

# 차량 탐지와 차량 추적에 대한 연구

김세영<sup>1</sup>, 민재은<sup>2</sup>, 표세훈<sup>3</sup>, 최상일<sup>4</sup>

<sup>1</sup> 단국대학교 인공지능융합학과

<sup>2</sup> 단국대학교 인공지능융합학과

<sup>3</sup> 단국대학교 컴퓨터공학과

<sup>4</sup> 단국대학교 컴퓨터공학과

[kimsy9587@dankook.ac.kr](mailto:kimsy9587@dankook.ac.kr), [min2ndid@naver.com](mailto:min2ndid@naver.com), [kimpyo9357@naver.com](mailto:kimpyo9357@naver.com), [choisi@dankook.ac.kr](mailto:choisi@dankook.ac.kr)

## A Study on Tools Vehicle Detection and Vehicle Tracking

Se-Young Kim<sup>1</sup>, Jae-Eun Min<sup>2</sup>, Se-Hun Pyo<sup>3</sup>, Sang-Il Choi<sup>4</sup>

<sup>1</sup> Dept of. AI-based Convergence

<sup>2</sup> Dept of. AI-based Convergence

<sup>3</sup> Dept of. Computer Engineering

<sup>4</sup> Dept of. Computer Engineering

### 요약

차량 탐지와 차량 추적 기술은 교통관리 시스템, 자율주행 자동차 시스템 및 이를 응용한 보안 감시 시스템, 군사 작전 및 안전 등 다양한 산업 분야에서 활용되고 있다. 본 논문에서는 차량 탐지는 YOLOv7 모델을, 차량 추적은 DeepSORT 알고리즘을 사용하여 도로의 차량들에 대해 탐지 및 추적을 순차적으로 진행하였다. 실험환경은 차량 탐지 데이터 셋 (dataset)을 직접 라벨링(labeling) 하여 실험하였고, 차량 추적은 차량 탐지에서 학습해서 얻은 체크포인트(checkpoint) 모델을 가중치로 설정하여 실험을 진행하였다. 차량 탐지 실험결과는 validation 과 test에서 높은 정확도를 확인할 수 있었고, 차량 추적은 Namsa 비디오 및 Seohaegyo 비디오에서도 차량 추적이 잘 되고 있음을 확인할 수 있었다.

### 1. 서론

현재 차량 탐지와 차량 추적 기술은 교통관리 시스템, 자율주행 자동차 시스템 및 이를 응용한 보안 감시 시스템, 군사 작전 및 안전 등 다양한 산업 분야에서 활용되고 있다. 차량 탐지와 차량 추적을 기반으로 한 교통관리 시스템은 교통 위반 행위를 모니터링하고 법 집행에 도움을 주는 데 활용되며, 자율주행 자동차 시스템은 주변 차량을 탐지하고 추적하여 안전한 주행을 지원한다. 또한 주차장, 공항, 주요 건물 및 시설의 보안 감시 시스템에도 적용되며, 적대적 차량 및 드론(drone)을 탐지하고 추적하여 군사 작전 및 안전을 강화하는 데에도 사용

한다. 본 논문에서는 도로의 효율적 교통관리 시스템을 수립할 수 있도록 차량의 탐지 및 위치 추적을 제공한다.

우선, YOLOv7 모델을 이용한 실시간 차량 탐지 결과와 실제 결과값의 비교를 통해 정확도를 향상시키는 방법을 제안한다. 그리고 차량 추적은 DeepSORT 알고리즘을 사용하여 정확한 추적 정보를 획득하고, Namsa 및 Seohaegyo 비디오에서 차량 추적 결과를 검증한다.

### 2. 연구방법

본 논문에서는 크게 차량 탐지와 차량 추적 2 가지 단계로 나눠서 순차적으로 진행하였다.

연구를 진행할 때 “First, deep learning technologies are used to detect and track vehicles on the road to calculate the distance that pixels corresponding to vehicles have traveled.” (Sungwoo Byun et al., 2021, p 4) [1] 을 참고하였다.

## 2.1 차량 탐지

우선, 차량 탐지를 할 때 YOLOv7 [2] 모델을 이용하였다.

YOLOv7 은 real-time object detectors 중 하나로 비디오 시퀀스에서 특정 객체를 탐지하는 컴퓨터 비전 모델이다. 그러나 기존의 real-time object detector 들과 달리 추론 비용을 증가시키지 않고도 탐지 정확도를 크게 향상시킬 수 있는 방법들을 제안한다. 그 방법들을 bag-of-freebies 라고 통칭한다. 또한, 확장된 ELAN 모델인 E-ELAN 모델을 제안하여 원래의 ELAN 아키텍처를 유지하는 것 외에도 다양한 기능을 학습할 수 있도록 여러 그룹의 computational block 을 안내할 수 있다. 이러한 작동 방법은 네트워크의 학습 능력을 지속적으로 향상시킬 수 있다. 그리고 연결 기반 모델에 대한 복합 스케일링 기법(compound model scaling method for a concatenation-based model)을 제안하여 초기 설계 시 모델이 가지고 있던 특성을 유지하고, 최적의 구조를 유지할 수 있다.

## 2.2 차량 추적

차량 추적을 할 때 DeepSORT [3] 알고리즘을 사용하였다.

DeepSORT 는 페색을 통한 추적에 취약하고, 객체들이 움직일 때, ID 추적이 변경되는 ID 스위치에 취약하다는 SORT [4]의 단점을 보완하고자 만든 알고리즘이다. SORT 에 사전 학습된 컨볼루션 신경망(CNN)을 적용하고, 매칭 캐스케이드(matching cascade)를 추가하여 불확실성이 낮은 객체에 우선 순위를 부여한다. 이러한 작동 방법은 이전보다 더 정확하고 세밀한 추적(tracking) 정보를 얻을 수 있다. 또한, 속도(speed)도 더 빨라졌다.

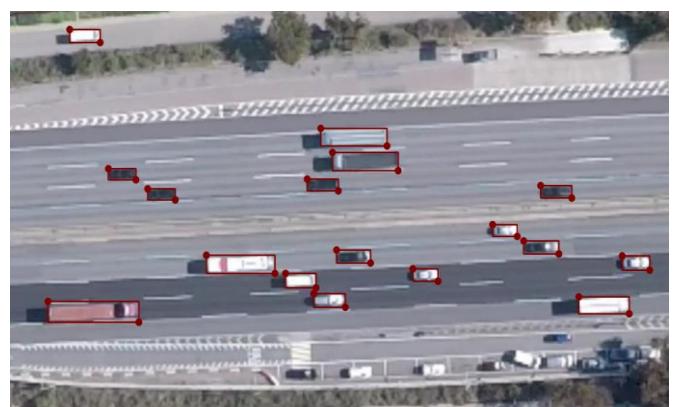
## 3. 실험환경

우선, 차량 탐지 데이터셋의 경우, 직접 라벨링(labeling) 하여 실험하였다. 총 실험을 2 번 진행하

였다.

첫 번째 실험의 경우, 데이터셋은 약 10 분짜리 Namsa 비디오를 1 초 간격으로 샘플링(sampling)하여 총 585 장의 이미지를 만들었고, train : valid : test = 2 : 1 : 1 의 비율로 설정하였다.

두 번째 실험의 경우, 데이터셋은 약 10 분짜리 Namsa 비디오와 Seohaegyo 비디오를 1 초 간격으로 샘플링(sampling)하여 총 585 장의 이미지를 만들었고, train : valid : test = 2 : 1 : 1 의 비율로 설정하되, Training set 은 Namsa, validation 과 test set 은 Seohaegyo 로 하였다.



(그림 1) 직접 라벨링(labeling) 한 Namsa 데이터셋 예시. 여기서 버스와 트럭도 같이 라벨링 하였다.

epoch 은 100 으로, batch size 는 16 으로, lr0 은 0.01, lrf 는 0.1 로 설정하였다. 또한, 이미지 사이즈는 640 \* 640 크기로 출력되도록 하였다.

Pytorch 는 1.11.0 버전을 설치하였고, NVIDIA GeForce RTX 3090 GPU 로 실험을 진행하였다.

차량 추적의 경우, 차량 탐지에서 직접 학습해서 얻은 체크포인트(checkpoint) 모델을 가중치로 설정하여 실험을 진행하였다.

## 4. 실험결과

	Precision	Recall	mAP0.5	mAP0.5:0.95
Valid	0.9908	0.9943	0.9964	0.7688
Test	0.988	0.993	0.995	0.77

<표 1> 첫 번째 실험의 차량 탐지 결과. (train, valid, test set 모두 Namsa)

	Precision	Recall	mAP0.5	mAP0.5:0.95
Valid	0.94	0.9354	0.9667	0.5025
Test	0.983	0.947	0.991	0.569

<표 2> 두 번째 실험의 차량 탐지 결과. (train set 은 Namsa, valid 와 test set 은 Seohaegyo)

YOLOv7 으로 차량 탐지를 한 결과이다. (표 1), (표 2) 를 보면 학습 후, validation 과 test 에서 유사한 수치가 나왔음을 확인할 수 있다.

Precision(P)은 positive 라고 예측한 비율 중 진짜 positive 의 비율을 나타낸다. Precision 이 높을수록 성능이 좋다고 할 수 있다.

Recall(R)은 진짜 positive 중 positive 라고 예측한 비율을 나타낸다. Recall 또한 높을수록 성능이 좋다고 할 수 있다.

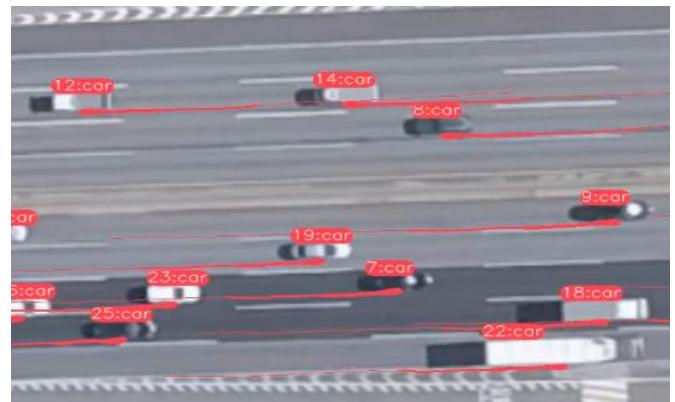
mAP 는 mean Average Precision 의 약자로, AP 의 평균을 의미한다. AP 는 precision-recall 그래프에서 그래프 선 아래쪽의 면적으로 계산된다. AP 가 높을수록 성능이 우수하다고 할 수 있다. mAP0.5 는  $IoU = 0.5$  기준일 때 계산된 mAP 값을 의미한다. mAP0.5:0.95 는 0.5 부터 0.95 까지 step 이 0.05 일 때, 각 임계값(thresholds) (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95)에 대해 AP 의 평균을 구한 것이다.



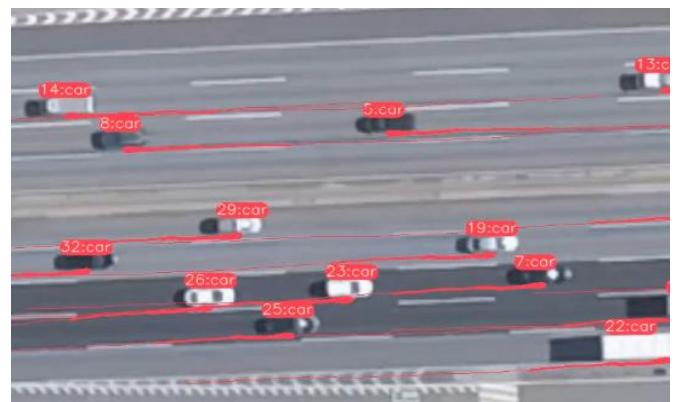
(그림 2) 직접 학습한 YOLOv7 모델을 가지고 차량 탐지(detection)한 모습. 탐지된 Namsa 비디오 중 한 장면을 캡쳐.



(그림 3)(그림 2) 확대.



(그림 4) DeepSORT 로 차량 추적(tracking)한 모습. 추적된 Namsa 비디오 중 한 장면을 캡쳐.



(그림 5)(그림 4)의 1 초 후 모습을 캡쳐.

## 5. 결론 및 향후 연구

본 논문에서는 차량 탐지와 차량 탐지를 기반으로 한 차량 추적에 대해 연구하고 실험하였다. 차량 탐지와 차량 추적은 서로 독립적이지 않고 연속적임을 확인할 수 있었다. <표 1>의 경우, validation 에서 precision 0.9908 을, test 에서는 0.988 을 기록하였고, <표 2>의 경우, validation 에서 precision 0.94 를, test 에서는 0.983

을 기록하여 validation 과 test 가 유사한 수치가 나왔음을 알 수 있었다. 이러한 실험을 바탕으로 앞으로 더 발전된 모델 및 알고리즘이 나와서 더 빠르고, 정확한 차량 탐지와 차량 추적을 할 수 있을 것이라고 생각한다. 차량 모니터링은 도시 및 교외 지역의 효율적인 교통 정책 및 전략을 개발하는 데 도움이 될 것이다. 더 나아가 차량에 국한되지 않고 다양한 객체에도 적용할 수 있음을 기대한다.

향후 연구에서는 폐색이 있고, 상대적으로 덜 선명한 난도 높은 이미지에 대해서도 학습 및 테스트를 해보자 한다. 또한, Bytetrack 과 같이 reID(재식별) 성능이 뛰어난 기법에 대해서도 적용성을 확인하고 관련된 평가 지표를 기준으로 성능 비교를 해보고자 한다.

### 사사문구

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 학석사연계 ICT 핵심인재양성사업의 연구결과로 수행되었음 (No. RS-2023-00259867). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) support program (No. RS-2023-00259867) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

### 참고문헌

- [1] Sungwoo Byun et al. Road Traffic Monitoring from UAV Images Using Deep Learning Networks, 2021.
- [2] Chien-Yao Wang et al. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.
- [3] Nicolai Wojke et al. SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC, 2017.
- [4] Alex Bewley et al. SIMPLE ONLINE AND REALTIME TRACKING, 2016.