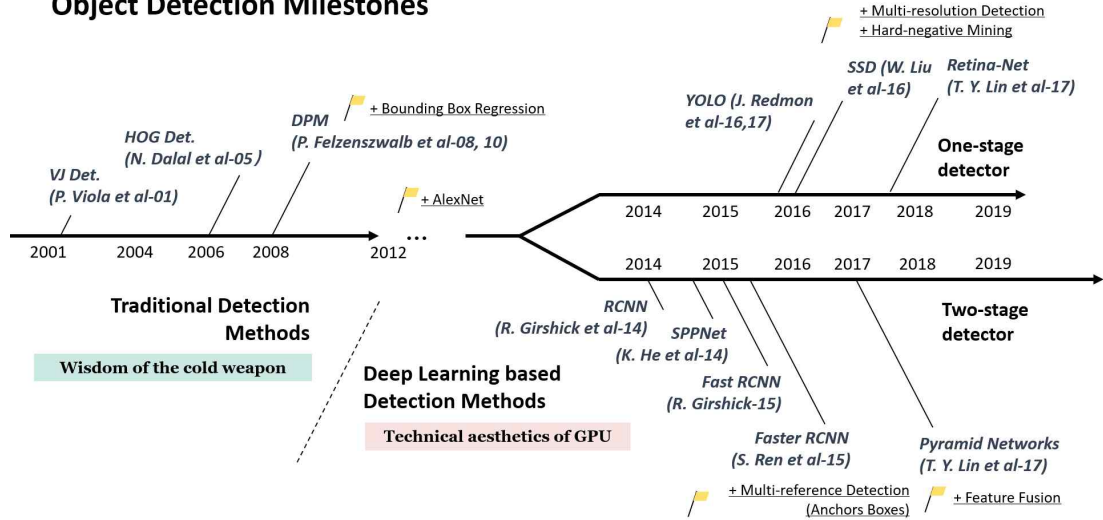


그림 2.2 객체검출방법 변화 (Zou et al., 2019, p. 2)

그림 2.2는 이러한 20년간의 object detection 연구흐름을 정리한 그림이다. 이러한 딥러닝 기반 객체검출은 크게 one-stage detector와 two-stage detector로 나뉜다. one-stage detector의 대표적 예시로는 YOLO, SSD, Retina-Net이 있으며(Redmon et al., 2016; Liu et al., 2016; Lin et al., 2017), two-stage detector의 대표적 예시로는 SPPNet, Fast RCNN, Faster RCNN, Pyramid Networks가 있다(He et al., 2015; Girshick, 2015, Ren et al., 2015; Lin et al., 2017). 이와 같이 검출 성능을 향상시키는 연구를 비롯하여 검출 속도를 빠르게 하기 위한 연구, 딥러닝 학습 과정을 효율적으로 구성하는 방법 등 수많은 연구들이 진행되고 있다.

제 3장 방법론

3.1 객체검출방법

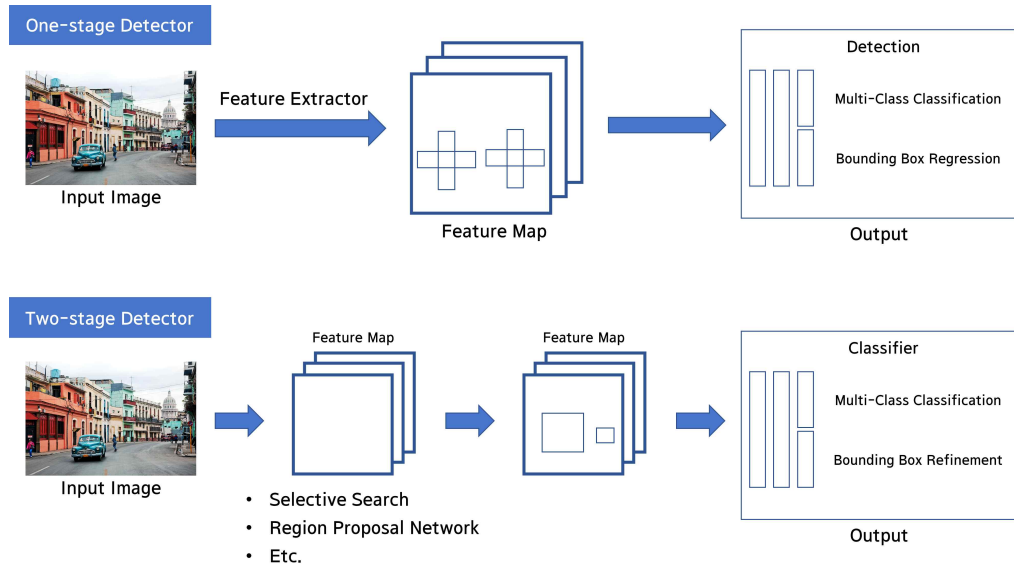


그림3.1 One-stage Detector와 Two-stage Detector 비교

그림 3.1은 앞 절에서 언급된 2가지의 딥러닝 기반 객체검출기법에 대한 방법론을 설명하고 있다. 이들은 이미지에 객체가 어디 존재하는지 결정하는 localization과 그 객체가 무엇인지 결정하는 classification 과정을 수행하며, 그림과 같이 localization과 classification을 동시에 진행하여 객체를 검출하면 one-stage detector, 순서대로 2단계를 나누어 수행하면 two-stage detector로 분류하여 설명하고 있다. 전자의 방법은 속도에 중점을 둔 방법이고, 후자의 방법은 정확도에 중점을 두었기 때문에 연구 목적이 달라 검출을 위한 구조적 차이를 갖는다. 본 연구는 앞으로 전국단위로 구축될 정밀도로지도 변화탐지를 위해서는 많은 양의 도로시설물 객체를 확인해야 하기 때문에 객체 검출에 있어서 정확도뿐만 아니라 속도도 중요하다고 판단했다. 이를 위해서, 검출속도가 빠른 one-stage detect methods 중에서 정확도가 two-stage detect methods에 비해 뒤떨어지지 않는 YOLO 기법을 사용하여 연구를 진행하였다.

3.1.1 YOLOv3

도로시설물의 상황이 변하여 정밀도로지도를 부분 수정하기 위해서는 지도 구축 시 사용했던 영상과 이후 촬영된 영상에 있어서 도로시설물 객체를 검출하여 비교해야 한다. 이를 위해 Redmon and Farhadi(2018)에서 제안된 딥러닝 기반 객체검출기법 YOLOv3를 사용했다. YOLO는 앞 장에서 설명한 one-stage detector의 일종으로 다른 객체검출방법보다 빠르고 상대적으로 좋은 성능으로 알려져 다양한 연구 영역에서 활용되는 기법이다. 본 논문에서 사용한 YOLOv3는 이름 그대로 YOLO 3번째 버전이며 2018년에 발표되었다. 이전 버전인 YOLOv2와 비교했을 때 backbone network 변경과 더불어 feature pyramid network와 유사한 기법을 적용한 3개의 feature map output에서 3개의 서로 다른 크기의 anchor box를 갖는 detection 방법을 사용하여 다양한 크기를 갖는 객체들을 검출할 수 있도록 했고 종합적인 성능 개선이 이루어졌다. 많은 객체검출기법 중에서 YOLOv3가 갖는 특징을 자세히 설명하기에 앞서 대표적인 딥러닝 기반 객체검출기법 간 검출 성능과 속도를 평가한 그림을 통해 선택한 이유를 설명하고자 한다. two-stage detector의 대표예시인 Faster R-CNN과 one-stage detector의 다른 기법들과의 정확도를 비교한 표를 그림 3.2에서 보여주고 있다.

	backbone	AP	AP ₅₀	AP ₇₅	AP _s	AP _M	AP _L	time
Two-stage methods								
Faster R-CNN+++	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9	-
Faster R-CNN w FPN	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2	172
Faster R-CNN by G-RMI	Inception-ResNet-v2	34.7	53.3	36.7	13.5	38.1	52.0	-
Faster R-CNN w TDM	Inception-ResNet-v2-TDM	36.8	57.7	39.0	16.2	39.8	52.1	-
One-stage methods								
SSD513	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8	125
DSSD513	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1	156
RetinaNet	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2	90
RetinaNet	ResNetXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2	198
YOLOv2	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5	-
YOLOv3	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9	51

그림 3.2 Object Detector 비교 (Redmon and Farhadi, 2018, p. 3)

그림 3.2에서 time은 millisecond 단위로 이미지 한 장을 처리하는데 걸리는 시간을 의미하며 작을수록 속도 측면에서 뛰어난 기법이고, AP 는 객체검출 성능을 의미하는 평가 지표이다. 먼저 AP_{50} 은 intersection over unit(IoU)가 0.5 이상의 값을 가지면 정답으로 인정하는 경우로 그 점수가 높을수록 성능이 좋다. AP_{75} 는 IoU가 0.75 이상의 값, AP 는 AP_{50} 부터 AP_{95} 까지 0.05씩 값을 IoU threshold를 증가시키며 구해진 값을 평균하는 경우이다. AP_S 는 작은 크기의 이미지, AP_M 는 중간 크기의 이미지, AP_L 는 큰 크기의 이미지에 대한 정확도를 의미한다. 일반적으로 IoU가 0.5 값을 갖는 경우 알고리즘이 예측한 bounding box 영역과 실제 bounding box 영역이 67% 정도 겹치기 때문에 YOLO는 AP_{50} 을 객체검출 성능으로 사용하고 있다. 결론적으로 YOLOv3를 사용한다면 도로주행 영상에서 해당 도로시설물을 빠르게 검출할 수 있고, AP_{50} 값을 비교한 결과를 확인한 결과 그 정확도가 뛰어난 결과임을 확인할 수 있다.

3.1.2 IoU(Intersection over Unit)

객체검출에서는 성능을 평가하기 위해서 IoU라는 지표를 사용하여 예측결과 (Prediction)와 실제(Ground Truth)가 얼마나 정확하게 겹치는지 0과 1 사이 값으로 수치화하게 된다. 이는 객체검출 결과 예측된 bounding box와 실제 bounding box의 교집합 면적을 전체 bounding box 합집합 면적으로 나눈 값으로 그림 3.3에서 보여주고 있다. 여기서 도출된 IoU 결과값은 AP_x 와 같이 특정 x에 대한 threshold 값으로 지정하여 예측을 성패를 결정하는 성능지표로 사용하게 된다.

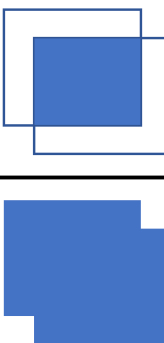
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


그림 3.3 IoU

3.1.3 Confusion Matrix

IoU값에 대해 특정 threshold를 설정하여 confidence score를 구해 객체검출에 있어서 성공 여부를 결정하는 판단기준을 세운 뒤 precision-recall curve로 객체검출의 성능을 최종 평가한다. 여기서 나오는 precision과 recall은 그림 3.4에서 설명하는 confusion matrix 요소를 통해 구할 수 있다. 먼저, TN은 true negative의 약자로 실제로 존재하지 않는데, 예측도 존재하지 않음으로 두 개의 결과 값이 negative로 일치된 경우를 의미하고, FN은 false negative의 약자로 이미지에 객체가 존재하나 객체검출 결과 객체를 발견하지 못 하는 경우를 뜻한다. 반대로, FP는 false positive의 약자로 이미지에 객체가 존재하지 않지만 객체검출 결과 객체를 발견하는 경우를 의미하고, TP는 true positive의 약자로 이미지에도 객체가 존재하고 객체검출결과에도 잘 객체를 발견한 경우를 말한다.

	[Predicted value]	
	Negative	Positive
[Ground truth]		
Negative	TN	FP
Positive	FN	TP

그림 3.4 Confusion Matrix

그림 3.5는 객체검출에서의 confusion matrix가 사용되는 사례를 설명하고 있다. 이 이미지에서 TP는 예측과 실제 box가 겹치지며, 객체가 객체임을 잘 인식한 경우이며, FN은 실제 객체는 존재하나 예측결과가 객체를 인식하지 못하여 객체검출을 하지 못하고 놓친 경우를 의미한다. 마지막으로, FP는 객체를 잘못 맞춘 경우를 의미하며, class가 제대로 매칭되지 않은 경우, IoU가 0.5보다 작아 정답으로 인정되지 않는 경우, 배경을 객체로 인식하여 잘못 검출한 경우를 예시로 들 수 있다.

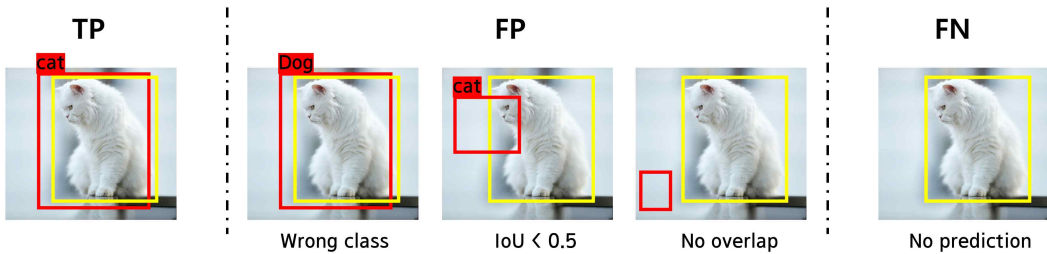


그림 3.5 객체검출에서 Confusion Matrix가 사용되는 예시

$$Precision = \frac{TP}{FP + TP} \dots\dots\dots (1)$$

$$Recall = \frac{TP}{FN + TP} \dots\dots\dots (2)$$

confusion matrix를 통한 식 (1)을 보면 알 수 있듯이 정밀도(Precision)의 분모는 검출결과 positive라고 예측한 것들의 집합을 의미한다. 이 중에서 얼마나 잘 맞았는지를 나타내는 비율이다. 식 (2)를 보면 재현율(Recall)은 실제 총 positive 중에서 몇 개나 검출했는지 비율을 의미한다. 정밀도가 좋으려면 FP가 작아야하고, 재현율(Recall)이 좋으려면 FN이 작아야한다. 이러한 성질에 따라 정밀도와 재현율은 일반적으로 서로 반비례 관계이며, 연구목적에 따라 정밀도와 재현율 둘 중에서 상대적으로 중요하게 여기는지 다를 수 있다. 예를 들어, 병 진단 같은 경우 양성을 음성으로 잘못 판단하는

경우 큰 문제가 되기 때문에 재현율을 중요하게 여기고 스팸메일 분류의 문제에서는 정밀도에 집중해서 일반메일이 스팸메일로 분류되어 휴지통으로 분류되지 않게 하는 것이 중요하다. 하지만, 객체검출연구에서는 어느 한 가지 값이 중요하다기보다 둘 다 좋아야한다. confidence score에 대한 threshold가 너무 낮아 이미지 전체에 후보 bounding box가 많다면 개수를 의미하는 재현율은 높아지겠지만 정밀도는 낮아질 것이고, threshold가 너무 높다면 후보 bounding box가 적어져 정밀도가 높아지기 때문에 재현율과 정밀도 둘 모두를 고려한 average precision(AP)를 구하기 위해 precision-recall curve를 활용해서 curve의 아래 면적을 계산한다.

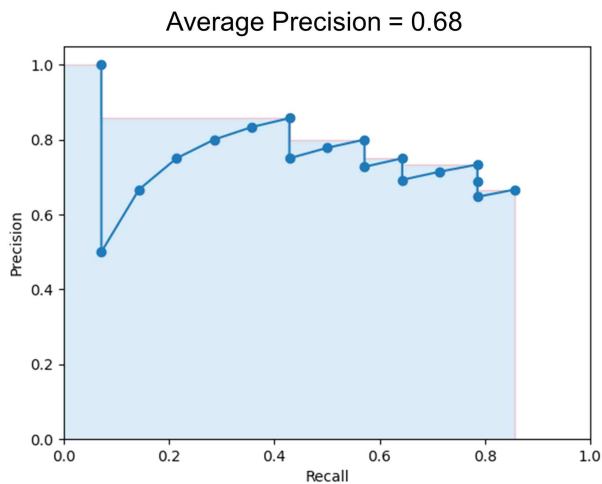


그림 3.6 Precision-Recall Curve

그림 3.6에 precision-recall curve 예시가 있다. 이는 confidence score threshold 변화에 따른 정밀도와 재현율 값을 나타낸 것이며 x축에 재현율 값을 y축에 정밀도 값을 놓는다. 여기서 AP를 계산하기 전에 계산의 편의성과 속도 개선을 위해 오른쪽 최대 정밀도 값을 연결한 보정 곡선을 만든 후 AP값을 구한다. 단일 객체를 의미하는 1개의 class를 다루는 것이 아닌 다중 class를 다루기 때문에 각각의 객체별로 AP를 구하고 이를 평균을 낸 mean average precision(mAP)를 결정하게 된다.

3.1.4 Location Prediction

객체검출은 object의 중심 X,Y좌표(b_x, b_y)와 박스의 크기를 나타내는 너비(b_w)와 높이(b_h)를 의미하는 4개의 숫자로 최종예측결과인 bounding box를 표현한다. 이를 위해 YOLOv3는 피쳐 맵 기반에서 anchor Box를 서로 다른 3개의 피쳐맵에서 서로 다른 크기의 앵커박스 3개를 사용하여 개별 그리드칸마다 총 9개의 anchor box를 k-means clustering 하여 Bounding box를 계산한다. 그림 3.7은 box 위치를 결정하기 위해 계산하는 값을 보여주고 있다.

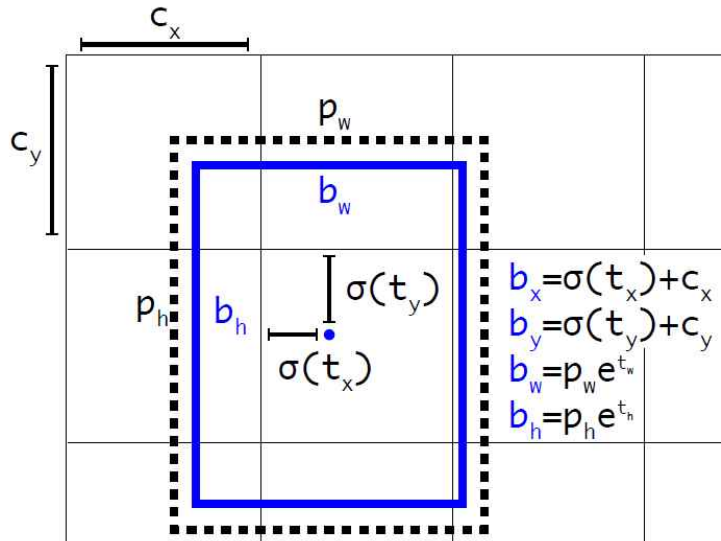


그림 3.7 Bounding Boxes with Location Prediction (Redmon and Farhadi, 2018, p. 3)

$$\sigma(t_o) = \Pr(object) \cdot IoU(b, object) \dots\dots\dots (3)$$

$$b_x = \sigma(t_x) + c_x \dots\dots\dots (4)$$

$$b_y = \sigma(t_y) + c_y \dots\dots\dots (5)$$

YOLO는 이미지를 여러 개의 그리드 셀로 나눈 뒤 각각의 셀이 하나의 Object detection을 수행하는 것을 기본으로 하며, c_x, c_y 는 각각의 그리드 셀 좌상단 x좌표, y

좌표이다. t_x, t_y, t_w, t_h 는 예측된 모델 offset 값을 의미하며, t_x 는 박스 중심 x좌표, t_y 는 박스중심 y좌표, t_w 박스 너비, t_h 박스 높이 shift를 나타내는 값이다. bounding box의 최종 x,y 좌표를 구하기 위해 $\sigma(t_o)$ 를 계산하는데 이는 logistic regression을 사용하여 box에 대한 objectness score(confidence score)를 구하는 과정이다. 이는 0과 1 사이의 값으로 object인지 아닌지를 판별한 값과 그에 해당하는 IoU 값을 곱한 결과를 c_x, c_y 에 더해지게 된다. t_x, t_y 가 0이 된다면 $\sigma(t_o)$ 가 시그모이드 함수를 거쳐 0.5값이 되고, 이는 결국 b_x, b_y 가 그리드 셀 중심에 위치하게 된다. 이를 통해 예측되는 bounding box 좌표가 cell 중심을 너무 벗어나지 않도록 고려된다.

$$b_w = p_w e^{t_w} \dots\dots\dots (6)$$

$$b_h = p_h e^{t_h} \dots\dots\dots (7)$$

p_w, p_h 는 anchor box size인 너비와 높이를 의미한다. 학습 과정에서 t_w, t_h 이 0에 가까운 값이 되려고 하며, 0인 경우 e^{t_w}, e^{t_h} 는 1이 되며 prior 값과 같아지게 된다.

3.1.5 Multi-Labels Classification

YOLOv3는 하나의 그리드 셀마다 객체가 있다는 가정하에 객체를 검출하고 binary cross-entropy loss를 통해 그 셀의 최종적인 class를 예측한다. 이 과정에서 softmax가 아닌 independent logistic classifier를 사용하기에 사람과 여자와 같은 계층적인 class 분류까지 가능했고 이는 class probabilities로 각각의 class마다 값을 갖는다. 이에 더하여, 최종 예측 class에 대한 confidence를 의미하는 objectness score(Po)를 통해 0과 1로 표현되며, 객체로 인식할 것인지 아닌지를 결정하게 된다. 이 모델은 microsoft common object in context(MS COCO) dataset을 통해 학습되었다. 이 데이터는 컴퓨터 비전 학습을 위해 만들어진 여러 일상 이미지들의 집합이며 객체별로 세분화되어 라벨링 되어있다.

이 모델은 80개의 class를 갖는 MS COCO dataset을 통해 학습되었으므로 80개의 class probabilities를 갖게 된다. 여기에 그리드 셀 하나당 3개의 anchor box를 갖고 각각의 box는 Bounding box 좌표 4개를 가지므로 정리하자면 box 좌표 4, object score 1, class probabilities 80이므로 output인 flatten 결과는 $N * N * [(4 + 1 + 80) * 3]$ 를 갖는다. 여기서 N은 이전 절에서 설명한 서로 다른 크기의 3개 피쳐맵을 의미하는 것으로, YOLOv3는 13*13, 26*26, 52*52 형태를 갖는다. 이러한 과정을 거친 후 ground truth 당 하나의 bounding box만 남게 된다. 이를 설명하고 있는 그림 3.8에서 설명하고 있다.

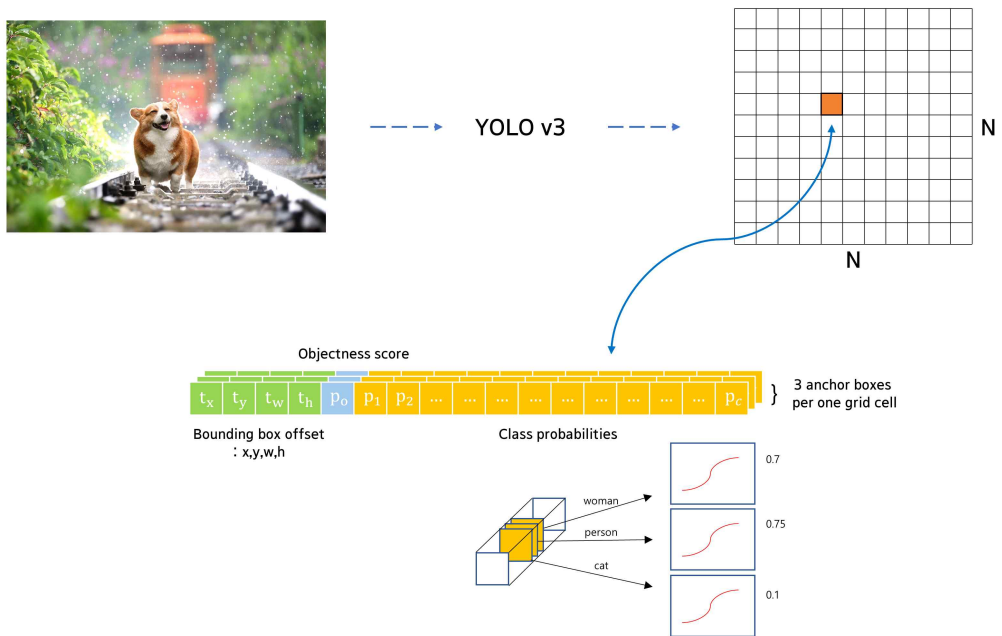


그림 3.8 YOLOv3 Multi-labels Classification

3.1.6 Non-Maximum Suppression

객체를 object detection을 하게 되면 객체가 있을 법한 곳을 탐색하게 되고 그 결과 각각의 객체마다 여러 개의 탐지결과(bounding box)가 검출되게 되는데 이 중에서 confidence score가 낮은 box는 제거하고 가장 적합한 box 하나를 선택해야 한다. 이를 위해 NMS(Non-Maximum Suppression)개념이 사용된다. NMS는 confidence score

와 IoU threshold가 복합적으로 사용하여 최적 box를 선택한다. 그 수행과정을 살펴보면 먼저 bounding box별로 특정 confidence score 값을 만족하지 못하는 bounding box를 제거한 다음 가장 높은 score를 기준으로 내림차순 정리하여 현재 예측 클래스와 같은 예측 box 중에서 낮은 score를 갖는 box를 제거한다. 여기서 가장 높은 score를 갖지는 않지만, 현재 예측 클래스와 다른 클래스를 갖는 객체에 있어서 가장 높은 score를 갖는 box가 제거되지 않기 위해 IoU 개념을 적용시켜 최종적으로 개별객체마다 가장 높은 score를 갖는 박스만 존재하게 한다. 이를 그림 3.9에서 보여주고 있다. 이러한 NMS를 통해서 YOLO는 여러 anchor box를 이용해 객체를 탐지한 다음 최종적으로 객체당 하나의 bounding box 결과를 보여줄 수 있게 된다.

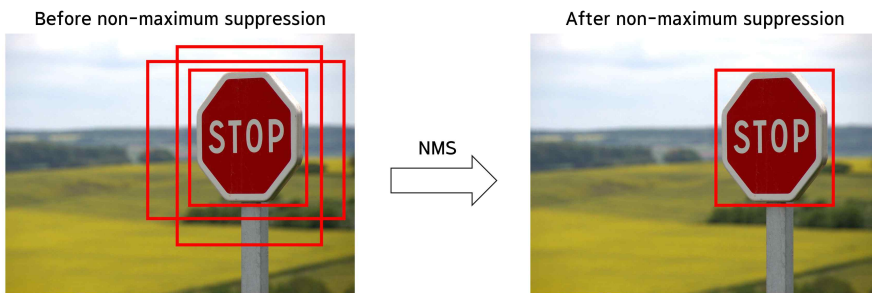


그림 3.9 NMS 결과 예시

3.1.7 Feature Extractor Network

본 연구에서 사용하는 YOLOv3는 DarkNet-53이라는 network를 사용한다. 이는 53개의 convolutional neural network으로 구성되었고 이전 버전인 YOLOv2에서 사용된 Darknet-19에 residual network를 추가함으로써 YOLO 저자가 과대적합(Over fitting) 문제를 새롭게 보완한 백본(Backbone) 네트워크이다. 그 결과, ResNet-101보다 1.5배 빠르고, ResNet보다 성능은 비슷하지만 2배 이상 빠른 검출결과를 보여주었다. 학습 이미지가 이러한 백본을 거침으로써 이미지에 존재하는 객체에 대해 특징을 파악하게 되고 객체를 검출할 수 있게 되는 이러한 과정을 Feature Extraction이라고 한다.

YOLOv3 Network Architecture

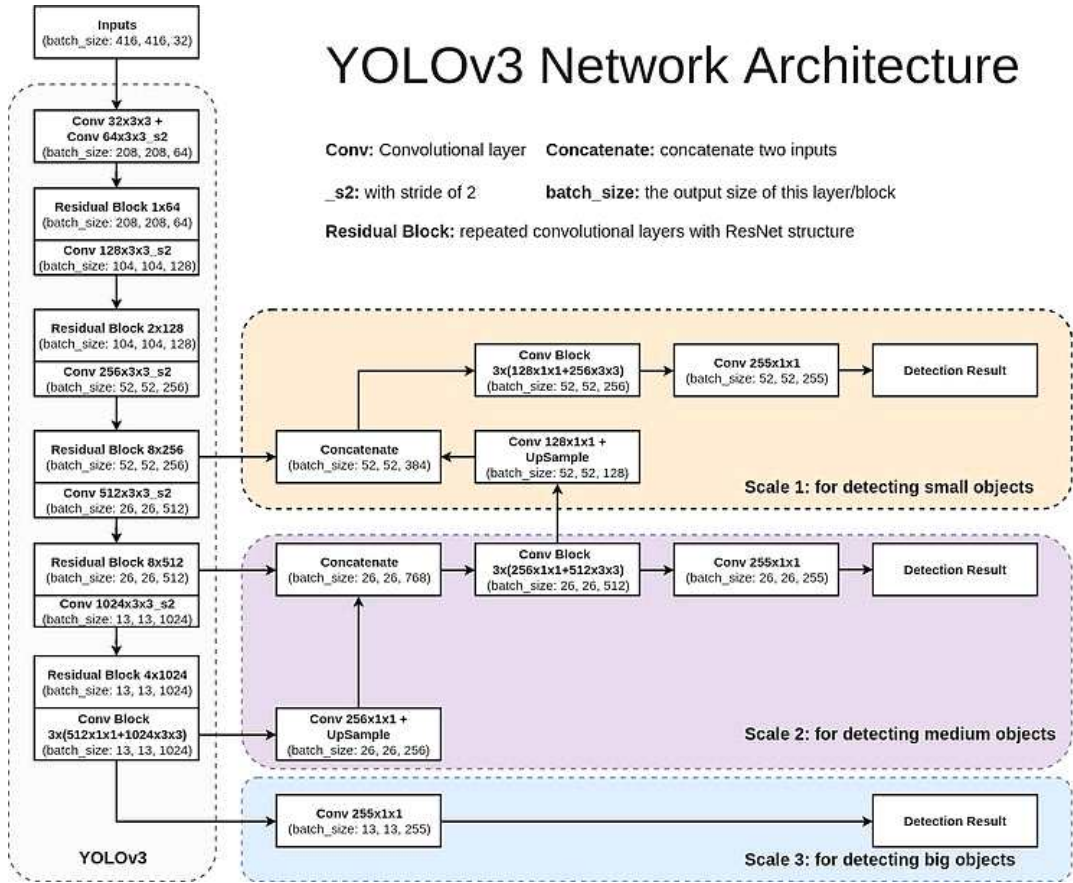


그림 3.10 YOLOv3 Architecture (Rosina De Palma, 2020, YOLOv3 Architecture: Best Model in Object Detection, Best in Australia, <https://bestinau.com.au/yolov3-architecture-best-model-in-object-detection>)

그림 3.10은 YOLOv3의 전체적인 architecture를 잘 설명하고 있으며, YOLOv3가 서로 다른 3개의 scale을 사용한 upsampling 기법으로 객체검출 성능을 높이려고 하는 것을 확인할 수 있다. 여기서 말하는 scale은 Multi-Labels Classification에서 설명한 13*13, 26*26, 52*52 형태를 의미한다.

3.2 전이학습

전이학습이란 YOLOv3에서 사용되는 DarkNet53과 같이 기존에 학습된 pre-trained model을 자신의 data에 이어서 학습함으로써 자신의 data에 대한 특징을 추출하는 새로운 network를 얻는 것을 의미한다. 이는 특징을 추출하는 CNN 전체를 다 학습시키지 않고 사전에 구축된 레이어를 통해 얻어진 weight를 활용하여 특정 계층에만 새로운 target 데이터를 학습함으로써 높은 정확도를 갖는 네트워크를 비교적 짧은 시간에 완성할 수 있는 방법으로 알려져있다. 이러한 특징 때문에 전이학습은 학습을 위한 충분한 데이터 셋을 구축하기 어려운 환경 속에서도 개인의 데이터 셋을 위한 객체검출 모델을 구축할 수 있다. 또한, 개인의 데이터 셋으로만 학습할 경우 모델이 편향될 수 있으므로, 검증된 데이터 셋을 통해 구축된 pre-trained 모델을 사용하는 것이 권장되고 있다.

성공적인 전이학습을 위해서는 자신이 가지고 있는 데이터의 특성을 파악하여 그에 맞게 fine-tuning을 해야 한다. 전이학습을 위해 사용하는 기존 CNN 레이어 중에서 이미지의 추상적인 특징을 추출하는 앞부분 레이어는 보통 건들이지 않고 그대로 사용하는데 이러한 레이어를 freeze한다고 표현하고 fine-tuning이란 전체 레이어 중에서 얼마나 freeze하여 학습할지 정하는 과정을 의미이다. Yoshinski et al.(2014)는 데이터의 양과 기존 CNN을 학습할 때 사용했던 데이터와의 유사성에 따라 사전학습 모델을 재정의함에 있어서 fine-tuning 방법을 4가지 경우로 정리해놓았다. 먼저 데이터 양이 많고 pre-trained된 모델의 데이터와 유사성 또한 적은 데이터를 갖는 경우는 모델을 다시 처음부터 학습시키는 것을 추천했다. 두 번째로 데이터 양이 많고 유사성도 큰 데이터를 갖는다면 가장 좋은 경우이며, 이 경우 본인이 전체를 다시 학습시키거나 전이학습을 하거나 선택할 수 있다. 다음은 데이터 양이 적지만 비슷한 특성을 갖는 경우는 마지막 fully connected layer(FC layer)만 새로 학습시킨다. 마지막으로 데이터 양도 적고 유사성도 적은 경우가 제일 문제인 경우인데, 이럴 경우 어떠한 방법으로 학습이 적절하게 되지 않는다고 평가했다. 정리하자면, 전이학습은 학습데이터의 양과 성질에 따라 사용 방법이 구분되어 있으며 이를 잘 활용한다면 효과적으로 모델을 완성시킬 수 있는 방법이다. 본 연구는 3번째 유형에 속하는 경우로 전체 학습 층에서 말단 부분만을 도로시실물 데이터에 대하여 새로 학습시키는 방법을 사용하였다.

제 4장 실험

4.1 실험데이터

연구를 진행하기 위해 한국지능정보사회진흥원 AIHub 사이트에서 제공하는 도로환경 파노라마 이미지 AI 데이터를 사용했다. 이 데이터는 자동인지를 위한 학습을 위해 서울시 주요도로 3400km를 주행하며 국내 도로환경 정적객체 189종에 대해 2,711,280 이미지를 구성한 데이터이다. 이 중에서 객체가 존재하는 14,184개 이미지를 전이학습에 사용하여 도로시설물을 검출하는 알고리즘을 완성할 수 있었다. 각각의 이미지는 이미지 내에 존재하는 도로시설물 객체 정보를 별도의 txt 형태로 정리한 설명 파일과 1:1쌍을 이루고 있으며, 이를 annotation이라고 한다. 이는 .csv, .json, .xml 파일과 같은 형식으로 주로 사용되며, 그림 4.1은 예시 이미지, 해당 예시 이미지의 annotation 파일, 이 둘을 매칭한 결과 사진을 시각화하여 보여주고 있다.



그림 4.1 Annotation 매칭 예시 (a)원본이미지, (b)annotation, (c)매칭결과 시각화

그림 4.1(b)에서 보여주는 annotation 파일이 Object detection 모델을 학습시키기 위해 객체에 대한 bounding box 좌표정보와 그에 해당하는 객체 class 정보를 포함하는 것을 확인할 수 있으며, 이는 이해를 위해 .json 구조를 갖는 원본 형태에서 수정된 구

조로 표현했다.

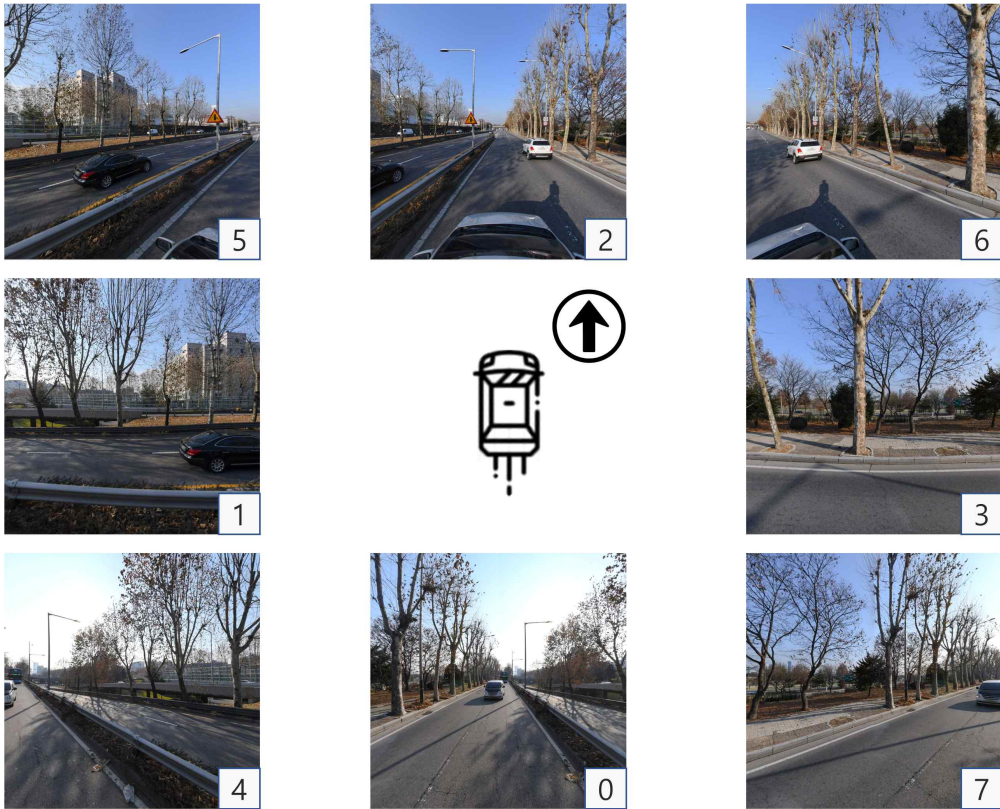


그림 4.2 도로환경 사진 예시

그림 4.2는 도로환경 파노라마 이미지 AI 데이터에 구축된 파노라마 사진을 8개의 평면사진으로 나눈 모습을 보여주고 있다. 이러한 데이터는 34만건 가량의 파노라마 사진을 왜곡보정 후 약 270만건의 평면사진으로 분할하여 생성되었으며 기존 사진 제목에 숫자 0부터 7까지 번호를 부여함으로써 사진을 방향별로 나누어 확인할 수 있다. 본 연구는 기존 객체에서 90° 회전 상태를 의미하는 1번, 3번 방향은 학습에 영향을 줄 수 있다고 판단하여 제외하고 나머지 방향의 이미지를 사용했다. 하지만, 전체 사진을 전부 학습시키기에는 무리가 있어 이들 중에서 객체가 존재하는 14,184개의 이미지를 무작위로 선택하여 학습을 진행하였다.

표 4.1 클래스 목록

분류	클래스명	Class code
교통안전표지판	정차 · 주차금지	No-Parking-or-Stopping
	주차금지	No-Parking
	최고속도제한	Maximum-Speed-Limit
	견인지역	Towing-Zone
도로노면표시	속도제한	Speed-Limit
	우회전	Right-Turn
	좌회전	Left-Turn
	직진	Straight
	직진 및 우회전	Straight-or-Right
	횡단보도	Crosswalk
	횡단보도예고	Pedestrian-Crossing-Ahead
신호등	신호등	Traffic-Light

연구에서 사용한 데이터는 기존 국토지리정보원의 정밀도로구축을 참조하여 189종의 정적 객체 class를 갖는 데이터였지만 도로환경 특성상 객체별 분포되어있는 객체 수 차이가 컸으며, 학습에 충분한 클래스만을 사용하기 위해 전체 평균을 넘는 상위 12개를 선택한 결과를 표 4.1에서 보여주고 있다. 데이터는 표지판, 도로노면표시, 신호등과 같이 분류되어 있었으며 객체검출결과는 영문으로 시각화할 수 있기에 한글 클래스명을 영문으로 변경하였다. 신호등 클래스의 경우 보행등, 차량횡형, 차량종형 등 16개로 세분화된 모습이지만 이 중에서 가장 많이 존재하는 신호등 객체인 삼색등, 사색등A 두 개의 클래스를 하나의 클래스로 통일해 사용했다. Oh et al.(2020)에 의하면 정밀도로지도 객체 활용성 검증 연구를 통해 세세하게 분할된 신호등 유형이 불필요하다는 의견이 있었기에 이를 참조했다. 정리하자면 본 연구는 12종 class를 검출하도록 학습한 모델, 교통안전표지판에 해당하는 4종 class를 검출하는 모델, 도로노면표시에 해당

하는 7종 class를 검출하는 모델 3개를 만들고 이들의 성능을 비교했다. 이러한 모델을 학습하기 위해 각각의 모델은 train 60%, validation 20%, test 20% 비율로 나누어 학습했으며 표 4.2에서 데이터셋 분류결과를 보여주고 있다.

표 4.2 데이터셋 분류

	The Number of Images		
	전체 12종 class	교통안전표지판 4종 class	도로노면표시 7종 class
train	9,079	2,034	5,792
validation	2,269	508	1,488
test	2,836	635	1,600
총 합계	14,184	3,177	8,880

전체 12종 class를 검출하는 모델 성능 평가에 사용된 test 이미지는 2,836개이며, 이미지 안에 존재하는 객체 수 분포는 그림 4.3과 같다. 본 연구는 이미지에서 단일 객체만을 검출하는 모델이 아닌 다중 객체를 검출하도록 했기 때문에 하나의 이미지에 여러 개의 객체가 무작위로 존재하는 도로특성 상 객체 수 분포를 균일하게 맞출 수 없었다.

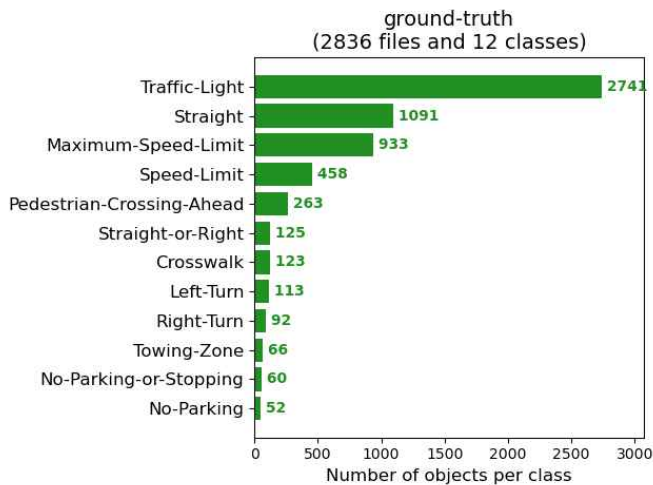
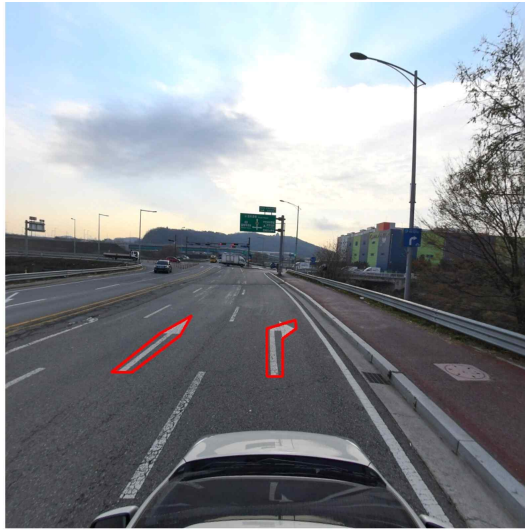


그림 4.3 클래스 객체별 분포



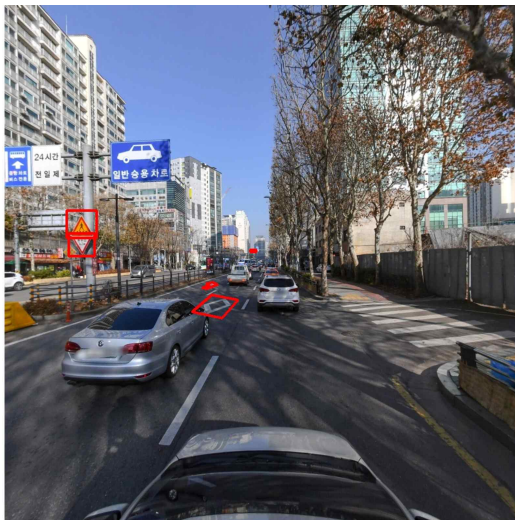
(a)



(b)

그림 4.4 Annotation 수정 예시 (a)변경 전, (b)변경 후

이미지 내에 존재하는 객체 정보를 담은 annotation 파일에서 도로노면표시에 해당하는 화살표 그림은 다각형 좌표로 저장되어 있었다. 하지만, YOLOv3는 사각형 bounding box를 필요로 하기 때문에 사각형 좌표로 변환해야 했으며, 그 결과를 그림 4.4에서 보여주고 있다.



(a)



(b)

그림 4.5 학습 이미지 예시1 (a)변경 전, (b)변경 후

그림 4.5는 클래스 및 좌표 전처리 후 학습에 사용될 이미지 예시를 보여주고 있다. 먼저 도로노면표시 중 횡단보도예고 표시를 윗변과 아랫변이 평행한 사각형 형태로 변환하였으며, 사진 좌측에 보이는 표지판 2개는 앞서 언급한 상위 12개 클래스에 속하지 않기 때문에 제외됨을 확인할 수 있다. 또한, 사진 중앙 부분에 신호등과 좌회전 노면 표시가 작게 위치하는 것을 볼 수 있다. 그림 4.6에서는 사람이 육안으로 보아도 쉽게 확인할 수 없는 신호등이 사진 중앙 부분에 작지만 3개 존재하고 있다고 Annotation 파일에 표기되어 있음을 보아 객체검출실험 결과 낮은 Recall 값을 가질 것임을 유추할 수 있다.

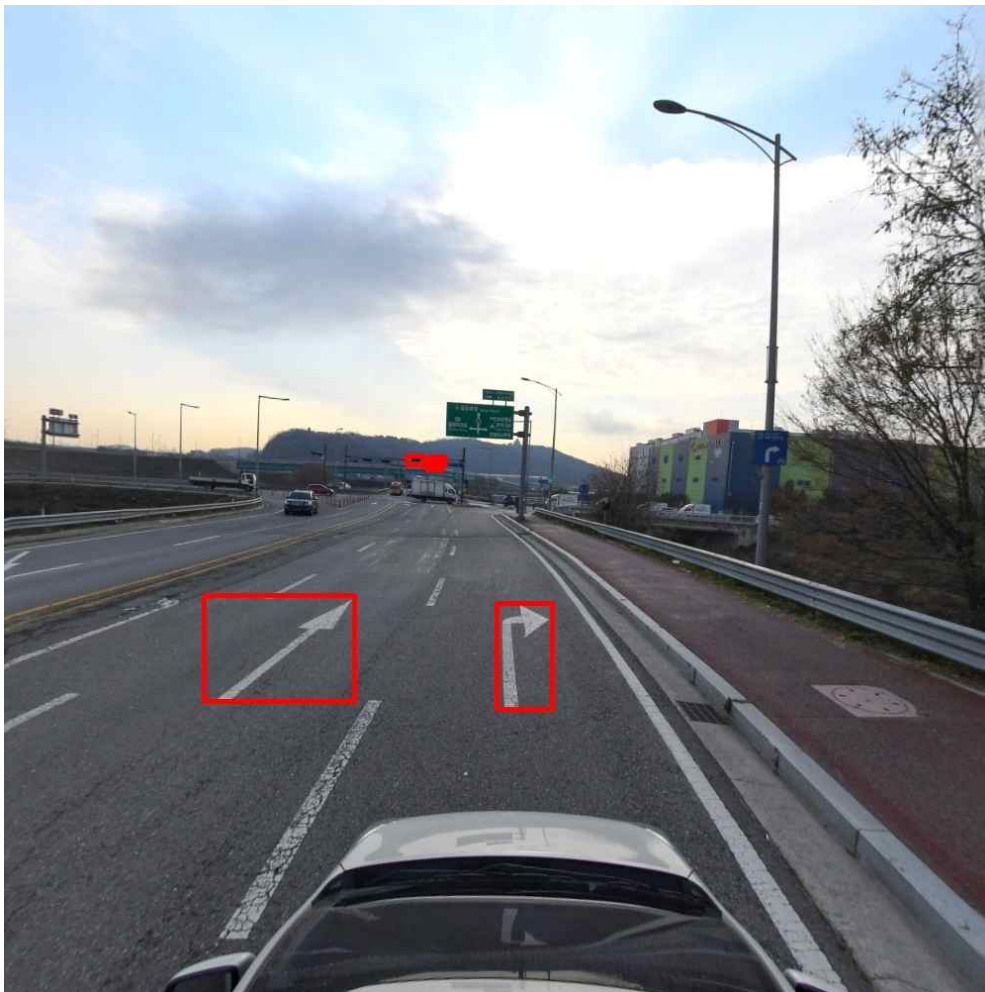


그림 4.6 학습 이미지 예시2

4.2 실험 환경

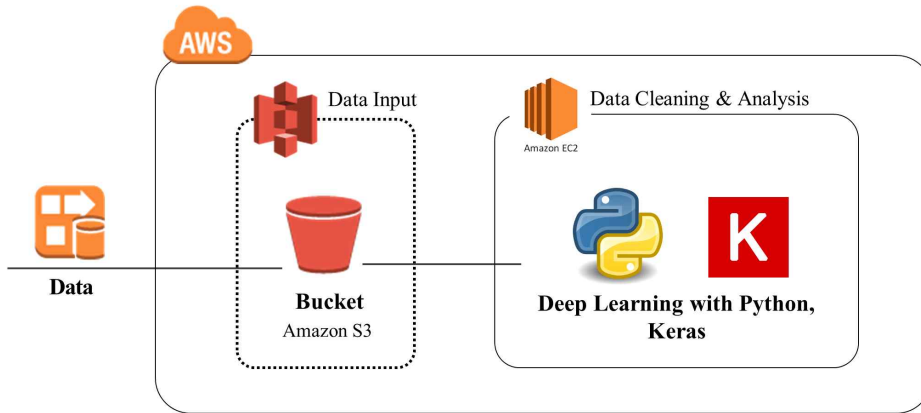


그림 4.7 시스템 아키텍처

로컬환경에 대한 한계를 극복하기 위해 사용한 클라우드 컴퓨팅 시스템은 Amazon Web Service(AWS)이다. 데이터를 AWS의 저장공간인 S3 Bucket에 업로드하고 데이터 전처리 및 분석을 위해 AWS EC2에 Python 환경을 구축하여 진행했다. 이를 그림 4.7에서 묘사하고 있다. EC2는 여러 리소스 자원을 선택할 수 있고 본 연구를 진행하기 위해 선택한 AWS instance 옵션은 p2.xlarge였으며 이에 해당하는 하드웨어 환경은 표 4.3과 같다. GPU를 사용하기 위해 CUDA version 10.0을 사용했으며, Keras를 이용하여 Deep Learning을 구현하였다.

표 4.3 AWS EC2 사용 사양

Component	Specification
CPU	2.5GHz
GPU	Tesla k80 (11441MiB)
RAM	61GiB
OS	Linux

4.3 실험 결과

도로시설물 12종 정적객체를 검출할 수 있게 해당 데이터를 YOLOv3 기존 base network에 전이학습 시켰다. 새로운 데이터에 학습된 YOLOv3는 학습에 사용되지 않은 일반도로 주행 이미지에서도 객체를 잘 검출하는 것을 확인할 수 있었다. 객체검출 결과는 [클래스명, confidence score, bounding box 좌상단 좌표, 우하단 좌표] 형태의 txt 결과로 반환되며, 이를 이미지와 매칭시켜 시각화된 결과는 그림 4.8과 같다. 이는 일반도로 주행 이미지에서 객체검출을 적용한 결과이며, 그림 4.8(a)을 보면 도로노면 표시에 해당하는 직진표시와 직진 및 우회전표시를 잘 검출한 모습을 확인할 수 있다. 그림 4.8(b)를 보면 삼색등 2개를 신호등으로 잘 검출한 모습을 확인할 수 있다.

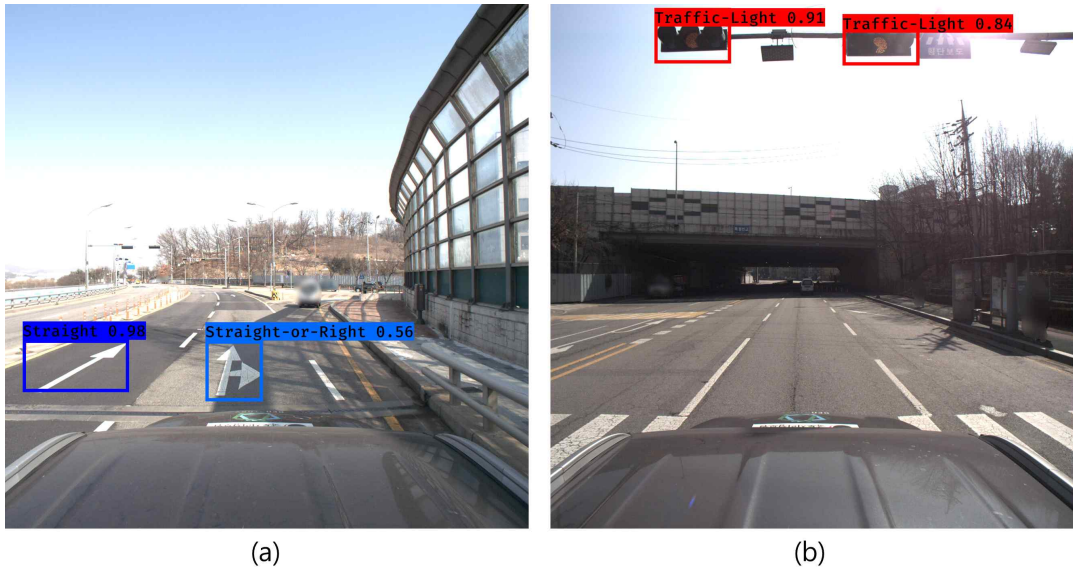


그림 4.8 객체검출결과

4.3.1 도로시설물 객체검출 성능 평가

앞 절에서 일반도로 주행 이미지를 이용하여 새롭게 학습한 모델이 도로시설물 객체를 적절하게 검출하는 것을 확인했다. 본 절에서는 최종적으로 최적의 성능을 갖는 결과를 도출하기 위해 표4.2에서 보여준 것과 같이 12종 class를 검출하도록 학습한 모델, 교통안전표지판에 해당하는 4종 class를 검출하는 모델, 도로노면표시에 해당하는 7종 class를 검출하는 모델에 해당하는 test 데이터셋을 각각 이용하여 객체검출성능을 AP_{50} 으로 수치화하고 이들을 비교했다.

먼저 도로노면, 표지판, 신호등을 한번에 검출하는 12종 class에 대한 모델에 대한 성능을 그림 4.9에서 확인할 수 있다. 그림 4.9(a)는 2,836개의 test 이미지 파일에 12개의 class가 ground-truth로 사용되었다는 것을 의미하며, class마다의 객체 수를 보여주고 있다. 그림 4.9(b)는 class마다의 AP와 이를 평균한 mAP를 보여주고 있다. AP_{50} 지표에 해당하는 mAP를 평가할 때 이미지에 객체가 몇 개 존재하는지와 객체 간 구별이 얼마나 어려운지에 따라서 평가 기준이 다를 수 있지만, 일반적으로 50% 이상은 보통 이상, 60% 이상은 뛰어남, 70% 이상은 매우 좋음으로 평가된다(Everingham et al., 2010). 결론적으로 본 연구가 목표한 도로시설물 객체검출모델이 56.56%의 값으로 보통 이상의 성능을 가졌다고 평가할 수 있다.

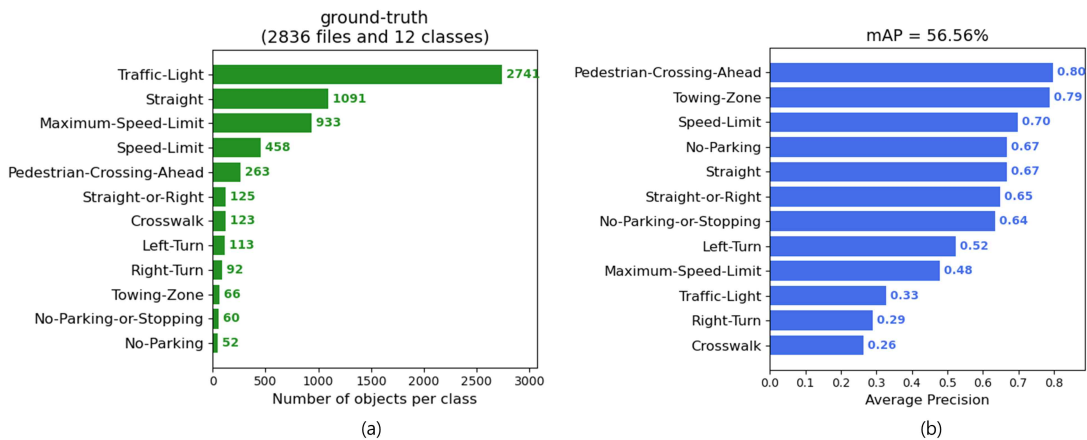


그림 4.9 도로시설물 객체검출 성능

그림 4.9(b)에서 class마다의 AP를 살펴보면 대부분의 도로시설물 객체를 잘 검출했

지만, 특정 도로노면과 신호등에 해당하는 객체에 있어서 낮은 성능을 가짐을 확인할 수 있는데, 이는 그림 4.6에서 언급한 것과 같이 사람이 육안으로도 확인하기 어려울 정도로 작은 객체까지 주석처리 되었기 때문에 검출 성능이 떨어지는 경우가 대부분이었다. 특히 도로노면표시는 원근법에 영향을 받는 객체이므로 사진을 촬영한 거리가 가깝지 않다면 형태를 인식하기 어렵기 때문에 검출 성능이 표지판에 비해 낮음을 확인했다. 이러한 이유로 사진이 아닌 영상으로 도로시설물을 검출한다면 그 성능이 더 좋을 것으로 판단되어 영상을 이용한 실험을 진행했다.

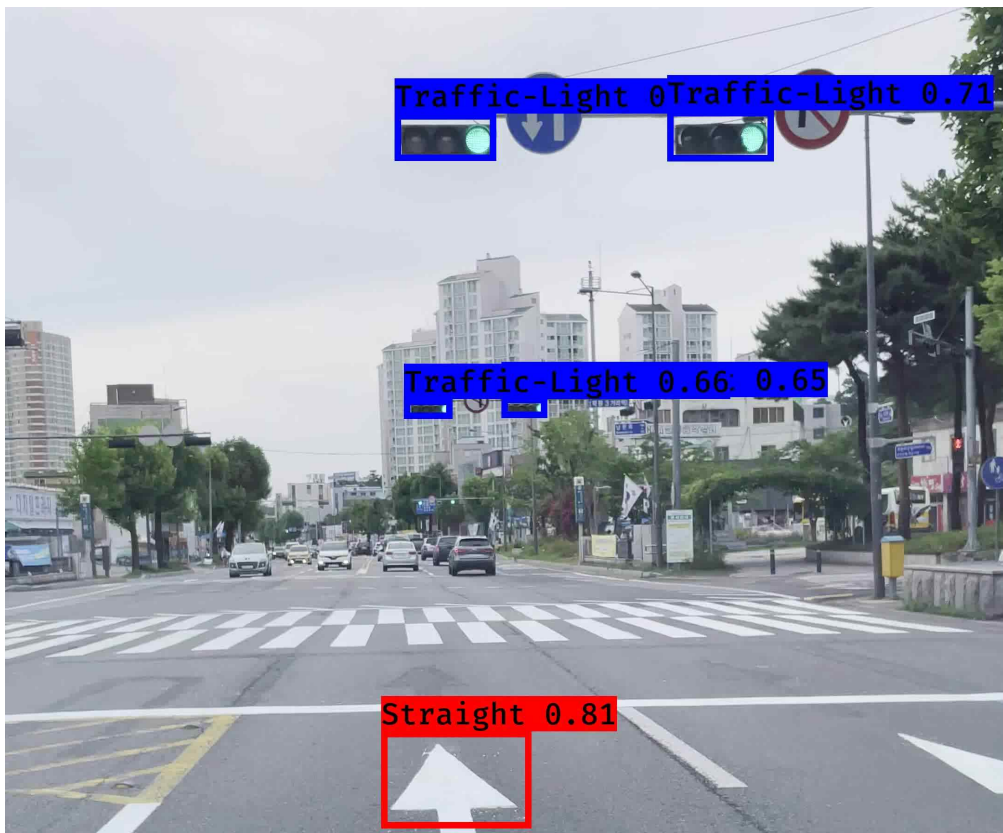


그림 4.10 도로주행 영상에서의 객체검출속도 평가 예시

그림 4.10은 광주광역시 소태역 출발 - 남광주역 도착 방향인 도로주행 동영상에서 캡처된 한 장의 프레임을 보여주고 있으며 이러한 동영상을 이용하여 도로시설물 객체 검출속도를 확인한 결과 1초당 3장의 이미지를 처리할 수 있었다. 본 연구에서는 Tesla K라는 GPU와 함께 Python이라는 언어환경에서 객체를 검출했기 때문에 부족

한 속도였지만, 기존 YOLOv3가 수행된 Titan X라는 GPU와 C 언어환경에서 객체를 검출한다면 초당 30장의 이미지를 처리할 것으로 예측된다. 또한, 해당 영상에 존재하는 14개의 신호등에서 14개를 검출하여 100%의 정확도를 보였고, 21개의 도로노면표시에서 19개를 검출하여 90%, 5개의 표지판에서 5개를 검출하여 100% 정확도를 가짐을 확인하였다. 이렇듯 사진이 아닌 영상을 이용하여 도로시설물 객체를 검출한다면 차량이 도로를 주행하기 때문에 도로시설물 객체와 가까이 위치하는 순간이 촬영되고 객체를 분명하게 식별할 수 있기에 사진을 통한 검출 성능과 비교하여 향상된 결과를 얻을 수 있다.

다음은 도로노면, 표지판, 신호등을 한번에 검출하는 모델이 아닌 7종의 도로노면표시 class만을 검출하는 모델을 새롭게 학습시켰을 때 성능을 알아보려고 했다. 그림 4.11(a)는 1,600개의 test 이미지 파일에 7개의 class가 ground-truth로 사용되었다는 것을 의미하며, class마다의 객체 수를 보여주고 있다. 7개 class는 순서대로 직진(Straight), 속도제한(Speed-Limit), 횡단보도예고(Pedestrian-Crossing-Ahead), 좌회전(Left-Turn), 우회전(Right-Turn), 직진 및 우회전(Straight-or-Right), 횡단보도(Crosswalk)를 의미한다. 그림 4.11(b)는 class마다의 AP와 이를 평균한 mAP를 보여주고 있다. 결론적으로, 도로노면만을 검출하는 모델을 구성할 경우 63.04%의 뛰어난 성능으로 검출 성능이 향상되는 것을 확인했다.

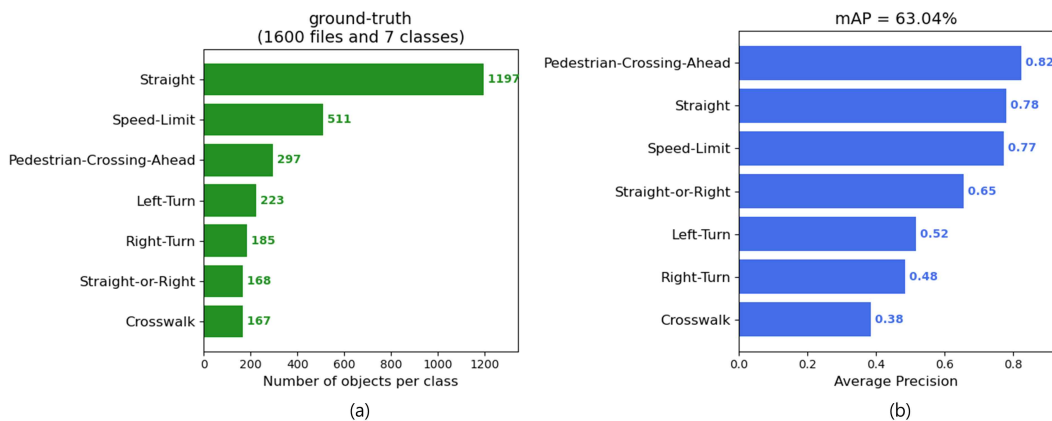


그림 4.11 도로노면 객체검출 성능

마지막으로 4종의 표지판 class만을 검출하는 모델을 새롭게 학습시켰을 때 성능을 알아보고자 했다. 그림 4.12(a)는 635개의 test 이미지 파일에 4개의 class가 ground-truth로 사용되었다는 것을 의미하며, class마다의 객체 수를 보여주고 있다. 4개의 class는 정차·주차금지(No-Parking-or-Stopping), 견인지역(Towing-Zone), 최고속도제한(Maximum-Speed-Limit), 주차금지(No-Parking)이다. 그림 4.12(b)는 표지판을 검출하는 성능이 72%의 매우 좋은 성능을 가짐을 보여주고 있으며, 표지판만을 검출하는 모델을 구성할 경우 성능이 더 향상되는 것을 확인하였다.

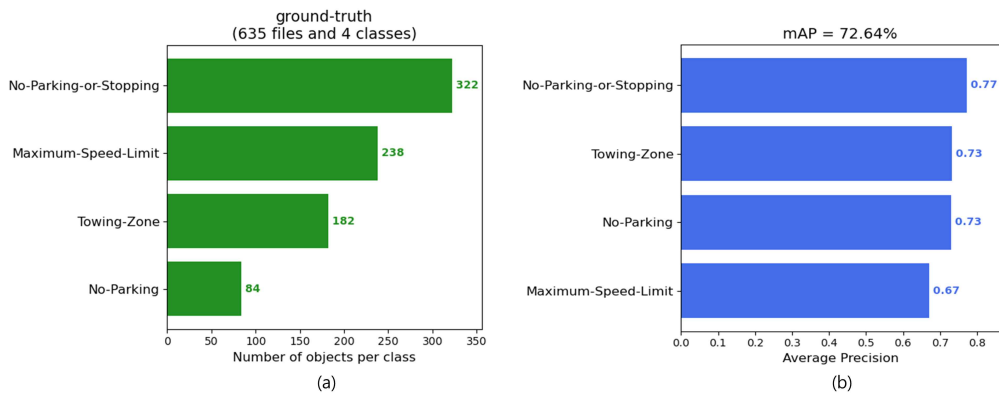


그림 4.12 표지판 객체검출 성능

국내 도로환경 정적객체 189종은 4개의 범주로 구분되고 신호등 16종, 도로노면표시 35종, 표지판 133종, 이동객체 5개로 이루어져 있다. 본 연구는 이동객체를 제외한 3개의 범주를 검출하고자 이에 해당하는 12종 객체를 실험군으로 설정했다. 이와 더불어 검출 성능을 높이고자 범주별로 모델을 각각 구축하여 검출 성능을 비교해보았다. 그 결과 전체 범주를 검출하고자 하는 모델도 영상기반으로 사용하게 된다면 좋은 성능을 기대할 수 있지만, 범주별로 나누어 객체검출을 수행한다면 더 향상된 정확도를 갖는 결과를 기대할 수 있음을 확인했다.

4.3.2 도로시설물 변화탐지 결과

본 연구의 목적인 도로시설물에 대한 변화탐지를 수행하기 위해서는 두 차례에 걸친 도로주행 영상이 필요하다. 하지만, 사용된 데이터는 특정 구간을 한 차례 주행한 결과이므로 기존 도로주행 이미지에 수작업으로 신규 객체 삽입, 객체 제거, 객체 클래스 변경과 같은 3가지 경우로 변화를 줌으로써 실험을 진행했다. 이와 같은 변화는 Adobe Photoshop을 이용하여 디자인했으며, 학습에 사용되지 않았으며 같은 해상도를 갖는 이미지에서 추출한 도로시설물을 사용했다.

A) 도로시설물 신규 객체 삽입

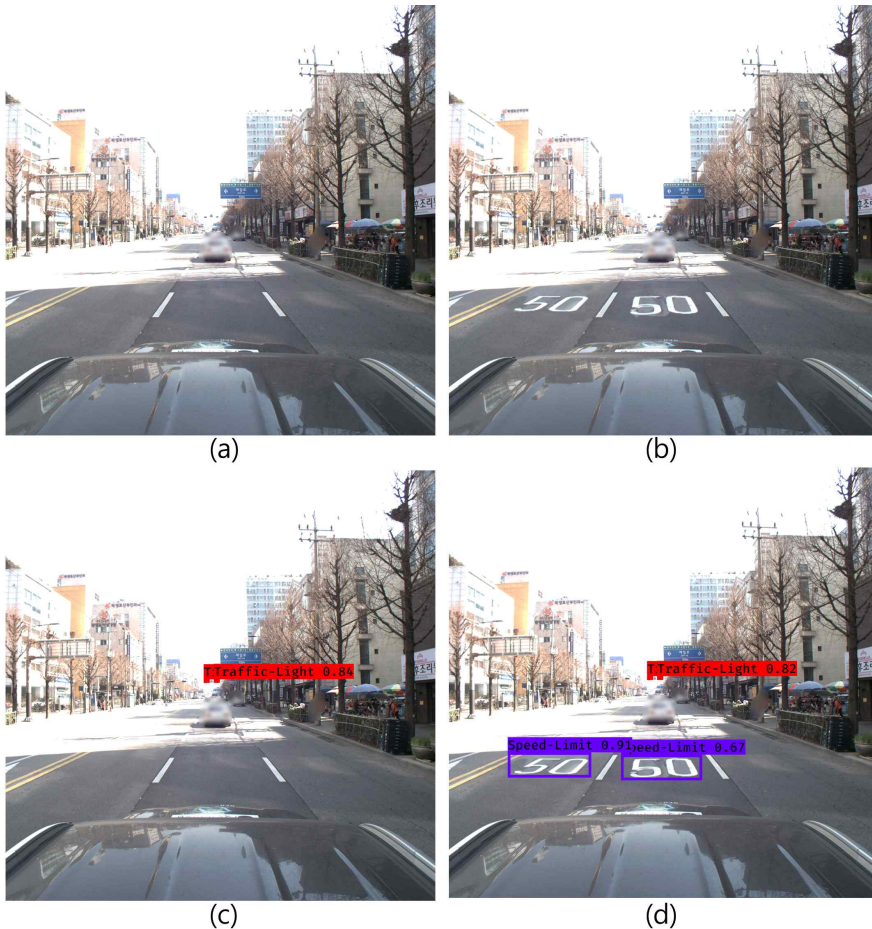
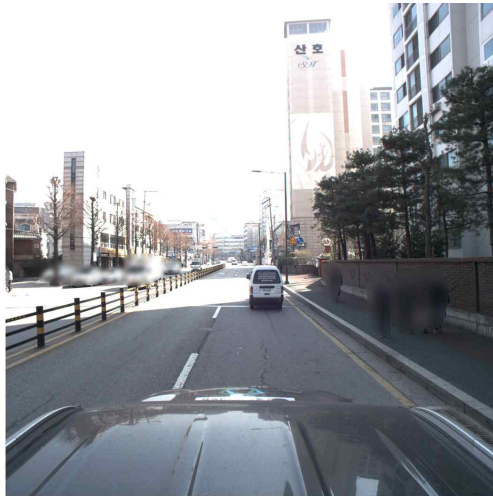
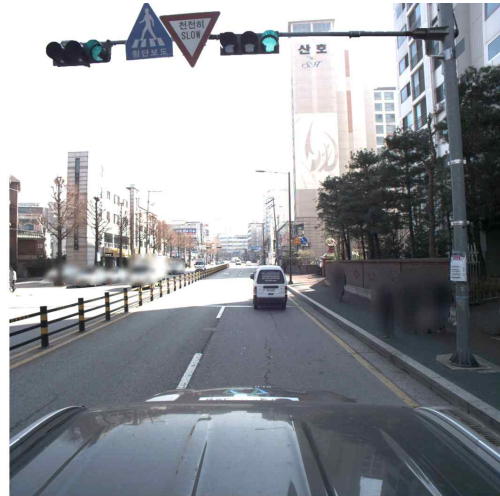


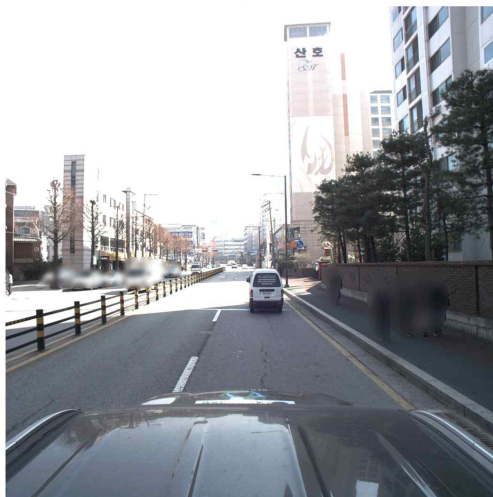
그림 4.13 신규 도로시설물 객체 검출 예시1 (a) 원본이미지, (b) 객체삽입 이미지, (c)원본이미지 객체검출결과, (d) 객체삽입 이미지 객체검출결과



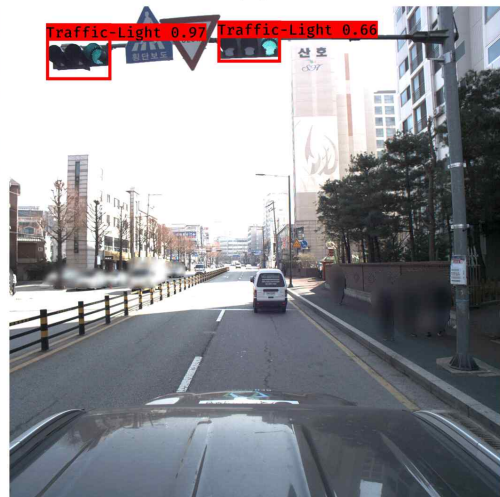
(a)



(b)



(c)



(d)

그림 4.14 신규 도로시설물 객체 검출 예시2 (a) 원본이미지, (b) 객체삽입 이미지, (c)원본이미지 객체검출결과, (d) 객체삽입 이미지 객체검출결과

먼저 그림 4.13(a)는 기존 이미지, 4.13(b)는 새로운 객체가 삽입된 이미지, 4.13(c)는 기존 이미지의 객체검출결과, 4.13(d)는 새로운 객체가 삽입된 이미지의 객체검출결과를 순서대로 나타내고 있으며 변화탐지 예시 그림의 순서는 모두 동일하다. 그림 4.13은 기존 이미지에는 도로노면표시가 없었던 경우에 새로운 노면표시가 생성되었을 경우를 보여주고 있으며, 그림 4.14에서는 신호등이 새롭게 생겼을 때를 보여주고 있다. 이를 통해 새로운 객체가 생성될 경우 잘 검출할 수 있음을 확인했다.

B) 도로시설물 객체 제거



그림 4.15 제거된 도로시설물 객체 검출 예시1 (a) 원본이미지, (b) 객체제거 이미지, (c)원본이미지 객체검출결과, (d) 객체제거 이미지 객체검출결과



(a)



(b)



(c)



(d)

그림 4.16 제거된 도로시설물 객체 검출 예시2 (a) 원본이미지, (b) 객체제거 이미지, (c)원본이미지 객체검출결과, (d) 객체제거 이미지 객체검출결과

먼저 그림 4.15는 기존 이미지에는 속도제한표지판이 있었지만 새로운 이미지에는 표지판이 제거되었을 경우를 보여주고 있으며, 그림 4.16은 기존 이미지에서는 도로노면표시 중 횡단보도예고표시가 있었지만 이가 제거된 경우를 보여주고 있다. 두 개의 그림과 같이 도로시설물 객체가 제거되었을 때 역시 잘 검출하는 것을 확인했다.

C) 도로시설물 객체 클래스 변경



그림 4.17. 변경된 도로시설물 객체 검출 예시1 (a) 원본이미지, (b) 객체변경 이미지, (c)원본이미지 객체검출결과, (d) 객체변경 이미지 객체검출결과



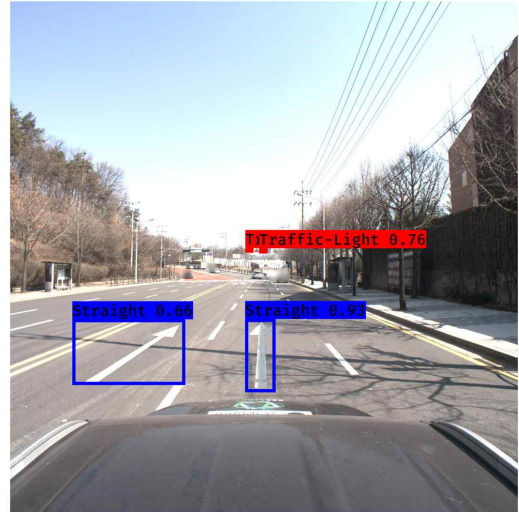
(a)



(b)



(c)



(d)

그림 4.18. 변경된 도로시설물 객체 검출 예시2 (a) 원본이미지, (b) 객체변경 이미지, (c)원본이미지 객체검출결과, (d) 객체변경 이미지 객체검출결과

먼저 그림 4.17은 속도제한표지판이 존재했던 경우에서 주정차금지 표지판으로 변경되었을 경우를 나타내고 있으며, 그림 4.18은 횡단보도예고표시에서 직진표시로 변경되었을 때를 예시로 보여주고 있다. 이 둘 모두 새로운 객체로 변경되었을 경우를 나타내며 이 역시 잘 검출하는 것을 확인할 수 있다.

제 5장 결론

본 연구의 목적은 앞으로 전국 단위로 구축될 정밀도로지도를 효율적으로 갱신을 하고자 저비용의 카메라 이미지를 통해 빠르게 변화탐지를 수행하는 것이다. 이러한 연구목적 달성을 위해 도로 주행 영상에서 정밀도로지도 내에 존재하는 객체를 검출할 수 있도록 딥러닝 기반 객체검출 모델을 구성했다. 이 모델을 통하여 이미지에 존재하는 객체를 txt 형식으로 나타내 기존과 달라진 점이 있는지 파악할 수 있게 하였다.

본 연구의 결과는 다음과 같다.

1. 도로주행 이미지에서 객체를 검출하기 위해 YOLOv3 알고리즘에 12종류의 도로 시설물이 포함된 이미지를 전이학습한 결과 mAP가 56.56 AP_{50} 이었다. 여기서 객체범주로 나누어 7종류의 도로노면에 대해서 학습시킨 결과 mAP가 63.04 AP_{50} 이었으며, 4종류의 표지판에 대하여 학습시킨 결과 mAP는 72.64 AP_{50} 이었다. 또한, 이미지가 아닌 영상으로 도로시설물 객체를 검출하였을 때 도로시설물 객체와 가까이 위치하는 순간이 촬영되어 객체를 분명하게 식별하기에 사진으로 검출한 성능보다 향상된 결과를 얻을 수 있음을 확인했다. 이러한 성능 갖는 도로시설물 검출 모델을 도로환경 변화탐지에 적용한다면 적합할 것이라 평가된다.

2. 변화유무를 파악하기 위해서는 같은 구간을 여러 기간에 걸쳐 촬영한 도로 주행 영상에 딥러닝 기반 객체검출기법을 적용하여 변화된 부분을 탐지해야 하지만 본 연구는 한 차례 촬영이 이루어진 이미지만 가지고 있었다. 이러한 이유로 변화탐지에 대해서 정확도를 수치화할 수 없었지만, Adobe Photoshop으로 객체를 합성하여 변화를 감지할 수 있는지 확인했다. 변화 유형은 새롭게 객체가 생성되는 경우, 기존에 존재했던 객체가 제거된 경우, 기존에 존재했던 객체가 다른 종류의 객체로 변경된 경우와 같이 3가지로 구분하여 진행하였으며 그 결과 변화를 잘 탐지했다.

본 연구는 현재 정밀도로지도 갱신작업에 있어서 수작업으로 진행되는 변화유무 파

악 과정을 자동으로 탐지하여 알려주고 빠른 부분수정을 가능하게 하도록 하는 의도를 가지며 도로시설물 객체검출 모델을 구축하여 성능을 비교하였다. 하지만, 100%의 정확도를 갖는 검출결과는 아니기에 변화를 제대로 감지하지 못하는 경우 사람의 확인이 필요하다는 한계점이 있다. 또한, 위치정보를 갖지 않는 이미지 데이터를 사용했기 때문에 변화유무를 파악하여도 이를 정밀도로지도에 갱신할 수 없다는 한계점이 존재했다. 향후 연구로 정밀도로지도의 도로시설물에 대한 Data Base 정보, POI 정보를 이용해 객체 단위로 변화탐지를 수행하고 이를 정밀도로지도에 갱신하는 연구를 하고자 한다. 더 나아가 YOLOv3 이후에 새롭게 개발되고 있는 알고리즘을 사용함으로써 도로시설물 객체를 검출하여 성능을 비교하는 연구를 진행할 예정이며, semi-supervised learning 혹은 open set 기반의 객체검출기법을 적용하여 시설물 DB에 존재하지 않은 객체를 검출하는 연구 또한 진행할 예정이다.

참고문헌

1. 김민정, 김성은 & 김건희 (2020). YOLO-v3 딥러닝 모델을 활용한 로고 검출 및 인식 알고리즘 개발. *정보 및 제어 논문집*, 65-66.
2. 김현숙, 권오천, 이승준, 김정숙, 김우진, 윤대섭 & 이인환 (2020). 레벨 3 자율주행차량의 인적요인 가이드라인 연구 동향.
3. 국토지리정보원 (2018). 정밀도로지도 연계 효율화 연구 및 구축·갱신 연구보고서.
4. 박유경, 강원평, 최지은 & 김병주 (2019). 자율주행 지원을 위한 정밀도로지도 갱신기술 평가를 위한 기준 도출 연구, *한국지리정보학회지*, 22(3), 146 - 154.
5. Alcantarilla, P. F., Stent, S., Ros, G., Arroyo, R., & Gherardi, R. (2018). Street-view change detection with deconvolutional networks. *Autonomous Robots*, 42(7), 1301-1322.
6. Chun P., Zhiguo Y., Xiaoming X., Mingxia S., Jie S., & Di W. (2013). Vehicle logo recognition based on deep learning architecture in video surveillance for intelligent traffic system. *IET International Conference on Smart and Sustainable City 2013*, 123-126.
7. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *In 2005 IEEE computer society conference on computer vision and pattern recognition*, 886-893.
8. Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303-338.
9. Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. *In 2008 IEEE conference on computer vision and pattern recognition*, 1-8.

10. Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, 1440-1448.
11. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 580-587.
12. Hata, A., & Wolf, D. (2014). Road marking detection using LIDAR reflective intensity data and its application to vehicle localization. In *17th International IEEE Conference on Intelligent Transportation Systems*, 584-589.
13. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904-1916.
14. Heo, M., Kim, J., & Kim, S. (2020) HD Map Change Detection with Cross-Domain Deep Metric Learning, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*.
15. Jinno, I., Sasaki, Y., & Mizoguchi, H. (2019). 3D Map Update in Human Environment Using Change Detection from LIDAR Equipped Mobile Robot. In *2019 IEEE/SICE International Symposium on System Integration*. 330-335.
16. Jo, K., Kim, C., & Sunwoo, M. (2018). Simultaneous localization and map change update for the high definition map-based autonomous driving car. *Sensors*, 18(9), 3145.
17. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2117-2125.
18. Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, 2980-2988.
19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In European conference on computer vision, 21-37.
20. Ma, W. C., Tartavull, I., Bărsan, I. A., Wang, S., Bai, M., Mattyus, G., & Urtasun, R. (2019). Exploiting sparse

- semantic HD maps for self-driving vehicle localization. *arXiv preprint arXiv:1908.03274*.
21. NGII. (2016). A study on the improvement of precision guidance for autonomous vehicles, National Geographic Information Institute, Research report, Republic of Korea, 27-90.
 22. NGII. (2019). Hd map construction manual.
 23. NGII. (2015). study on map construction with accuracy for supporting autonomous vehicles, National Geographic Information Institute, Research report, NGII, Republic of Korea, 23-93.
 24. Oh, J. M., Song, Y. H., Hong, S. P., Shin, Y. M., & Ko, Y. C. (2020). High Definition Road Map Object usability Verification for High Definition Road Map improvement. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 38(4), 375-382.
 25. Qin, R., & Gruen, A. (2014). 3D change detection at street level using mobile laser scanning point clouds and terrestrial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90, 23-35.
 26. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.
 27. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
 28. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.
 29. Revaud, J., Heo, M., Rezende, R. S., You, C., & Jeong, S. G. (2019). Did it change? learning to detect point-of-interest changes for proactive map updates. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4086-4095.
 30. Su, H., Xiatian Z., & Shaogang G. (2018). Open logo detection challenge. *arXiv preprint arXiv:1807.01964*.

31. Tüzkö, A., Herrmann, C., Manger, D., & Beyerer, J. (2017). Open set logo detection and retrieval. *arXiv preprint arXiv:1710.10891*.
32. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*.
33. Wang, W., Yu, R., Huang, Q., & Neumann, U. (2018). Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2569-2578.
34. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. *arXiv preprint arXiv:1411.1792*.