



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0161291
(43) 공개일자 2024년11월12일

(51) 국제특허분류(Int. Cl.)

G06V 20/58 (2022.01) G01S 17/89 (2020.01)
G06N 3/08 (2023.01) G06V 10/14 (2022.01)
G06V 10/82 (2022.01)

(52) CPC특허분류

G06V 20/58 (2022.01)
G01S 17/89 (2022.01)

(21) 출원번호 10-2023-0058213

(22) 출원일자 2023년05월04일

심사청구일자 없음

(71) 출원인

현대자동차주식회사

서울특별시 서초구 현릉로 12 (양재동)

기아 주식회사

서울특별시 서초구 현릉로 12 (양재동)

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

박종원

서울특별시 성동구 왕십리로21길 12, 401호 (행당동)

김진규

서울특별시 성북구 안암로 145, 고려대학교산학협력단 (안암동5가)

이다은

서울특별시 성북구 고려대로22길 15, 306호 (안암동5가, 영산빌딩)

(74) 대리인

박병석

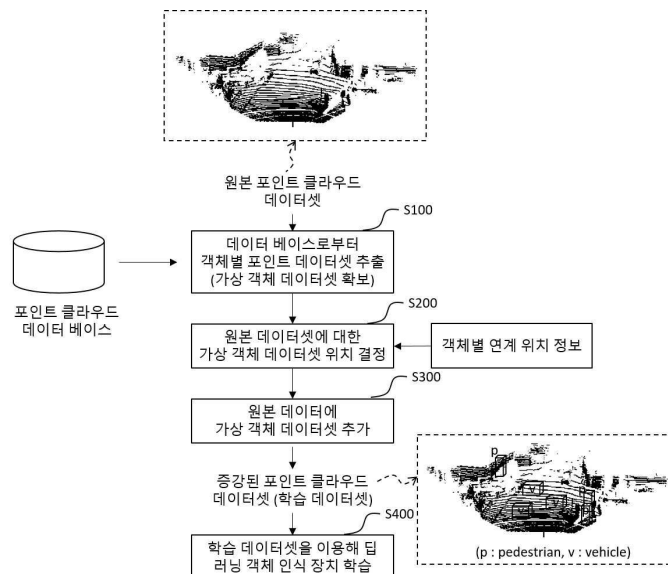
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법 및 그 라이다 기반 딥러닝 객체 인식 장치

(57) 요약

본 발명의 하나의 실시예에 의한 라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법은, 라이다 포인트 클라우드의 원본 데이터셋을 준비하는 단계, 포인트 클라우드 데이터 베이스로부터 가상 객체 데이터셋을 획득하는 단계, 객체별 연계 위치 정보를 기반으로 상기 가상 객체 데이터셋을 상기 원본 데이터셋에 추가하여 학습 데이터셋을 얻는 단계, 및 상기 학습 데이터셋을 이용해 라이다 기반 딥러닝 객체 인식 장치를 학습시키는 단계를 포함한다.

대표도 - 도2



(52) CPC특허분류

G06N 3/08 (2023.01)

G06V 10/14 (2023.08)

G06V 10/82 (2022.01)

명세서

청구범위

청구항 1

포인트 클라우드 데이터 베이스로부터 가상 객체 데이터셋을 획득하는 단계;

객체별 연계 위치 정보를 기반으로 상기 가상 객체 데이터셋을 원본 데이터셋에 추가하여 학습 데이터셋을 얻는 단계; 및

상기 학습 데이터셋을 이용해 라이다 기반 딥러닝 객체 인식 장치를 학습시키는 단계

를 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 2

제1 항에 있어서,

상기 가상 객체 데이터셋은 상기 데이터 베이스의 적어도 하나의 포인트 클라우드 데이터셋으로부터 획득한 적어도 하나의 동적 객체 데이터셋을 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 3

제2 항에 있어서,

상기 적어도 하나의 동적 객체 데이터셋은 보행자, 자전거, 모터사이클, 승용차, 트럭, 버스, 트레일러, 건설용 중장비 차량(construction vehicle) 중 적어도 하나를 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 4

제3 항에 있어서,

상기 객체별 연계 위치 정보는 상기 원본 데이터셋의 포인트 라벨 정보에 따라 결정되는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 5

제4 항에 있어서,

상기 포인트 라벨 정보는 사이드워크, 주행 가능 영역, 로드 중 적어도 하나를 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 6

제5 항에 있어서,

상기 객체별 연계 위치 정보는 상기 모터사이클, 승용차, 트럭, 버스, 트레일러 및 건설용 중장비 차량은 상기 로드로 연계시키고, 상기 보행자는 상기 사이드워크에 연계시키며, 상기 자전거는 상기 사이드워크 및/또는 상기 도로에 연계시킨 것을 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 7

제1 항에 있어서,

상기 딥러닝 객체 인식 장치는 딥러닝 모델에 기반한 객체 인식 소프트웨어를 저장한 메모리 및 상기 메모리의 상기 소프트웨어를 실행시키는 적어도 하나의 프로세서를 포함하고,

상기 딥러닝 모델은, 공통 백본 네트워크와 클래스별로 로스(loss)를 출력하는 복수의 헤드 네트워크를 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 8

제7 항에 있어서,

상기 딥러닝 모델은 상기 클래스별 로스를 웨이트드 합산(weighted sum)해 최종 로스를 출력하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 9

제8 항에 있어서,

상기 최종 로스는 다이내믹 웨이트 평균(DWA; Dynamic Weight Average) 법을 통해 얻는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 10

제1 항에 있어서,

상기 데이터 베이스는 누센(nuScenes) 및/또는 키티(KITTI) 데이터셋을 포함하는,

라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법.

청구항 11

딥러닝 모델에 기반한 객체 인식 소프트웨어를 저장한 메모리; 및

상기 메모리의 상기 소프트웨어를 실행시키는 적어도 하나의 프로세서를 포함하고,

상기 딥러닝 모델은, 포인트 클라우드 데이터 베이스로부터 가상 객체 데이터셋을 획득하고, 객체별 연계 위치 정보를 기반으로 상기 가상 객체 데이터셋을 원본 데이터셋에 추가하여 얻은 학습 데이터셋을 이용해 학습된,

라이다 기반 딥러닝 객체 인식 장치.

청구항 12

제11 항에 있어서,

상기 가상 객체 데이터셋은 상기 데이터 베이스의 적어도 하나의 포인트 클라우드 데이터셋으로부터 획득한 적어도 하나의 동적 객체 데이터셋을 포함하는,

라이다 기반 딥러닝 객체 인식 장치.

청구항 13

제12 항에 있어서,

상기 적어도 하나의 동적 객체 데이터셋은 보행자, 자전거, 모터사이클, 승용차, 트럭, 버스, 트레일러, 건설용 중장비 차량(construction vehicle) 중 적어도 하나를 포함하는,

라이다 기반 딥러닝 객체 인식 장치.

청구항 14

제13 항에 있어서,

상기 객체별 연계 위치 정보는 상기 원본 데이터셋의 포인트 라벨 정보에 따라 결정되는,
라이다 기반 딥러닝 객체 인식 장치.

청구항 15

제14 항에 있어서,
상기 포인트 라벨 정보는 사이드워크, 주행 가능 영역, 로드 중 적어도 하나를 포함하는,
라이다 기반 딥러닝 객체 인식 장치.

청구항 16

제15 항에 있어서,
상기 객체별 연계 위치 정보는 상기 모터사이클, 승용차, 트럭, 버스, 트레일러 및 건설용 중장비 차량은 상기 로드로 연계시키고, 상기 보행자는 상기 사이드워크에 연계시키며, 상기 자전거는 상기 사이드워크 및/또는 상기 로드에서 연계시킨 것을 포함하는,
라이다 기반 딥러닝 객체 인식 장치.

청구항 17

제11 항에 있어서,
상기 딥러닝 모델은, 공통 백본 네트워크와 클래스별로 로스(loss)를 출력하는 복수의 헤드 네트워크를 포함하는,
라이다 기반 딥러닝 객체 인식 장치.

청구항 18

제17 항에 있어서,
상기 딥러닝 모델은 상기 클래스별 로스를 웨이트드 합산(weighted sum)해 최종 로스를 출력하는,
라이다 기반 딥러닝 객체 인식 장치.

청구항 19

제18 항에 있어서,
상기 최종 로스는 다이나믹 웨이트 평균(DWA; Dynamic Weight Average) 법을 통해 얻는,
라이다 기반 딥러닝 객체 인식 장치.

청구항 20

제11 항에 있어서,
상기 데이터 베이스는 누센(nuScenes) 및/또는 키티(KITTI) 데이터셋을 포함하는,
라이다 기반 딥러닝 객체 인식 장치.

발명의 설명

기술 분야

[0001] 본 발명은 라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법 및 그 라이다 기반 딥러닝 객체 인식 장치에 관한 것이다.

배경 기술

[0002] 자율주행 레벨이 고도화되면서, 객체 인식 장치의 성능 향상이 요구되고 있다.

- [0003] 이를 위해, 딥러닝 인공지능 모델을 이용한 객체 인식 방법이 개발되고 있다.
- [0004] 종래에 딥러닝 인공지능 모델을 이용한 라이다 기반 객체 인식 장치, 즉, 라이다 기반 딥러닝 객체 인식 장치는 객체 종류별(즉, 클래스별) 인식 네트워크를 달리한다. 객체 클래스별 인식 네트워크를 달리하기 때문에, 신뢰도 높은 인식 결과를 얻을 수 있지만, 네트워크가 많아져 연산 부하량이 증가하는 단점이 있다.
- [0005] 그 대안으로, 논문 "Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection" (Benjin Zhu 외 4, 2019, arXiv:1908.09492)에서는 멀티-헤드(milti-head) 아키텍처 딥러닝 모델을 제안하고 있다.
- [0006] 상기 멀티-헤드(milti-head) 아키텍처 딥러닝 모델은 백본을 공유하며 인식 결과를 출력하는 헤드를 클래스별로 달리 포함한다. 이 모델은 학습 데이터에 포함된 객체들 중 특정 클래스의 객체가 많은 경우, 학습이 해당 클래스에 집중되기 때문에 다른 클래스 객체에 대한 인식 신뢰도는 저하되는 문제가 있다. 즉, 예시적으로 학습 데이터에 차량에 대한 객체 수가 많고 보행자 데이터 수는 상대적으로 적은 경우, 차량에 대해서는 높은 인식 성능을 내지만, 보행자에 대한 인식 성능은 저하될 수 있다.
- [0007] 학습 데이터에 포함된 클래스별 객체 수의 불균형(imbalance) 문제는 필연적이며, 위 논문에서는 이러한 클래스별 불균형에 기인하는 헤드별 그라디언트(gradient) 불균형 문제에 대한 해결 방안을 제안하지 못하고 있다.
- [0008] 다만, 위 논문에서는 클래스별 불균형 해소를 위한 데이터 증강(data augmentation)을 위해, 그라운드-트루쓰(Ground-thrth) 샘플링(sampling) 기법을 제안하고 있다.
- [0009] 그라운드-트루쓰 샘플링 기법은 포인트 클라우드 데이터셋으로부터 특정 클래스의 객체에 대한 그라운드-트루쓰 포인트 클라우드 데이터셋을 수집한 후에 이를 가상 객체로써 추가함으로써 학습 데이터 내에서 해당 클래스 객체의 전체 수를 증가시키는 방법이다.
- [0010] 이 기법을 통해 특정 클래스에 속하는 가상 객체들의 추가를 통해 클래스별 불균형이 해소될 수 있지만, 가상의 데이터를 학습 데이터 내에 있어 임의의 위치에 추가하는 방식이기 때문에, 객체가 현실적이지 않은 위치에 추가되는 문제가 있다.
- [0011] 예시적으로, 도 1을 참조하면, (a)와 같은 원본 씬(scene)에 종래의 그라운드-트루쓰 샘플링 기법을 통해 데이터를 증강하는 경우, (b)에 예시되는 바와 같이 보행자가 도로 상으로 추가될 수 있는 문제가 있다.
- [0012] 이러한, 비현실적인 데이터 증강으로 인해 객체 인식에 대한 성능 향상에 한계가 있다.

발명의 내용

해결하려는 과제

- [0013] 본 발명은 전술한 종래기술의 적어도 하나의 문제를 해결하여, 객체 인식 장치의 인식 성능을 향상시키는 것으로 목적으로 한다.
- [0014] 본 발명의 하나의 실시예는 하나의 엔코더 (혹은 백본)을 공유하며 다양한 객체 클래스들에 대한 인식을 동시에 수행할 수 있는 멀티 헤드 딥러닝 네트워크 기반 객체 인식 방법 및 장치를 제공한다.

과제의 해결 수단

- [0015] 본 발명의 하나의 실시예에 의한 라이다 기반 딥러닝 객체 인식 장치 머신 러닝 방법은, 라이다 포인트 클라우드의 원본 데이터셋을 준비하는 단계, 포인트 클라우드 데이터 베이스로부터 가상 객체 데이터셋을 획득하는 단계, 객체별 연계 위치 정보를 기반으로 상기 가상 객체 데이터셋을 상기 원본 데이터셋에 추가하여 학습 데이터셋을 얻는 단계, 및 상기 학습 데이터셋을 이용해 라이다 기반 딥러닝 객체 인식 장치를 학습시키는 단계를 포함한다.
- [0016] 본 발명의 적어도 하나의 실시예에서, 상기 가상 객체 데이터셋은 상기 데이터 베이스의 적어도 하나의 포인트 클라우드 데이터셋으로부터 획득한 적어도 하나의 동적 객체 데이터셋을 포함한다.
- [0017] 본 발명의 적어도 하나의 실시예에서, 상기 적어도 하나의 동적 객체 데이터셋은 보행자, 자전거, 모터사이클, 승용차, 트럭, 버스, 트레일러, 건설용 중장비 차량(construction vehicle) 중 적어도 하나를 포함한다.
- [0018] 본 발명의 적어도 하나의 실시예에서, 상기 객체별 연계 위치 정보는 상기 원본 데이터셋의 포인트 라벨 정보에 따라 결정된다.

- [0019] 본 발명의 적어도 하나의 실시예에서, 상기 포인트 라벨 정보는 사이드워크, 주행 가능 영역, 로드 중 적어도 하나를 포함한다.
- [0020] 본 발명의 적어도 하나의 실시예에서, 상기 객체별 연계 위치 정보는 상기 모터사이클, 승용차, 트럭, 버스, 트레일러 및 건설용 중장비 차량은 상기 로드로 연계시키고, 상기 보행자는 상기 사이드워크에 연계시키며, 상기 자전거는 상기 사이드워크 및/또는 상기 로드와 연계시킨 것을 포함한다.
- [0021] 본 발명의 적어도 하나의 실시예에서, 상기 딥러닝 객체 인식 장치는 딥러닝 모델에 기반한 객체 인식 소프트웨어를 저장한 메모리 및 상기 메모리의 상기 소프트웨어를 실행시키는 적어도 하나의 프로세서를 포함하고, 상기 딥러닝 모델은, 공통 백본 네트워크와 클래스별로 로스(loss)를 출력하는 복수의 헤드 네트워크를 포함한다.
- [0022] 본 발명의 적어도 하나의 실시예에서, 상기 딥러닝 모델은 상기 클래스별 로스를 웨이트드 합산(weighted sum)해 최종 로스를 출력한다.
- [0023] 본 발명의 적어도 하나의 실시예에서, 상기 최종 로스는 다이내믹 웨이트 평균(DWA; Dynamic Weight Average)법을 통해 얻는다.
- [0024] 본 발명의 적어도 하나의 실시예에서, 상기 데이터 베이스는 누씬(nuScenes) 및/또는 키티(KITTI) 데이터셋을 포함한다.
- [0025] 본 발명의 하나의 실시예에 의한 라이더 기반 딥러닝 객체 인식 장치는, 딥러닝 모델에 기반한 객체 인식 소프트웨어를 저장한 메모리; 및 상기 메모리의 상기 소프트웨어를 실행시키는 적어도 하나의 프로세서를 포함하고, 상기 딥러닝 모델은, 포인트 클라우드 데이터 베이스로부터 가상 객체 데이터셋을 획득하고, 객체별 연계 위치 정보를 기반으로 상기 가상 객체 데이터셋을 원본 데이터셋에 추가하여 얻은 학습 데이터셋을 이용해 학습된다.
- [0026] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 가상 객체 데이터셋은 상기 데이터 베이스의 적어도 하나의 포인트 클라우드 데이터셋으로부터 획득한 적어도 하나의 동적 객체 데이터셋을 포함한다.
- [0027] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 적어도 하나의 동적 객체 데이터셋은 보행자, 자전거, 모터사이클, 승용차, 트럭, 버스, 트레일러, 건설용 중장비 차량(construction vehicle) 중 적어도 하나를 포함한다.
- [0028] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 객체별 연계 위치 정보는 상기 원본 데이터셋의 포인트 라벨 정보에 따라 결정된다.
- [0029] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 포인트 라벨 정보는 사이드워크, 주행 가능 영역, 로드 중 적어도 하나를 포함한다.
- [0030] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 객체별 연계 위치 정보는 상기 모터사이클, 승용차, 트럭, 버스, 트레일러 및 건설용 중장비 차량은 상기 로드로 연계시키고, 상기 보행자는 상기 사이드워크에 연계시키며, 상기 자전거는 상기 사이드워크 및/또는 상기 로드와 연계시킨 것을 포함한다.
- [0031] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 딥러닝 객체 인식 장치는 딥러닝 모델에 기반한 객체 인식 소프트웨어를 저장한 메모리 및 상기 메모리의 상기 소프트웨어를 실행시키는 적어도 하나의 프로세서를 포함하고, 상기 딥러닝 모델은, 공통 백본 네트워크와 클래스별로 로스(loss)를 출력하는 복수의 헤드 네트워크를 포함한다.
- [0032] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 딥러닝 모델은 상기 클래스별 로스를 웨이트드 합산(weighted sum)해 최종 로스를 출력한다.
- [0033] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 최종 로스는 다이내믹 웨이트 평균(DWA; Dynamic Weight Average)법을 통해 얻는다.
- [0034] 본 발명의 적어도 하나의 실시예의 객체 인식 장치에서, 상기 데이터 베이스는 누씬(nuScenes) 및/또는 키티(KITTI) 데이터셋을 포함한다.

발명의 효과

- [0035] 본 발명의 실시예에 의하면, 객체의 클래스 불균형을 현실 상황에 맞는 데이터 증강을 통해 해소함으로써 인식 성능을 향상시킬 수 있다.

도면의 간단한 설명

- [0036] 도 1은 종래 그라운드-트루쓰 샘플링 기법에 의한 데이터 증강을 예시한다.
- 도 2는 본 발명의 하나의 실시예에 의한 객체 인식 장치 머신 러닝 방법의 플로우 차트를 나타낸다.
- 도 3은 본 발명의 하나의 실시예에 의한 딥러닝 네트워크 아키텍처를 나타낸다.
- 도 4는 본 발명의 하나의 실시예에 의한 딥러닝 네트워크의 학습 과정을 나타낸다.
- 도 5는 본 발명의 하나의 실시예에 의한 라이다 기반 딥러닝 객체 인식 장치
- 도 6 및 도 7은 비교예와 본 발명의 실시예들을 비교한 결과이다.

발명을 실시하기 위한 구체적인 내용

- [0037] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0038] 본 명세서에서 사용되는 접미사 "모듈" 및 "부"는 단지 구성요소 간에 명칭적인 구분만을 위해 사용되는 것일 뿐으로, 물리화학적으로 구분 또는 분리되어 있다가거나 그렇게 구분 또는 분리될 수 있음을 전제하는 것으로 해석되어서는 안 된다.
- [0039] “제1”, “제2” 등과 같이 서수를 포함하는 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되지는 않는다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.
- [0040] “및/또는”이라는 용어는 그 대상이 되는 복수 항목들의 여하한 조합의 경우를 모두 포함하기 위해 사용된다. 예컨대, “A 및/또는 B”는 “A”, “B”, “A 및 B” 등 3 가지 경우를 모두 포함하는 의미이다.
- [0041] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다.
- [0042] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0043] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0044] 또한, 유닛(Unit) 또는 제어 유닛(Control Unit)은 차량 특정 기능을 제어하는 제어 장치(Controller)의 명명에 널리 사용되는 용어일 뿐, 보편적 기능 유닛(Generic function unit)을 의미하는 것은 아니다. 예컨대, 각 유닛(Unit) 또는 제어 유닛(Control Unit)은 담당하는 기능의 제어를 위해 다른 제어기나 센서와 통신하는 통신 장치, 운영체제나 로직 명령어와 입출력 정보 등을 저장하는 메모리 및 담당 기능 제어에 필요한 판단, 연산, 결정 등을 수행하는 하나 이상의 프로세서를 포함할 수 있다.
- [0045] 먼저, 첨부된 도면에 대해 간략히 설명한다.
- [0046] 도 1은 종래 그라운드-트루쓰 샘플링 기법에 의한 데이터 증강을 예시하고, 도 2는 본 발명의 하나의 실시예에 의한 객체 인식 장치 머신 러닝 방법의 플로우 차트를 나타낸다. 또한, 도 3은 본 발명의 하나의 실시예에 의한 딥러닝 네트워크 아키텍처를 나타내며, 도 4는 본 발명의 하나의 실시예에 의한 딥러닝 네트워크의 학습 과정을 나타낸다. 도 5는 본 발명의 하나의 실시예에 의한 라이다 기반 딥러닝 객체 인식 장치이며, 도 6 및 도 7은 비

교예와 본 발명의 실시예들을 비교한 결과이다.

- [0047] 도 2를 참조하면, 본 발명의 하나의 실시예의 머신 러닝 방법에 의하면 먼저 원본 포인트 클라우드 데이터셋이 마련된다.
- [0048] 원본 데이터셋은 후술하는 데이터 베이스(DB)에 포함된 것일 수 있다.
- [0049] 또한, 원본 데이터셋은 후술하는 S100, S200, 및 S300 단계를 통해 증강되며, 이를 통해 증강된 학습 데이터셋이 확보된다.
- [0050] S100 단계는 데이터 증강을 위한 객체 데이터셋을 추출하는 과정이며, 이를 위해 라이다 포인트 클라우드 데이터 베이스(DB)로부터 적어도 하나의 데이터셋이 선택되고, 선택된 데이터셋으로부터 객체 데이터셋이 추출된다. 추출된 객체 데이터셋은 원본 데이터셋의 씬(scene)에 가상 데이터로서 추가되어 데이터 증강을 실현하는 가상 데이터셋으로 활용된다.
- [0051] 데이터 베이스(DB)는 각 포인트에 대해 라벨이 표기된 그라운드 쓰루스 클라우드 포인트 데이터셋을 포함한다. 예시적으로, 데이터 베이스(DB)는 오픈 라이다 데이터셋으로서, 누씬(nuScenes) 및/또는 키티(KITTI) 데이터셋을 포함할 수 있다.
- [0052] 가상 데이터셋은 그라운드 쓰루스 데이터셋의 포인트별 라벨 정보에 따라 추출될 수 있다.
- [0053] 그리고, 추출되는 가상 데이터셋은 동적 객체에 대한 것일 수 있다.
- [0054] 예시적으로, 가상 데이터셋은 보행자, 자전거, 모터사이클, 승용차, 트럭, 버스, 트레일러, 건설용 중장비 차량(construction vehicle) 중 적어도 하나의 동적 객체에 대한 것일 수 있다.
- [0055] 본 실시예에서는 새로운 데이터 증강법으로서 컨텍스츄얼 그라운드-트루쓰 샘플링(contextual ground-truth sampling) 기법을 제안한다. 컨텍스츄얼 그라운드-트루쓰 샘플링 기법은 가상 객체를 학습 데이터에 추가함에 있어 해당 상황(context)에 맞게 추가하는 방법이며, 기존 그라운드-트루쓰 샘플링 기법에 의해 비현실적인 위치에 가상 객체가 추가되는 문제를 해결한다.
- [0056] 이를 위해, 본 실시예에서 가상 데이터셋은 원본 데이터셋에 추가될 때, 객체별 연계 위치 정보를 기반으로 가상 데이터셋의 원본 데이터셋 내에서의 위치가 결정될 수 있다(S200).
- [0057] 객체별 연계 위치 정보는, 원본 데이터셋의 포인트 라벨 정보에 따라 결정될 수 있다.
- [0058] 예시적으로, 상기 포인트 라벨 정보는 사이드워크, 주행 가능 영역, 로드 중 적어도 하나에 상응하는 라벨을 포함할 수 있다.
- [0059] 또한, 예시적으로, 객체별 연계 위치 정보는 모터사이클, 승용차, 트럭, 버스, 트레일러 및 건설용 중장비 차량은 로드로 연계시키고, 보행자는 사이드워크에 연계시키며, 자전거는 사이드워크 및/또는 로드엔 연계시킨 것을 포함할 수 있다.
- [0060] 그리고, 그 연계 위치 정보에 따라, 가상 데이터셋을 원본 데이터셋에 추가한다(S300).
- [0061] 즉, 모터사이클, 승용차, 트럭, 버스, 트레일러 및 건설용 중장비 차량 등은 원본 데이터셋 내의 위치 중 로드 위로 배치되고, 보행자는 사이드워크 위로 배치되며, 자전거는 사이드워크 및/또는 로드엔 배치되도록 원본 데이터셋에 추가될 수 있다.
- [0062] 도 2에 있어서, 증강된 데이터셋은 원본 데이터셋에 각각 보행자에 해당하는 3개의 객체가 사이드워크 위로 추가되고, 각각 차량에 해당하는 3개의 객체가 도로 위로 추가된 것을 예시한다.
- [0063] 데이터 증강은 씬(scene)이 다른 복수의 원본 데이터셋에 대해 수행될 수 있으며, 이를 통해 원하는 양의 학습 데이터를 얻을 수 있다.
- [0064] 학습 데이터가 확보되면, 이를 이용해 딥러닝 객체 인식 장치에 대한 머신 러닝이 수행된다(S400).
- [0065] 본 실시예에 의한 딥러닝 모델은, 도 3에 예시되는 바와 같이, 공통 백본 네트워크와 클래스별로 로스(loss)를 출력하는 복수의 헤드 네트워크를 포함할 수 있다.
- [0066] 본 실시예의 딥러닝 모델은 컨텍스츄얼 그라운드-트루쓰 샘플링 기법에 의해 증강된 학습 데이터를 이용해 학습됨에 따라 클래스 불균형에 기한 그레디언트 불균형 문제가 일부 해소될 수 있으며, 또한, 추가적으로, 헤드별

로스들은 다이내믹 웨이티디 평균(DWA; Dynamic Weight Average) 법을 통해 웨이티드 합산되어 최종 로스로 출력됨에 따라 헤드별 그래디언트 불균형이 더욱 해소될 수 있다.

- [0067] DWA 법은 논문 "End-to-end multi-task learning with attention" (Liu, Shikun 외 2인, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019)에 소개된 방법이 이용될 수 있으며, DWA 법 자체는 종래에 알려져 있으므로 상세한 설명은 생략한다.
- [0068] 멀티 태스크 학습 기법인 DWA 법을 통해 각 head 별 학습을 평준화 하고 전반적인 학습 성능을 높일 수 있다.
- [0069] 도 4는 클래스의 갯수를 2로 가정하여 DWA 법이 적용되는 과정을 예시한다.
- [0070] 도 4에 보이는 바와 같이, 각 헤드에서 출력되는 로스의 크기를 이용하여 각 헤드별 가중치 α 를 업데이트하는 방식으로 현재 스텝 t 에서의 α 를 계산하기 위해 스텝 $t-1$ 과 $t-2$ 의 로스 값을 이용한다.
- [0071] 도 4의 수식을 보면 각 헤드별 가중치 α 는 해당 헤드의 로스 변화량이 클수록 큰 크기의 가중치를 가지게 된다. 가중치가 크다는 것은 최종 로스 계산에서 해당 헤드의 출력에 의한 비중이 커짐을 의미하고, 결과적으로 학습에 더 큰 영향을 줌을 의미한다.
- [0072] 가중치가 적용된 최종 로스를 통해 백프로퍼게이션(backpropagation)이 수행되고 공통 백본 및 모든 헤드의 네트워크 웨이트가 업데이트된다. 이런 식으로 매 스텝에서 가중치의 수정과 백프로퍼게이션을 통한 네트워크 학습이 진행된다.
- [0073] 본 실시예에서는 공통 백본 및 멀티 헤드로 구성되는 네트워크 부분의 내부 아키텍처를 자세하게 제시하지 않지만, 예시적으로 'PointPillars', 'PV-RCNN', 'Voxel-RCNN' 등 이미 공개된 네트워크가 이용될 수 있다.
- [0074] 한편, 본 발명의 하나의 실시예에 의한 라이다 기반 딥러닝 객체 인식 장치는, 도 5에 보이는 바와 같이, 객체 인식 소프트웨어를 저장한 메모리 및 메모리에 저장된 그 소프트웨어를 실행시키는 적어도 하나의 프로세서를 포함한다.
- [0075] 객체 인식 소프트웨어는 전술한 방법에 따라 학습된 딥러닝 모델에 기반한다.
- [0076] 또한, 라이다 기반 딥러닝 객체 인식 장치는 주변 환경에 대한 포인트 클라우드를 획득하는 라이다 센서를 포함할 수 있다.
- [0077] 그리고, 프로세서는 라이다 센서를 이용해 획득된 주변 환경에 대한 포인트 클라우드 데이터에 대해 객체 인식 소프트웨어를 실행시켜 그 인식된 결과를 출력한다.
- [0078] 본 실시예의 프로세서는, 예시적으로 컴퓨터, 마이크로프로세서, CPU, ASIC, 전자 회로(circuitry, logic circuits) 중 어느 하나 또는 그 조합일 수 있다.
- [0079] 또한, 본 실시예의 컴퓨터가 읽을 수 있는 기록 매체 또는 메모리에는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 저장 장치를 포함한다. 예시적으로, 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 마이크로 타입(micro type), 및 카드 타입(예컨대, SD 카드(Secure Digital Card) 또는 XD 카드(eXtream Digital Card)) 등의 메모리와, 램(RAM, Random Access Memory), SRAM(Static RAM), 롬(ROM, Read-Only Memory), PROM(Programmable ROM), EEPROM(Electrically Erasable PROM), 자기 메모리(MRAM, Magnetic RAM), 자기 디스크(magnetic disk), 및 광디스크(optical disk) 타입의 메모리 중 적어도 하나를 포함할 수 있다.
- [0080] 본 실시예에서, 컨텍스츄얼 그라운드-트루쓰 샘플링 기법은 딥러닝의 입력에 해당하는 학습데이터 자체를 변경하여 학습데이터 내 클래스 별 객체 수의 불균형을 해소하는 한다. 그리고, 다이내믹 웨이티디 평균법은 딥러닝 네트워크의 출력에 해당하는 로스를 계산하는 단계에서 클래스 별 로스 가중치를 변경하여 클래스 별로 네트워크 웨이트를 업데이트 하는 양을 컨트롤 한다.
- [0081] 컨텍스츄얼 그라운드-트루쓰 샘플링 기법은 학습데이터 자체를 변경시키기 때문에 클래스 불균형 문제를 보다 직접적으로 컨트롤할 수 있다. 다만, 이 기법은 미리 설계된 룰을 통해 학습데이터 증강이 일어나므로 학습 과정 중 네트워크 웨이트가 수렴하는 과정에서 발생하는 변수에는 대응할 수 없다.
- [0082] 다이내믹 웨이티디 평균법의 경우 이론적으로 매 스텝마다 계산되는 헤드별 로스를 통해 수학적 계산으로 헤드별 로스 웨이트를 업데이트하기 때문에 네트워크가 수렴하는 과정 중 발생하는 성능 편향 문제를 해소할 수 있다. 다만, 그 영향성이 학습데이터를 직접 조작하는 것에 비해서 낮을 수 있고, 원하는 성능을 얻기 위해 컨트롤

물라기가 어려울 수 있다.

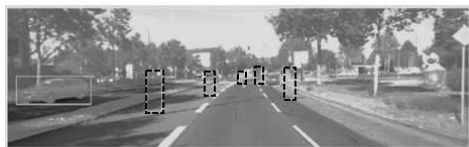
- [0083] 도 6은 비교예들 및 본 발명의 실시예들을 이용한 인식 결과를 정량적으로 나타낸다.
- [0084] 비교예 1은 'PointPillars'를 이용하고 종래기술에 학습 데이터를 증강한 경우이고, 실시예 1은 'PointPillars'를 이용하고 본 실시예에 의한 데이터 증강이 적용된 경우이다.
- [0085] 도 6에 보이는 바와 같이, 차량(car), 보행자, 및 자전거에 있어서, 3D 데이터는 물론 2차원 데이터인 버드아이뷰(BEV; Bird Eyes View)에 대해서도 실시예 1의 인식 성능이 향상된 것을 알 수 있다.
- [0086] 한편, 비교예 2는 'PV-RCNN'를 이용하고 종래기술에 학습 데이터를 증강한 경우이고, 실시예 2는 'PV-RCNN'를 이용하고 본 실시예에 의한 데이터 증강이 적용된 경우이다.
- [0087] 실시예 2의 경우 또한, 모든 클래스들에서 비교예 2보다 인식 성능이 향상된 결과를 나타내고 있다.
- [0088] 비교예 3은 'Voxel RCNN'를 이용하고 종래기술에 학습 데이터를 증강한 경우이고, 실시예 3은 'Voxel RCNN'를 이용하고 본 실시예에 의한 데이터 증강이 적용된 경우이다.
- [0089] 실시예 3의 경우 또한, 비교예 3보다 인식 성능이 향상된 결과를 나타내고 있다.
- [0090] 한편, 도 6의 비교예들 및 실시예들에 사용된 데이터셋으로 키티 데이터셋이 이용되었다.
- [0091] 또한, 도 7은 비교예 4와 실시예 4의 인식 결과를 정성적으로 비교하고 있다.
- [0092] 비교예 4 및 실시예 4에 이용된 데이터셋은 누씬(nuScenes) 데이터셋이며, 도 7에 보이는 바와 같이, 종래기술을 따르는 비교예 4의 경우 실시예 4에 비해 오인식 데이터가 포함됨을 알 수 있다.

도면

도면1



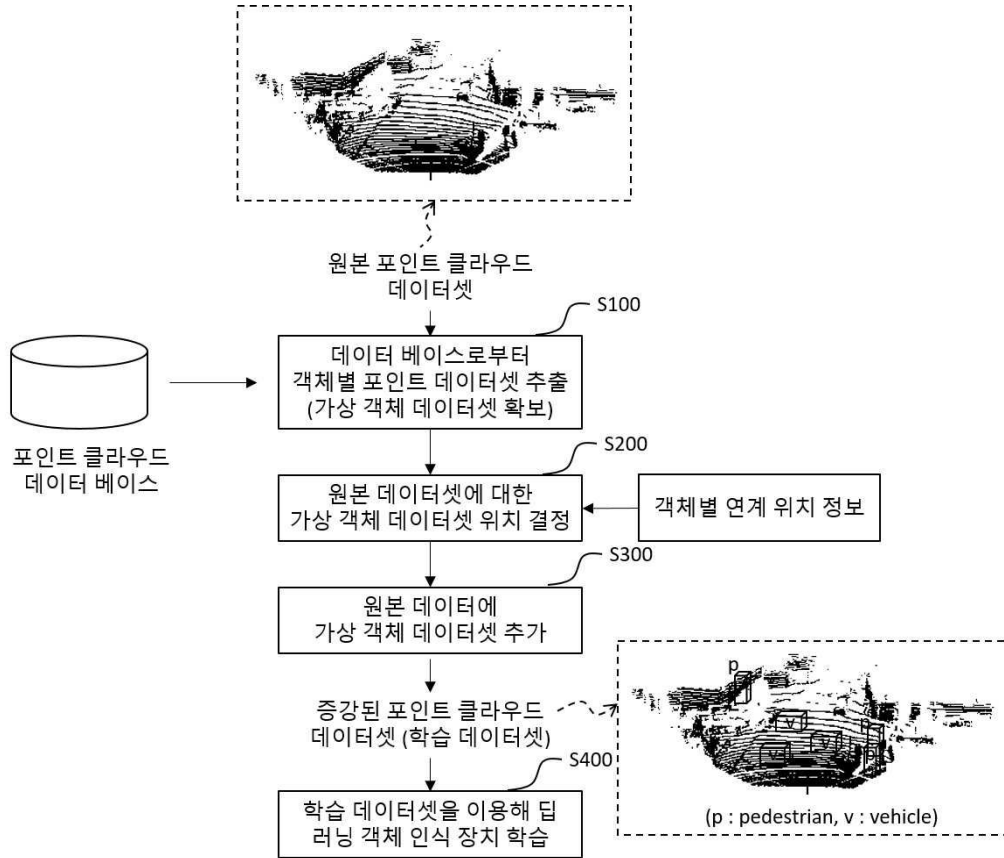
(a)



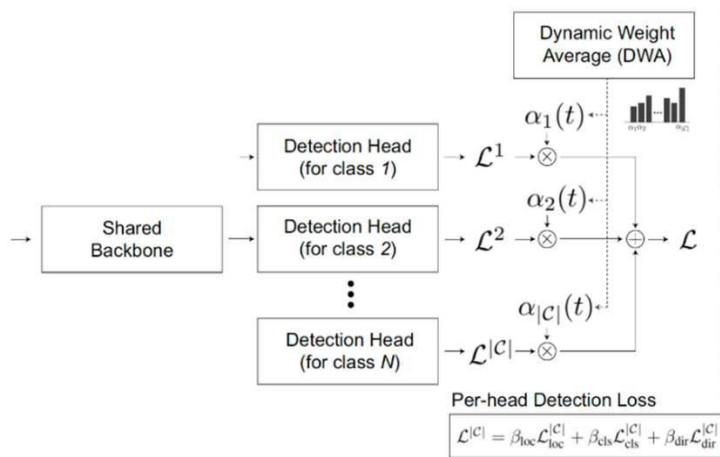
□ 보행자

(b)

도면2



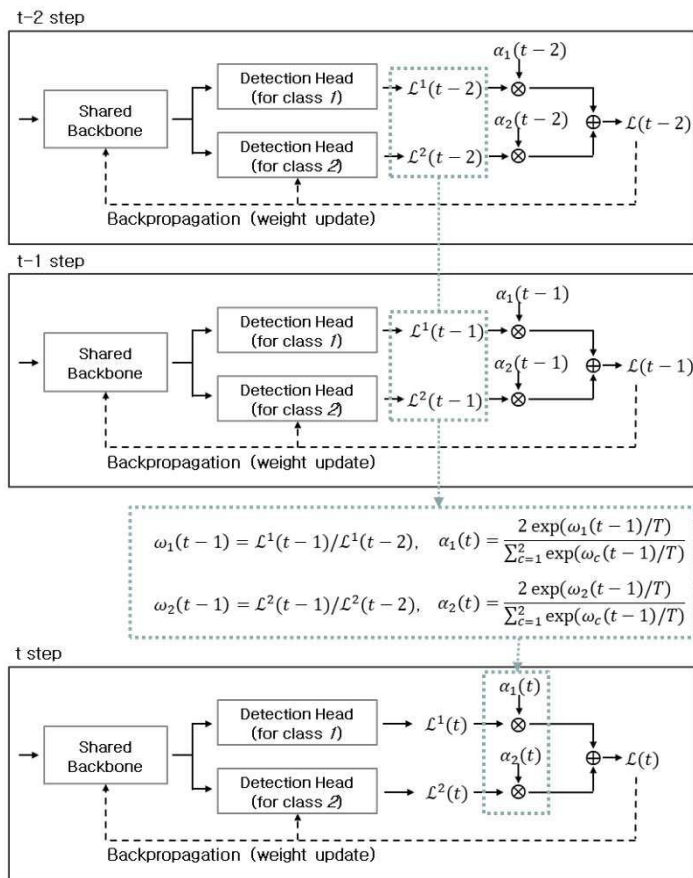
도면3



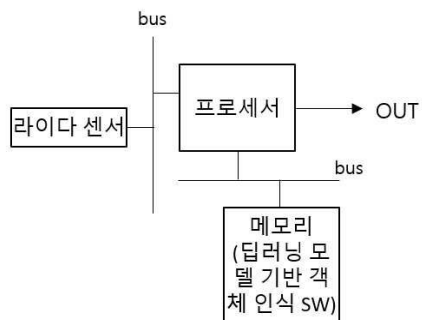
$$\alpha_c(t) = \frac{|C| \exp(w_c(t-1)/T)}{\sum_c \exp(w_c(t-1)/T)}$$

$$w_c(t-1) = \mathcal{L}_c(t-1) / \mathcal{L}_c(t-2)$$

도면4



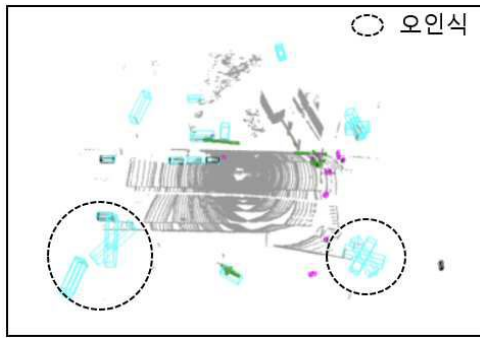
도면5



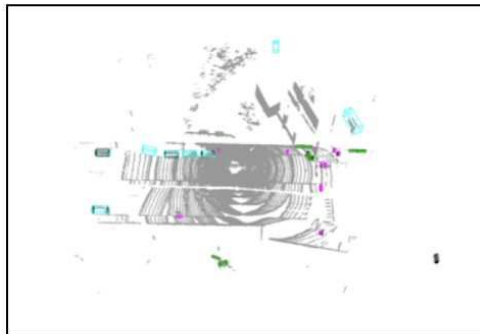
도면6

	Car (0.5)		Pedestrian (0.25)		Cyclist (0.25)		Avg.	
	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑	3D↑	BEV↑
비교예 1 실시예 1	94.50 94.99 (0.45↑)	94.66 95.09 (0.40↑)	70.40 70.93 (0.53↑)	70.72 71.23 (0.51↑)	71.03 72.80 (1.77↑)	71.03 72.80 (1.77↑)	78.64 79.57 (0.93↑)	78.80 79.71 (0.91↑)
비교예 2 실시예 2	94.42 94.74 (0.32↑)	96.20 96.50 (0.30↑)	75.32 75.50 (0.18↑)	75.60 75.92 (0.32↑)	79.15 83.25 (4.10↑)	79.15 83.25 (4.10↑)	82.96 84.50 (1.54↑)	83.65 85.22 (1.57↑)
비교예 3 실시예 3	94.91 97.08 (2.17↑)	96.66 97.31 (0.65↑)	-	-	-	-	94.91 97.08 (2.17↑)	96.66 97.31 (0.65↑)

도면7



(a) 비교예 4



(b) 실시예 4