



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0029268
(43) 공개일자 2024년03월05일

(51) 국제특허분류(Int. Cl.)
G08G 1/095 (2006.01) G08G 1/07 (2006.01)
(52) CPC특허분류
G08G 1/095 (2013.01)
G08G 1/075 (2013.01)
(21) 출원번호 10-2022-0107524
(22) 출원일자 2022년08월26일
심사청구일자 2022년08월26일

(71) 출원인
동서대학교 산학협력단
부산광역시 사상구 주례로 47(주례동,
동서대학교)
(72) 발명자
황기현
부산광역시 해운대구 반여로 67 해운대메가센텀한
화꿈에그린아파트 109동 2203호
박준혁
경상북도 구미시 형곡서로1길 40 삼우APT 911동
101호
전대호
경상남도 양산시 양주로 114 현대아파트 104동
503호
(74) 대리인
김석계

전체 청구항 수 : 총 3 항

(54) 발명의 명칭 AI를 이용한 실시간 인공지능 신호등 시스템

(57) 요약

본발명은 AI를 이용한 실시간 인공지능 신호등 시스템에 관한 것으로, AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하는 것을 특징으로 한다.

또한, 상기 교차로가 사거리, 삼거리 교차로일 경우, 차량인식 후 교차로의 차량 수를 세어 기존의 정해둔 기준치보다 초과할 경우, 제어부는 신호의 시간을 유동적으로 늘리며, 기준치보다 미만일 경우엔 기존의 신호체계에 따르며, 다음 교차로 넘어가게 하는 것으로, 따라서 본발명은 AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하여 교통을 원활히 하고 차량과 사람들의 통행에 불편이 없게 하는 현저한 효과가 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1345356165
과제번호	LINC3.0-2022-60
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	3단계 산학협력 선도대학 육성사업(LINC 3.0)
연구과제명	3단계 산학협력 선도대학 육성사업(LINC 3.0)(수요맞춤성장형)
기 여 율	1/1
과제수행기관명	동서대학교산학협력단
연구기간	2022.03.01 ~ 2028.02.28

명세서

청구범위

청구항 1

AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하는 것을 특징으로 하는 AI를 이용한 실시간 인공지능 신호등 시스템

청구항 2

제1항에 있어서, 상기 교차로가 사거리, 삼거리 교차로일 경우, 차량인식 후 교차로의 차량 수를 세어 기존의 정해둔 기준치보다 초과할 경우, 제어부는 신호의 시간을 유동적으로 늘리며, 기준치보다 미만일 경우엔 기존의 신호체계에 따르며, 다음 교차로 넘어가게 하는 것을 특징으로 하는 AI를 이용한 실시간 인공지능 신호등 시스템

청구항 3

제2항에 있어서, 상기 교차로가 꼬리물기가 심한 교차로의 경우, 교차로 중앙을 인식하여 이동 방향으로 중앙까지 정지된 차량이 있는 경우 혹은 진행 중인 차로에 계속 차량이 밀리는 경우, 제어부는 기존의 시간보다 빠르게 정지 신호를 부여하게 하는 것을 특징으로 하는 AI를 이용한 실시간 인공지능 신호등 시스템

발명의 설명

기술 분야

[0001] 본발명은 AI를 이용한 실시간 인공지능 신호등 시스템에 관한 것으로, 보다 상세하게는 AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하여 교통을 원활히 하고 차량과 사람들의 통행에 불편이 없게 하는 AI를 이용한 실시간 인공지능 신호등 시스템에 관한 것이다.

배경 기술

[0002] 일반적으로 도심등의 교차로에는 차량이 정체되기 쉽고, 따라서 교통체증 등이 심해져 불편을 초래하며, 이에 대한 해결책으로서 종래특허기술의 일례로서 공개특허공보 공개번호 10-2021-0128823호에는 저장매체에 저장된 빅데이터 및 인공지능 기반 교차로 서비스수준 예측 프로그램이 설치된 관리서버를 통해 수행되는 빅데이터 및 인공지능 기반 교차로 서비스수준 예측방법에 있어서, 교차로 서비스수준과 상관관계를 가지는 복수 개의 교통관계데이터를 수집하는 (a)단계;

[0003] 상기 (a)단계에 의해 수집된 교통 관계데이터에 대해 전처리를 수행하는 (b)단계;

[0004] 상기 전처리된 교통 관계데이터로부터, 기 설정된 시간단위 기준에 따라 교차로의 각 유입부에 대한 차량 당 제어지체 값을 산출하는 (c)단계;

[0005] 상기 (c)단계에 의해 산출된 차량 당 제어지체 값을 인공신경망에 입력데이터로 입력하고, 기 설정된 시간단위로 학습 및 평가하는 (d)단계; 및

[0006] 상기 (d)단계에 의해 상기 인공신경망이 학습한 모델을 통해 교차로 서비스수준을 예측하는 (e)단계;

[0007] 를 포함하는 빅데이터 및 인공지능 기반 교차로 서비스수준 예측방법. 공개번호 공개특허공보 10-2021-0031975호에는 교차로에서 교통 관리 시스템을 제어하는 시스템이 공개되어 있다.

[0008] 그러나 상기 종래기술들은 교차로 근처의 차량인식이 부정확하고 제어연산에 시간이 많이 걸려 교통체증을 유발하거나 불편을 오히려 초래하는 단점이 있었다.

발명의 내용

해결하려는 과제

[0009] 따라서 본발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로, AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하여 교통을 원활히 하고 차량과 사람들의 통행에 불편이 없게 하는 AI를 이용한 실시간 인공지능 신호등 시스템을 제공하고자 하는 것이다.

과제의 해결 수단

[0010] 본발명은 AI를 이용한 실시간 인공지능 신호등 시스템에 관한 것으로, AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하는 것을 특징으로 한다.

[0011] 또한, 상기 교차로가 사거리, 삼거리 교차로일 경우, 차량인식 후 교차로의 차량 수를 세어 기존의 정해둔 기준치보다 초과할 경우, 제어부는 신호의 시간을 유동적으로 늘리며, 기준치보다 미만일 경우엔 기존의 신호체계에 따르며, 다음 교차로 넘어가게 하는 것을 특징으로 한다.

발명의 효과

[0012] 따라서 본발명은 AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하여 교통을 원활히 하고 차량과 사람들의 통행에 불편이 없게 하는 현저한 효과가 있다.

도면의 간단한 설명

[0013] 도 1은 Car_Detection의 Image
 도 2, 3은 Car_Detection의 학습 과정
 도 4는 Vehicles_Detection의 Image
 도 5는 Vehicles_Detection의 학습 과정
 도 6은 순서도
 도 7은 사거리 교차로
 도 8은 순서도
 도 9는 사거리 교차로
 도 10은 순서도
 도 11은 꼬리물기가 심한 교차로
 도 12는 confusion matrix 결과
 도 13은 손실평가지표 결과
 도 14는 Result
 도 15는 confusion matrix 결과
 도 16은 손실평가지표 결과
 도 17은 Result
 도 18은 인공지능의 테스트를 위한 영상 자료 수집
 도 19는 영상에 적용한 결과
 도 20은 Dataset Car_Detection 학습 결과물
 도 21은 Dataset Vehicles_Detection 학습 결과물
 도 22는 YOLO v5 Sample Code를 이용한 결과물

발명을 실시하기 위한 구체적인 내용

[0014] 본발명은 AI를 이용한 실시간 인공지능 신호등 시스템에 관한 것으로, AI모듈을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하고 신호등을 제어하는 것을 특징으로 한다.

- [0015] 또한, 상기 교차로가 사거리, 삼거리 교차로일 경우, 차량인식 후 교차로의 차량 수를 세어 기존의 정해둔 기준치보다 초과할 경우, 제어부는 신호의 시간을 유동적으로 늘리며, 기준치보다 미만일 경우엔 기존의 신호체계에 따르며, 다음 교차로 넘어가게 하는 것을 특징으로 한다.
- [0016] 또한, 상기 교차로가 꼬리물기가 심한 교차로의 경우, 교차로 중앙을 인식하여 이동 방향으로 중앙까지 정지된 차량이 있는 경우 혹은 진행 중인 차로에 계속 차량이 밀리는 경우, 제어부는 기존의 시간보다 빠르게 정지 신호를 부여하게 하는 것을 특징으로 한다.
- [0018] 본발명을 첨부도면에 의해 상세히 설명하면 다음과 같다.
- [0019] 본 발명 AI를 이용한 '실시간 인공지능 신호등 구현'은 AI 기술을 통해 영상인식으로 교차로 근처에 차량을 인식하고 측정하여 혼잡 여부를 판단하는 구성은 다음과 같다.
- [0020] 실시간 영상처리 기반 혼잡 여부 판단 알고리즘 구현은 시간별로 영상인식을 통해 교차로 차량 데이터 수집, 차량 이미지를 인식하고 학습하여 교차로 Traffic을 측정하기 위한 AI 인공지능 사용한다.
- [0022] 일부 수정된 알고리즘에 수집된 데이터를 학습은 GitHub에서 Car-Detector-YOLOv3-tiny 오픈소스 다운로드, ROBOFLOW에서 roboflow_Vehicles-OpenImages Dataset 오픈소스 다운로드한다.
- [0024] 인공지능 신호등 제어 알고리즘 구현은 측정된 알고리즘 신호로 기존의 신호등에 신호를 판단, 혼잡하고 원활한 차로에 각각 유동적인 신호 시간을 부여하는 시스템으로, 여러 가지 형태의 교차로에 다양한 알고리즘 구현된다.
- [0026] 본발명의 세부구성에 대해 설명하면,
- [0027] 실시간 영상처리 기반 혼잡 여부 판단 알고리즘을 구현하는 것으로, Python을 위한 오픈소스 딥러닝 라이브러리인 PyTorch 사용하며, GitHub에서 Object Detection 오픈소스 코드 YOLO v5 사용한다.
- [0029] 일부 수정된 알고리즘에 수집된 데이터를 학습으로서, Dataset Car_Detection의 Image를 학습은 GitHub에서 오픈소스 Car-Detector-YOLOv3-tiny를 학습시키는 과정으로, 학습하는데 소요된 시간은 약 75분이 소요되었다.
- [0031] Dataset Vehicles_Detection의 Image를 학습하는 것은 ROBOFLOW에서 roboflow_Vehicles-OpenImages Dataset을 학습시키는 과정으로, 학습하는데 소요된 시간은 약 8시간이 소요되었다.
- [0033] 인공지능 신호등 제어 알고리즘에 대해 설명하면 다음과 같다.
- [0034] 1. 일반적인 사거리, 삼거리 교차로일 경우
- [0035] 차량인식 후 교차로의 차량 수를 세어 기존의 정해둔 기준치보다 초과할 경우, 제어기의 제어부는 신호의 시간을 유동적으로 늘리며, 기준치보다 미만일 경우엔 기존의 신호체계에 따르며, 다음 교차로 넘어간다. 제어기는 카메라의 영상이 입력되는 입력부, 측정값과 설정치를 비교하는 비교부, 판단부, 데이터를 저장하는 저장부, 출력부, 송수신부, 제어부로 구성된다.
- [0036] 2. 삼거리 교차로일 경우 좌회전 차로에 차량인식 후 차량이 있는 경우, 양방향 차로 정지 신호 및 좌회전 차로 직진신호 부여, 좌회전 차로에 차량인식 후 차량이 없는 경우, 좌회전 차로 신호 생략한다.
- [0037]
- [0038] 3. 꼬리물기가 심한 교차로 예방 및 차단하는 것으로, 교차로 중앙을 인식하여 이동 방향으로 중앙까지 정지된 차량이 있는 경우 혹은 진행 중인 차로에 계속 차량이 밀리는 경우, 기존의 시간보다 빠르게 정지 신호 부여한

다.

- [0040] 본발명의 구현된 Dataset Car_Detection의 학습 결과
- [0041] confusion matrix는 예측값과 결과값이 얼마나 일치하는지를 나타내는 표이며, 좌상단-우하단 대각선상의 값, 높을수록 예측이 잘 이루어졌다는 지표가 된다.
- [0042] 평가지표는 요약하면 시간이 지나면서 학습해서 실제 값과 매우 근사해지며, 평균 정밀도는 상승세를 보이다 일정 구간에서 평균값을 내보이고 있다.
- [0043] train / cls_loss와 val / cls_loss의 경우는 예측 경계 상자가 분류 정확성을 측정하는 손실이 조금 늦게 줄어드는 모습을 볼 수 있었다.
- [0045] 구현된 Dataset Vehicles_Detection의 학습 결과, Dataset Vehicles_Detection의 confusion matrix는 앞의 Dataset Car_Detection과 달리 이상적인 결과를 도출해 낸다. 손실평가지표 역시 전반적으로 앞의 Dataset Car_Detection에 비해 이상적인 결과로 나온다.
- [0046] Dataset Vehicles_Detection와 Dataset Car_Detection의 confusion matrix와 손실평가지표를 비교한 결과, Dataset Vehicles_Detection이 신뢰성이 높다.
- [0048] 일부 수정된 알고리즘에 수집된 데이터를 학습하는 것에 대해 설명하면,
- [0049] 시험 데이터 자료 수집 : 부산 교통정보 센터에서 일부 영상을 수집 (크롤링)시험 데이터 가공 : 확장자 변경
오픈 사이트 이용 (ts to mp4)
- [0051] 평균 차량인식 수는 1대에서 5대로 측정이 된다.
- [0052] 학습 이미지가 차량의 측면을 중점으로 학습이 된 영향으로 측면의 차량만 인식하는 모습을 보이며, 다른 Object는 탐지하지 않는 깔끔한 모습을 보인다.
- [0053] 평균 차량인식 수는 6대에서 10대로, Dataset Car_Detection에 비해 소폭 상승한 탐지율을 보인다.
- [0054] 평균 차량인식 수는 11대에서 최대 18대까지 측정하고 다른 Object까지 측정하는 경우 더 많이 측정 가능하며, 학습한 Dataset 보다 높은 인식률을 보인다.

도면

도면1



도면5

```

Class Images Loss P R mAP@5 mAP@5 .95: 100% 43/41 (00:46:30:00, 1.33it/s)
all 218 0.55 0.493 0.527 0.57 0.415

Epoch gpus mem bsz obj rlx loss lr_size
43/49 00 0.82391 0.82316 0.00139 0 438: 145% 43/43W (07:10:00:00, 1.8211/s)
Class Images Loss P R mAP@5 mAP@5 .95: 100% 43/41 (00:46:30:00, 1.33it/s)
all 218 0.55 0.481 0.509 0.56 0.408

Epoch gpus mem bsz obj rlx loss lr_size
43/49 00 0.82344 0.82319 0.00351 10 438: 145% 43/43W (07:10:00:00, 1.8211/s)
Class Images Loss P R mAP@5 mAP@5 .95: 100% 43/41 (00:46:30:00, 1.33it/s)
all 218 0.55 0.475 0.542 0.54 0.417

Epoch gpus mem bsz obj rlx loss lr_size
43/49 00 0.82431 0.82375 0.00493 0 438: 145% 43/43W (07:11:00:00, 1.8211/s)
Class Images Loss P R mAP@5 mAP@5 .95: 100% 43/41 (00:46:30:00, 1.33it/s)
all 218 0.55 0.425 0.549 0.51 0.404

Epoch gpus mem bsz obj rlx loss lr_size
43/49 00 0.82451 0.82398 0.00427 7 438: 145% 43/43W (07:11:00:00, 1.8211/s)
Class Images Loss P R mAP@5 mAP@5 .95: 100% 43/41 (00:46:30:00, 1.33it/s)
all 218 0.55 0.345 0.614 0.55 0.408

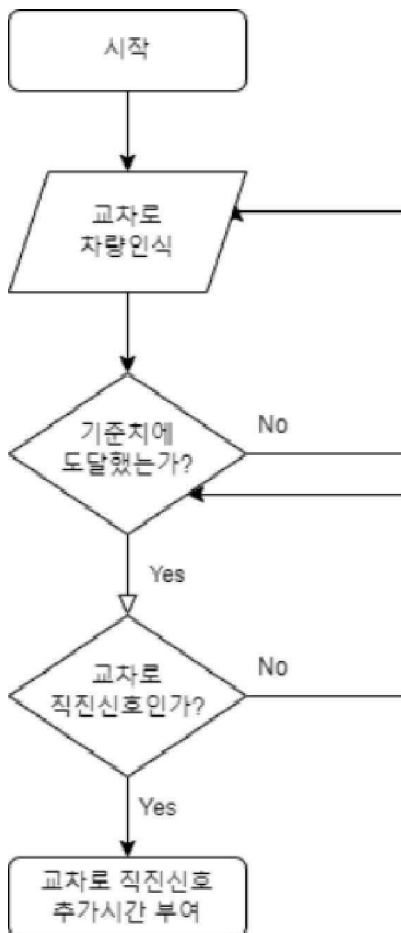
50 epochs completed in 0.744 hours.
npz_loader is ripped from runAttnet/vehicles.py:net_results/weights/last.pt, 14.49
npz_loader is ripped from runAttnet/vehicles.py:net_results/weights/last.pt, 14.49

We loading -runAttnet/vehicles.py:net_results/weights/last.pt...
Finishing layers...
YOLOv5 summary: 212 layers, 182368 parameters, 4 gradients
Class Images Loss P R mAP@5 mAP@5 .95: 100% 43/41 (00:46:30:00, 1.4211/s)
all 218 0.55 0.48 0.499 0.57 0.408
-noinstance 218 0.4 0.781 0.512 0.14 0.649
Box 218 46 0.858 0.652 0.76 0.551
Car 218 236 0.488 0.361 0.37 0.31
Motorcycle 218 46 0.388 0.435 0.55 0.541
Truck 218 00 0.480 0.233 0.33 0.231

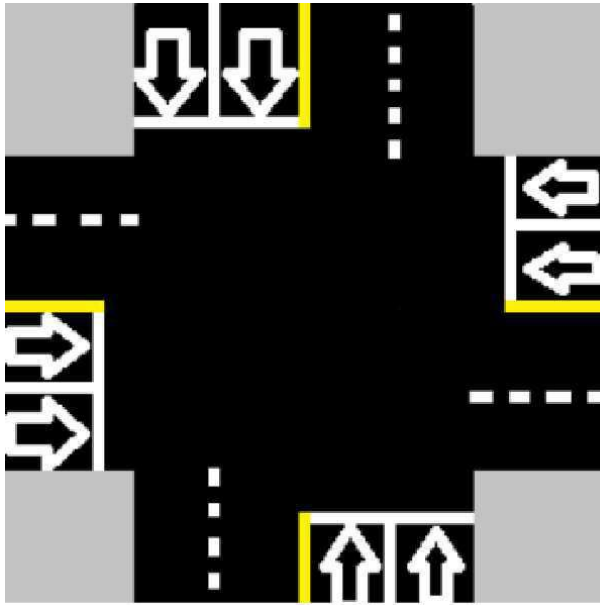
Results saved to runAttnet/vehicles.py:net_results
Exception ignored in: <function StorageRef...set... at 0x0000000000000000>

```

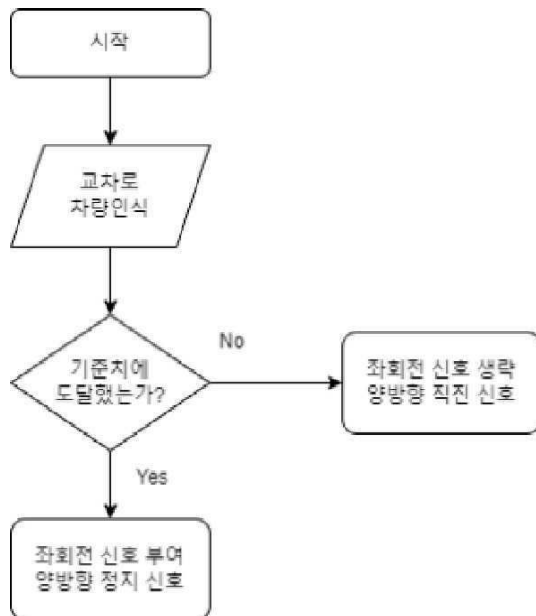
도면6



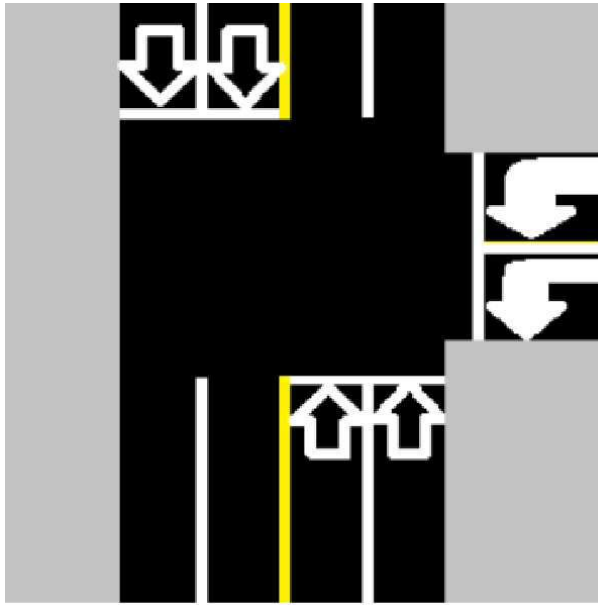
도면7



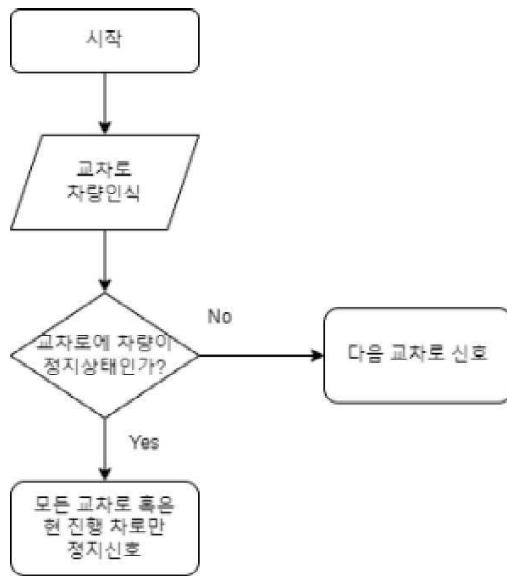
도면8



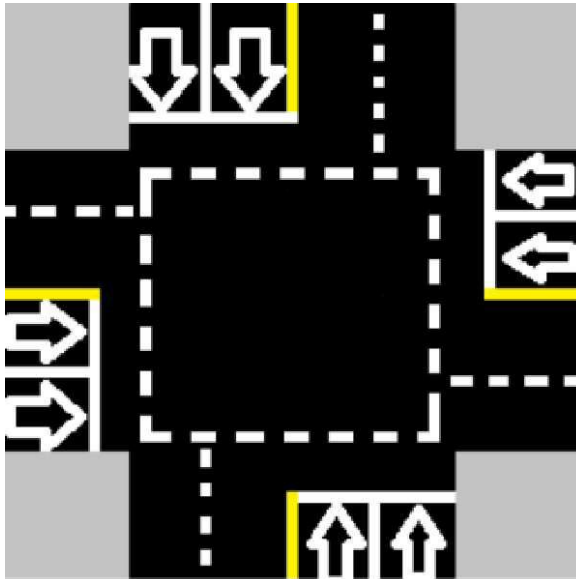
도면9



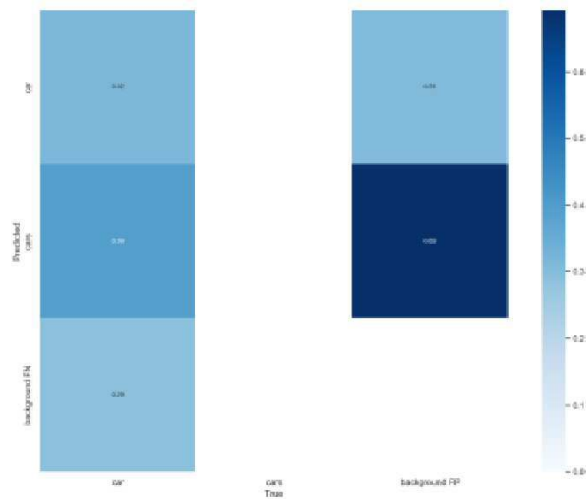
도면10



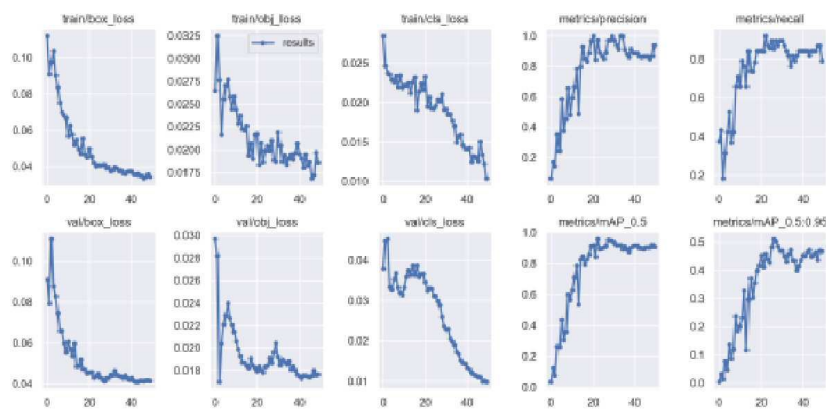
도면11



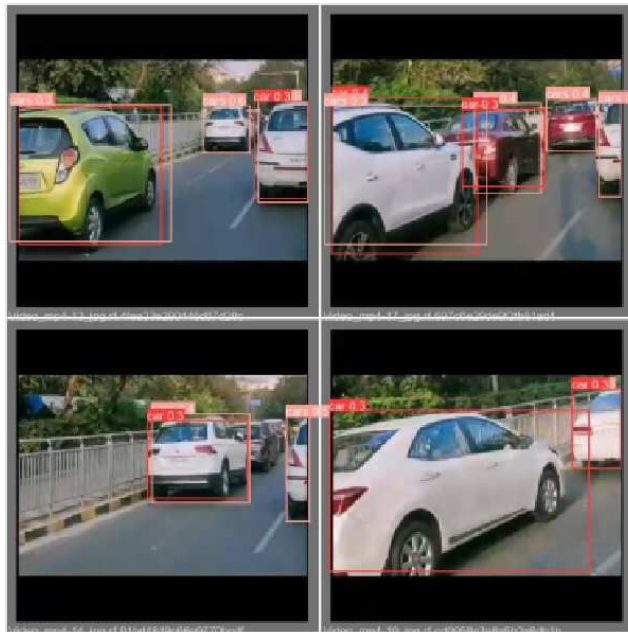
도면12



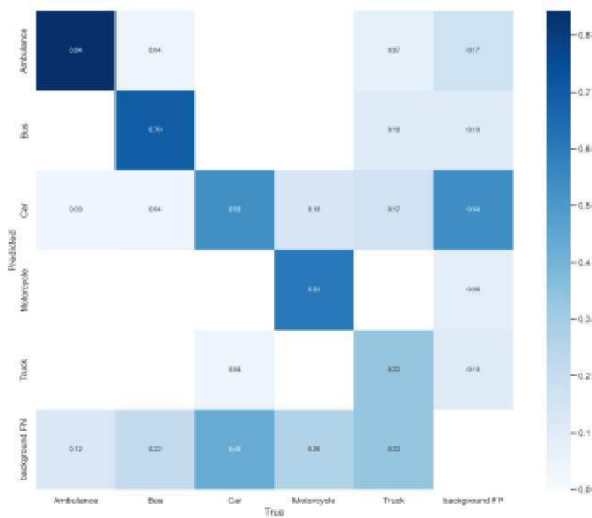
도면13



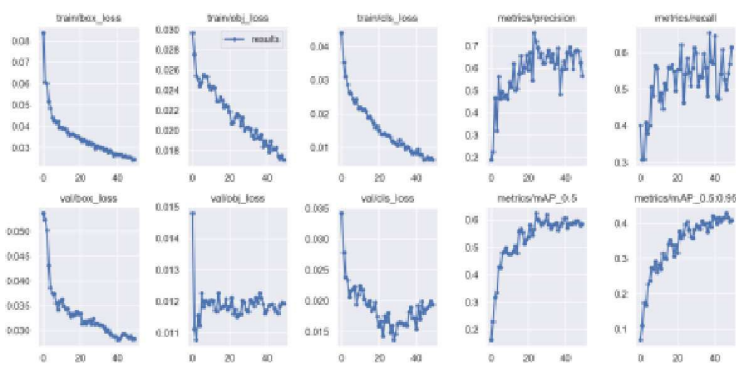
도면14



도면15



도면16



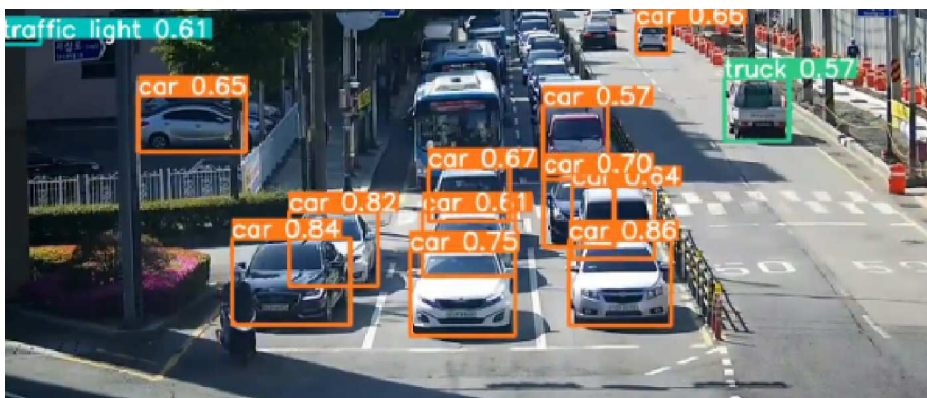
도면17



도면18



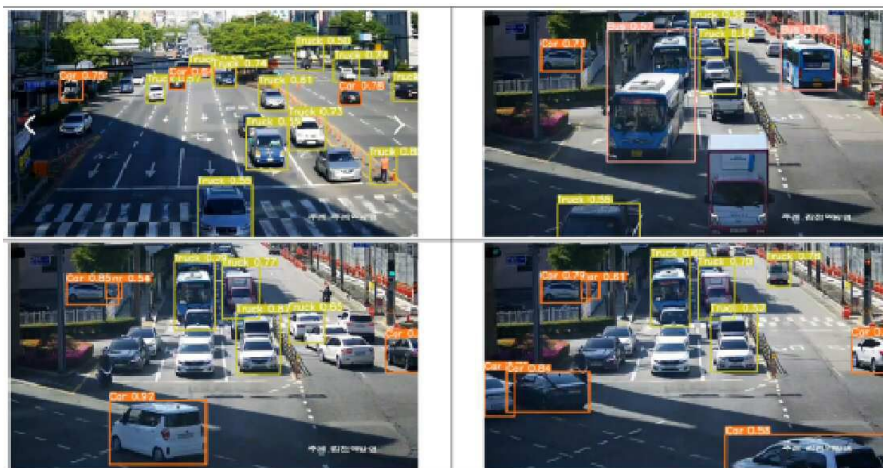
도면19



도면20



도면21



도면22

