

딥러닝 기반의 군집 자율주행 시스템 구현

김현열, 김준수, 이인환, 이훈

부경대학교 정보통신공학과

{khy4088, jssb0101, dlsghks1227}@pukyong.ac.kr, hlee@pknu.ac.kr

Deep Learning-Based Self-Driving Systems for Swarm Vehicles

Hyunyeol Kim, Junsu Kim, Inhwan Lee, and Hoon Lee

Dept. Information and Communications Eng., Pukyong National University

요약

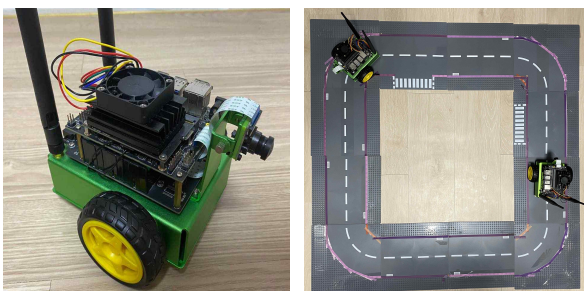
본 논문에서는 딥러닝 기반의 군집 자율주행 시스템을 제안한다. Jetson Nano 임베디드 보드를 활용하여 다수의 자율주행체를 구현하고, 미리 제작된 도로 주행 환경을 학습시켜 실시간으로 자율주행하는 테스트베드를 구축한다. 성공적인 군집 자율주행 생태계를 구성하기 위해서는 자율주행체가 도로의 형태를 파악하고 스스로 주행하도록 하는 Road Following 기능과, 선행하는 자율주행체와의 거리를 인식하여 동작을 제어하는 Object Following 기능이 필요하다. 각 기능을 효과적으로 수행하는 합성곱 신경망 기반 기술을 제안한다. 자율주행체가 실시간으로 촬영하는 영상에서 주행해야 할 지점의 픽셀 위치를 추정하도록 하는 ResNet18 기반의 Road Following 기술을 활용한다. Object Following은 YOLOv5를 기반으로 선행 주행체의 크기를 인식하여 후속 주행체의 속도를 제어한다. 두 딥러닝 모델을 모두 저전력 임베디드보드에 구현하기에는 무리가 있으므로 비교적 연산 복잡도가 높은 Object Following기능을 엡지 서버가 담당하도록 시스템을 구축하여 군집 자율주행체의 중앙제어 프로토콜을 구성하였다. 모의실험을 통해 군집 자율주행체의 Road Following 및 Object Following 기능이 모두 잘 수행됨을 확인한다.

I. 서론

최근 빠르게 발전하는 딥러닝 기술을 기반으로 자율주행 구현 연구가 급속도로 진행되고 있다. 특히, 다수의 이동체가 군집을 이루어 스스로 주행하는 군집 자율주행 기술이 물류분야를 기점으로 최근 각광받고 있다. 이러한 최신 연구동향을 기반으로 본 논문에서는 군집 자율주행 시스템을 구축하고 그 실효성을 파악한다. 저비용 테스트베드 구성을 위해 자율주행체를 저전력 임베디드보드로 설계한다. 실시간 영상 정보를 기반으로 스스로 주행하는 이동체를 구현하기 위해 딥러닝 기법을 활용한다. 도로를 주행하고 장애물을 판단하는 Road Following 기능과, 선행 이동체를 인식하는 Object Following 기능을 합성곱 신경망으로 처리한다.

저전력 임베디드보드로 다수의 신경망의 연산을 처리하면 매우 낮은 프레임 처리 속도를 보여, 빠른 주기로 주행환경을 인식해야 하는 군집 자율주행환경에서 사용에 어려움이 있다. 이를 해결하기 위해 높은 성능이 필요한 Object Following 기능을 엡지서버에서 분담하는 Mobile Edge Computing (MEC) 네트워크를 구성한다. 실시간 영상을 엡지서버에 전송하고, 객체 검출 신경망을 통한 결과값을 도출하여 제어 명령 구성 및 전송받는 중앙제어 시스템을 구현한다.

II. 군집 자율주행 테스트베드



(a) 자율주행체 하드웨어 구현

(b) 자율주행 실험 환경

그림 1. 군집 자율주행 테스트베드

본 논문에서는 그림 1과 같이 선행하는 Host Car를 다수의 Sub Car가

뒤따르는 군집 자율주행 시스템을 구현한다. 자율주행체는 NVIDIA Jetson Nano 임베디드 보드로 구성된다 (그림 1(a) 참고). CSI 카메라를 연결하여 주행 영상을 실시간으로 촬영하고, 실시간 영상 전송 및 제어 신호 수신을 위해 Intel Wireless-AC8265 무선랜카드를 사용한다. 두 개의 DC 모터를 설치하여 좌우 바퀴의 동작을 제어한다. 주행 환경은 그림 1(b)와 같이 직선로 및 좌회전 선로가 다양하게 포함되도록 제작한다. 자율주행체를 보다 효과적으로 훈련하기 위해 실선을 테이핑하여 점선과 실선의 특징점을 구분한다.

III. 딥러닝 기반 군집 자율주행 시스템

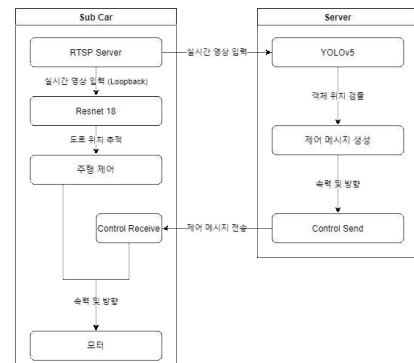


그림 2. 제안하는 딥러닝 기반 군집 자율주행 시스템

그림 2은 제안하는 시스템의 구조도를 나타낸다. 군집 자율주행을 위해서는 다음의 두 가지 영상처리 기능이 필요하다: 1) Sub Car가 도로를 스스로 인식하고 주행하는 Road Following 기능, 2) 선행하는 Host car를 인지하는 Object Following 기능. Road Following은 도로를 실시간으로 인식하고 라인을 벗어나지 않도록 제어하는 동작이 필요하므로 매우 낮은 지연시간을 요구한다. 입력 영상으로부터 도로 라인의 위치를 추정하는 비교적 간단한 회귀 기술로 구현 가능하므로 자율주행체에서 계산을 직접 수행하는 것이 유리하다. 이와는 반대로 Host Car를 인식하는 Object Following은 영상에서 객체의 위치를 검출함과 동시에 객체의 분류를 수행하는 고난도의 multi-task 학습 기술이 필요하므로 높은 계산 복잡도를

요구한다. 하지만 선행 Host Car의 인식이 매우 빠른 주기로 수행될 필요는 없으므로 Object Following 계산을 서버에 분배하는 MEC 개념을 활용한다. 각 기능의 상세한 동작 연산은 다음과 같다.

3-1. 실시간 영상 전송 기능

Object Following 연산을 MEC로 처리하기 위해 위해 자율주행체에서 촬영한 영상을 RTSP 프로토콜을 활용하여 엣지서버로 전송한다. 수신한 영상을 바탕으로 서버는 객체를 인식하고, 해당 자율주행체의 주행 제어 생성하여 전송한다. Object Following 관련 내용은 3-3절에서 후술한다.

3-2. Road Following 기능

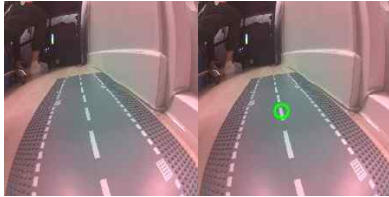


그림 3. Road Following 학습 데이터 예시

자율주행체는 실시간으로 촬영한 영상을 합성곱 신경망 기반의 ResNet18 모델 [1]로 처리하여 자신이 주행할 도로의 위치를 추정한다. ResNet18 신경망을 학습하기 위해 그림 3과 같이 영상에서 이동할 위치 (x, y) 를 레이블로 수집하여 지도학습을 수행한다. 이때 (x, y) 는 영상의 중앙이 원점이 되도록 처리한다. 효과적인 학습을 위해 전이학습(transfer learning) 개념을 활용하여 미리 훈련된 ResNet18 모델을 초기 점으로 사용한다. 훈련은 GPU가 설치된 PC에서 진행하고, 자율주행체에는 학습된 신경망을 메모리에 저장하여 실시간 회귀를 구현한다.

ResNet18로 추정된 좌표정보를 활용하여 주행체의 좌·우 모터 속도를 제어한다. 주행할 위치의 y 좌표가 영상 높이의 20% 미만이라면 도로가 장애물 등에 의해 막혀있는 것으로 판단하고 운행을 정지한다. 그 외에는 주행이 가능한 상황으로 인지하여 이동할 각도를 계산하고, 이를 기반으로 각 모터의 속도를 결정한다.

3-3. Object Following 기능



그림 4. YOLO기반의 객체 검출 예시

선행 차량을 탐지하기 위해 YOLOv5[2]를 사용한다. 영상을 다수의 그리드 영역으로 나누고, 각 그리드 영역에서 물체가 있을 만한 영역에 해당하는 다수 bounding box의 중심점, 너비, 높이를 예측한다. 각 box의 신뢰도를 계산하고, 그리드별로 각 클래스에 대한 사후확률을 추정한다. 다수의 개별 그리드에 대한 bounding box의 탐지 및 분류를 동시에 수행하는 고복잡도 연산이 요구되므로 YOLOv5 연산은 엣지서버가 분담한다. 탐지한 객체의 크기가 영상의 80% 이상을 차지하면 주행체에게 정지하는 제어 메시지를 전송하고, 그렇지 않으면 각 자율주행체가 Road Following 기능을 활용하여 개별적으로 주행 업무를 수행한다.

3-4. 자율주행 차량 동작

3-2와 3-3에서 언급 했던 Road Following, Object Following 기능을 결합하기 위하여, 우선순위를 적용한다. Road Following을 적용하여 도로 주행을 하는 도중, 중앙서버로부터 명령이 도착하면, 중앙서버의 명령을 우선으로 동작하도록 구성한다.

IV. 모의실험 결과

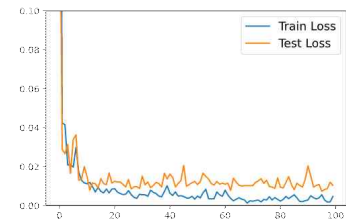
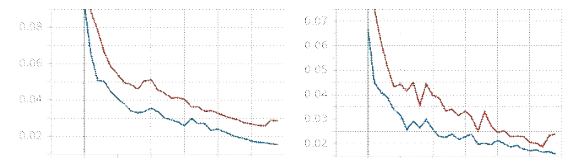
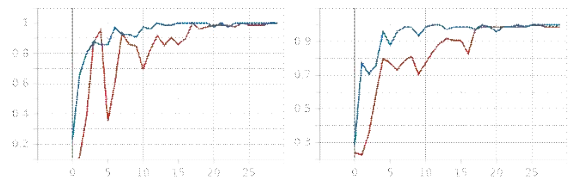


그림 5. ResNet18 신경망의 학습 성능

본 절에서는 Road Following 및 Object Following 기능을 수행하는 ResNet18과 YOLOv5 신경망의 학습 성능을 제시한다. Road Following을 위해 그림 4와 같은 학습 영상을 200장 수집하고, 훈련에 180장, 검증에 20장을 활용한다. 학습에는 Adam 알고리즘을 사용하고, batch size는 4, learning rate는 0.001로 설정한다. 손실함수는 평균 제곱 함수를 사용하였다. ResNet18 학습 과정의 손실함수 성능을 그림 5에 도시한다. 훈련 및 검증 데이터 집합에 대한 손실함수가 모두 적절히 감소하여 학습이 잘 수행된 것을 알 수 있다.



(a) Bounding box 예측 손실함수 (좌: 학습 데이터, 우: 검증 데이터)



(b) 좌: 예측 정밀도, 우: 재현율

그림 6. YOLOv5 신경망의 학습 성능

YOLOv5 모델을 훈련하기 위해 300장의 데이터 수집하고, 학습 성능향상을 위해 data augmentation을 활용하여 총 900장의 영상을 활용한다. 그림 6는 YOLOv5 신경망의 학습 성능을 나타낸다. 파란색 실선이 data augmentation을 적용한 경우, 빨간색 실선이 적용하지 않은 경우의 성능을 의미한다. 그림 6(a)는 bounding box 예측 관련 손실함수를 도시한다. Data augmentation을 활용한 경우 학습 데이터 및 검증 데이터 집합에서 모두 성능이 향상된다. 그림 6(b)의 예측 정밀도 및 재현율 성능은 두 경우 모두 비슷한 지점으로 수렴하나, 학습 속도 및 안정성이 data augmentation을 적용한 상황에서 크게 향상됨을 확인할 수 있다.

V. 결론

군집 자율주행 시스템을 구현하기 위해 저전력 임베디드보드, 엣지 서버, 그리고 딥러닝 기술을 활용하였다. 실시간 영상처리를 위해 비교적 간단한 Road Following 기능을 자율주행체에 이식하고, 고복잡도 객체 탐지 연산이 요구되는 Object Following 기능은 엣지 서버가 담당하여 계산을 분담하는 MEC 네트워크를 제안하였다. 모의실험을 통해 군집 자율주행이 원활히 수행됨을 확인하였다.[3]

참고 문헌

- [1]He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [2]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 (pp. 779-788).
- [3]<https://youtu.be/1i-T2uQ4qg8>