

자율주행 시스템을 위한 효율적인 객체 탐지 모델

김 시 온*, 이 선 영°

Efficient Object Detection Models for Autonomous Driving System

Si-On Kim*, Sun-Young Lee°

요 약

대용량 데이터 처리와 머신 러닝을 통한 물체 탐지를 통해 자율주행기술 또한 발전을 이루었다. 자율주행을 위해 차량에 수많은 센서와 ECU가 장착되었으며, 각 센서와 ECU 간 통신을 위한 케이블들이 장착되어 차량의 무게 증가와 연비 감소라는 문제가 발생하였다. 가장 오래도록 사용된 차량용 네트워크인 CAN은 최대 전송속도 1Mbps로 실시간 영상 전송에 알맞지 않다는 단점이 있다. 이를 해결하기 위해 사용되는 차량용 이더넷은 최고 전송속도 100Mbps로 영상 데이터를 전송할 수 있으나, 고해상도의 영상 전송이 힘들다는 단점이 있다. 이 단점을 해결하기 위해 본 연구에서는 검출 물체 외 데이터 삭제 후, 검출 물체의 영역 데이터를 객체 인코딩하여 객체 탐지 성능을 유지하면서 데이터 전송 시간을 줄이는 방법을 제안하고 실험을 통하여 평가하였다. 그 결과 FHD 환경에서 41.02%, 4K 환경에서 62.8% 전송 시간이 줄어들었음을 확인하였다.

키워드 : 자율주행시스템, 객체 인코딩, YOLOv5, 100Base-T1

Key Words : autonomous Driving System, Object Encoding, YOLOv5, 100Base-T1

ABSTRACT

The development of autonomous driving technology has also been accelerated by object detection using big data processing and machine learning. Numerous sensors and ECUs were installed in the vehicle for autonomous driving, and cables for communication between each sensor and ECU were installed, causing problems such as increasing the weight of the vehicle and reducing fuel efficiency. CAN, the longest-used vehicle network, has the disadvantage that it is not suitable for real-time video transmission with a maximum transmission speed of 1 Mbps. The Ethernet for vehicles used to solve this problem can transmit image data at a maximum transmission speed of 100 Mbps, but it has the disadvantage that it is difficult to transmit high-resolution images. In this study, after deleting non-detection object data, a method of reducing data transmission time while maintaining object detection performance by object encoding the area data of the detection object was proposed and evaluated through experiments. As a result, it was confirmed that the transmission time was reduced by 41.02% in the FHD environment and 62.8% in the 4K environment.

* 본 연구는 정부(MSIT)의 재원으로 한국연구재단(NRF)의 지원을 받아 수행된 연구임(No.RS-2024-00346749).

• First Author : Soonchunhyang University Department of Information Security Engineering, gu100433@naver.com, 학생회원

° Corresponding Author : Soonchunhyang University Department of Information Security Engineering, sunlee@sch.ac.kr, 종신회원
논문번호 : 202406-112-E-RN, Received June 4, 2024; Revised September 4, 2024; Accepted October 10, 2024

I. 서 론

많은 센서와 데이터 처리, 딥 러닝 기술을 발달로 사용자가 직접 조작하지 않아도 되는 자율주행차량 기술이 발전해 왔다. 특히, 자율주행 기술 중 장애물을 찾아 판단하는 객체 인식 기술의 연구가 활발히 이루어져 왔다^[1]. 정확한 탐지를 위해 고해상도의 영상 데이터를 수집할 수 있는 카메라는 개발되었으나, 기존 차량에 탑재된 CAN(Controller Area Network) 만으로는 대용량 데이터의 처리와 많은 센서들이 수집한 데이터를 전송하는 것이 어려워졌다^[2,3]. 차량에 장착된 센서와 전자장치들의 상호작용을 위한 차량 내부 네트워크에는 LIN, CAN, MOST, FlexRay, Ethernet이 존재한다. 이중, CAN은 수많은 ECU(Electronic Control Unit) 제어를 위한 대표적인 네트워크이다. 현재 ECU 내부로부터 생성되는 데이터는 초당 50Mb이며, 레이더와 영상 센서는 초당 100MB, 그 외 여러 계측 데이터를 합치면 전송되는 데이터는 초당 1GB에 이르며, 기술이 개발될 수록 전송되는 데이터의 크기는 더 커진다.

CAN 데이터 압축을 활용하거나, 기존 CAN 네트워크보다 빠른 속도를 지닌 CAN-FD와 FlexRay 같은 차량용 네트워크가 연구되었으나, 고해상도 영상을 처리하기에 역부족이었다^[4,5].

이를 해결하기 위해 이더넷을 개조한 차량용 이더넷인 100Base-T1과 1000Base-T1이 연구되고 있다^[6]. 100Base-T1은 현재 많은 차량에 탑재되고 있지만, 많은 데이터 처리를 필요로 하는 엔터테인먼트 시스템과 정보통신 시스템에만 사용되고 있다. 1000Base-T1은 100Base-T1보다 빠른 속도를 제공하나, 아직 많은 차량에 탑재되지 않았다. 그 이상의 속도를 지원하는 차량용 이더넷은 상용화까지 많은 시간이 필요하고 더 많은 연구가 필요하다. 따라서, 높은 객체 탐지 정확도를 위해 고해상도 영상을 100Base-T1과 1000Base-T1 환경에서 실시간 처리가 가능한 연구가 필요하다.

컴퓨터 과학과 객체 지향 프로그래밍에서 객체 인코딩(Object Encoding)은 데이터를 특정 형식으로 변환하여 객체를 데이터베이스에 저장하거나 네트워크를 통해 전송하는 데 사용한다. 데이터 전송에 객체 인코딩을 적용하면 원본 데이터를 전송하는 것 대비 더 적은 자원을 사용하므로 빠른 전송이 가능하다. 자율주행 시스템과 같이 실시간 데이터 처리가 필요한 문제를 객체 인코딩을 통해 해결하는 방법에 대해 연구하였다.

본 연구에서는 탐지된 객체의 정보를 가진 영역 외 데이터를 삭제하고, 객체 인코딩을 적용한 데이터를 전송함으로써 데이터 전송 시간을 단축시켰다. 이를 통해

고해상도 영상 데이터를 실시간으로 처리하는 것이 가능하다.

본 논문은 2장 관련 연구, 3장 제안 시스템, 4장 실험, 5장 결론으로 구성된다.

II. 관련 연구

2.1 차량용 이더넷

과거 차량 주요 통신망은 CAN이었지만, 차량의 안전성과 편의성을 위해 ADAS(Advanced Driver Assistance Systems)와 ECU 개수의 증가로 데이터량이 증가하여 차량용 이더넷이 등장하였다. 초기 차량용 이더넷은 IT에서 사용하는 이더넷을 그대로 적용하고자 하였으나, 온도, 신뢰성, 방수 등 요구사항이 맞지 않아 사용하지 못하였다.

일반적인 표준 이더넷인 100Base-TX^[7]는 2pair 혹은 4pair 구성의 전선을 사용하며 100Mbps의 전송속도를 가진다. 차량용 이더넷인 100Base-T1은 100Base-TX와 동일한 100Mbps의 전송속도를 가지나, 1pair 구성의 전선을 사용하여, 신뢰성을 높이고 비용을 줄이는 구조로 설계되었다. 차량용 이더넷은 표준 이더넷과 물리 계층 구조의 차이를 제외하면 동일한 구조를 가지며, 송신과 수신 채널이 분리되어 동시에 제어기가 송수신 가능한 점과 1:1 통신으로 네트워크를 구성하는 점, 주소 기반 통신으로 데이터를 전송하는 등 이더넷의 특징을 모두 만족한다^[6].

2.2 객체 인코딩 (Object Encoding)

객체 인코딩은 컴퓨터 프로그램에서 객체를 표현하고 전송 및 저장하기 위한 방법 중 하나로, 추상적 개념인 데이터와 메소드를 컴퓨터가 이해할 수 있도록 변환하는 프로세스이다. 주로 데이터베이스 최적화와 객체 지향 프로그래밍을 통한 데이터 생성 및 전송에 사용된다^[8,9].

객체 인코딩 기술의 데이터 저장과 전송 최적화에 특화된 특성을 활용하여 주로 데이터 전송 API와 데이터 찾기의 최적화에 사용된다.

본 연구에서는 네트워크 전송을 위한 패킷 생성 단계에서 객체 인코딩을 적용하여, 데이터 패킷을 분할하고 {탐지된 객체 정보, 객체가 탐지된 영상 데이터} 형태로 생성한다.

2.3 자율주행판단

자율주행 자동차는 사람의 조작과 동일한 인지, 판단, 제어 3단계를 거친다^[10,11]. 인지 단계에서는 레이더,

LiDAR(Light Detection And Ranging), GPS, 카메라 등 차량에 장착된 센서와 V2V(Vehicle-to-Vehicle), V2I(Vehicle-to-Infrastructure) 등 외부 통신을 통해 주변 상황을 수집하고 인지한다. 판단 단계에서는 인지된 정보를 해석하고 주행 상황을 판단하여 사람 대신 차량 스스로 운행한다. 제어 단계에서는 제어 시스템을 통해 결정된 판단을 실제 주행에 반영하여 운전 시스템을 제어한다.

자율주행의 판단 단계에서는 전후방 차량인식을 위한 레이더, 주변 환경 360도 인식을 위한 LiDAR, 신호 등, 차량, 보행자 등을 구별하기 위한 카메라와 차량의 경로, 위치 판단을 위한 GPS 등이 사용된다.

이중, 주행 판단과 장애물 회피 판단 역할을 위해 레이더, LiDAR, 카메라는 많은 양의 데이터를 수집하는 센서이다. LiDAR는 주변 360도를 인식하여 물체의 3D 형태 이미지와 물체의 거리 측정에 사용되며, 카메라는 물체 인식과 물체의 거리 측정에 사용된다.

카메라에서 수집된 영상 데이터를 머신 러닝을 통해 물체를 인식하고, LiDAR를 통해 수집된 3D 데이터와 조합하여 자율주행 판단을 내리게 된다.

2.4 You Only Look Once (YOLO)

객체 탐지는 이미지나 비디오 내 객체의 위치와 종류를 탐지하고 식별하는 기술이다. You Only Look Once(YOLO)^[12]는 객체 탐지 분야에서 1-Stage Object Detection 방식을 사용하는 대표 모델이다. 객체 탐지 모델은 1-단계 검출기와 2-단계 검출기로 분류된다. 기존 객체 탐지 방식인 슬라이딩 윈도우 방식은 이미지에서 모든 영역을 여러 크기의 윈도우로 탐색하는 비효율적인 방식이었으나, 물체가 있을만한 영역을 빠르게 찾아내는 알고리즘인 region proposal가 개발되며 객체 탐지 연구가 발전하였다. 2-단계 검출기는 region proposal 후, 분류가 순차적으로 이루어지는 방식으로 정확도가 높으나 속도가 느리다. 반면, YOLO가 사용하는 1-단계 검출기는 region proposal과 분류를 동시에 진행하여 2-단계 검출기보다 정확도는 낮으나, 탐지 속도가 빨라 실시간 객체 탐지를 위해 주로 사용된다.

YOLO는 객체 탐지를 위해 다음과 같은 4단계로 실행된다. 첫 번째, 입력된 이미지를 $N \times N$ 크기로 그리드화 한다. 두 번째, 각 그리드 마다 이미지 분류 및 지역화를 진행한다. 세 번째, 객체의 위치를 확인하고 객체에 바운딩 박스를 그린다. 네 번째로 바운딩 박스 내 객체의 클래스와 확률을 추론하고, 결과를 출력한다.

이러한 YOLO는 정확도에 특화된 모델, 실시간 처리 속도에 특화된 모델 등 환경에 적합한 모델을 선택할

수 있다.

최근에는 YOLO의 실시간 객체 탐지 성능을 활용하여, 무인기에 사용하거나^[13], 선박 탐지^[14], 교통 약자 탐지^[15] 등 연구에서 활용되고 있다.

III. 제안 시스템

기존 자율주행차량의 주행 결정은 데이터 수집 단계, 객체 탐지 단계, 3D 모델링 단계, 주행 결정 단계로 구성된다. 1단계는 데이터를 수집하는 단계로, LiDAR와 카메라를 통해 주행 결정에 필요한 데이터를 수집한다. 2단계는 객체를 탐지하는 단계로, 카메라를 통해 수집된 정보에서는 객체의 종류와 위치, LiDAR를 통해 수집된 데이터에서는 물체와의 거리를 수집한다. 3단계는 3D 모델링단계에서는 2단계에서 얻은 데이터를 바탕으로 3D 모델링을 진행하여, 주행 결정을 내리게 된다. 마지막으로 4단계에서는 주행 결정을 전송한다.

기존 시스템에서 영상 정보와 LiDAR 정보를 전송하기 위해 100Base-T1 기반의 차량용 이더넷을 사용하는 데, 최대 전송 속도는 100Mbps로 고해상도 영상 데이터 전송에 적합하지 않다. 따라서, 고해상도 영상 전송을 위해 1Gbps 전송 속도가 특징인 1000Base-T1을 사용하거나, 전송 데이터의 크기를 줄일 필요가 있다.

본 연구에서는 그림 1과 같이 카메라가 수집한 데이터를 객체 탐지를 수행하는 처리 장치로 전송하는 것이 아닌, 수집과 동시에 객체 탐지를 수행하는 시스템을 제안한다.

객체 탐지를 마친 영상과 객체 정보는 화면 내 객체 class와 영역을 표시하므로 원본 영상 데이터보다 크기가 커지는데, 이 문제를 해결하기 위해 본 논문에서는 다음과 같은 방법을 제안한다. (1) 객체 탐지, (2) 객체의 영역 외 배경 정보 삭제, (3) 객체 정보와 영상정보에

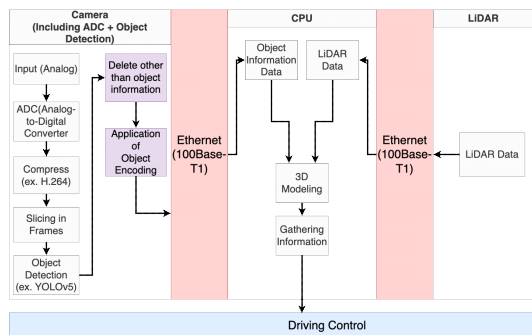


그림 1. 제안하는 자율주행 판단 시스템

Fig. 1. Proposed autonomous Driving Determination System

객체 인코딩 적용, (4) 이더넷을 통해 3D 모델링 장치로 전송 순으로 자율주행 판단을 위한 데이터를 생성 및 전송한다.

그림 2는 기존 주행 판단 시스템과 본 논문에서 제안하는 주행 판단 시스템을 비교한 그림이다. 기존 주행 판단 시스템에서 사용하는 원본 영상과 검출된 객체의 이미지와 다르게 제안 시스템에서는 객체 탐지 후, 객체 외 정보 삭제와 객체 인코딩을 적용한 데이터를 사용한다.

객체 탐지에 사용된 모델은 YOLOv5로 실시간 객체 탐지에 적합한 모델이다. 학습에 사용된 데이터는 Roboflow에서 공개한 Udacity Self Driving Car Dataset을 사용하였으며, 그림 3은 학습에 사용된 데이터의 Class Balance이다.

객체 탐지와 정보 삭제가 완료된 영상 데이터는 MPEG-TS 프로토콜을 통해 전송된다.

일반적인 MPEG-TS의 패킷 길이는 188바이트지만 오류 정정 바이트를 추가하여 204 혹은 208바이트로 구성하는 것이 가능하다. 또한, 오디오 데이터 혹은 비디오 데이터끼리 2개의 패킷을 묶어 44바이트의 오류 정정 정보와 함께 438바이트 길이의 MPEG-TS 패킷을 전송할 수 있다. 본 연구에서는 비디오 데이터 2개를

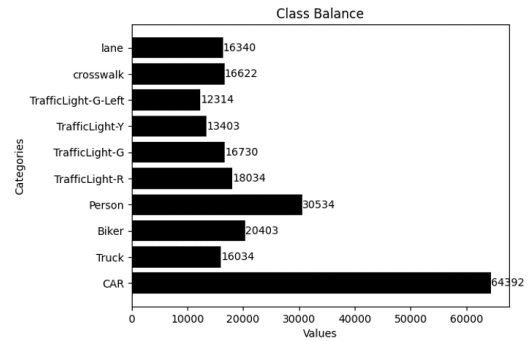


그림 3. 학습에 사용된 데이터의 Class Balance
Fig. 3. Class Balance used for learning

묶어서 44바이트의 오류 정정 정보를 추가하여 총 438바이트의 길이를 가지는 패킷을 전송한다. 객체 탐지된 영상과 객체 정보를 동시 전송하기 위해 객체 인코딩을 적용할 필요가 있다. 따라서, 데이터를 담을 수 있는 184바이트 길이의 페이로드를 29바이트의 객체 정보 4바이트의 구분 필드, 151바이트의 영상 정보로 분할하여 하나의 패킷으로 구성한다.

IV. 실험

4.1 시스템 설계

본 연구에서 제안하는 시스템이 고해상도 영상의 물체 탐지 및 전송이 가능함을 증명하기 위해 동일한 길이의 같은 영상을 서로 다른 해상도로 나눠 비교한다. 5분 실제 주행 환경을 담은 차량 블랙박스 영상을 사용한다. 다만, 1080p 영상과 4K 영상으로 해상도에서 차이를 둔다. 그림 5는 그림 1의 설계를 바탕으로 실험을 진행하였다. PC 1은 카메라와 객체 검출 역할을 하며, 검출된 객체의 영역 외 데이터 제거와 전송 데이터에 객체 인코딩을 적용한다. PC 2는 PC 1에서 검출한 데이터를 수신하는 역할을 한다.

이때, PC 1과 PC 2는 차량용 이더넷과 동일한 최대 전송 속도 100Mbps로 전송 속도를 제한하며, 1-Pair 이더넷으로 통신한다.

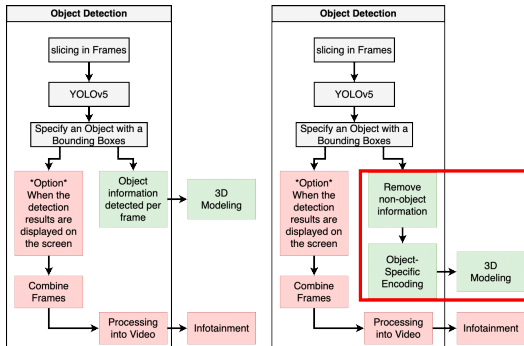


그림 2. 좌) 기존 주행 판단 시스템 우) 제안 주행 판단 시스템
Fig. 2. Left) Traditional Driving Determination System Right) Proposed Driving Determination System

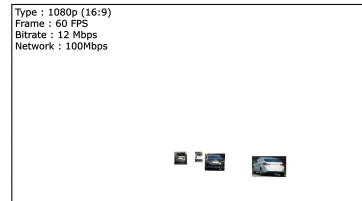


그림 4. 검출 이미지와 데이터 삭제 처리 이미지
Fig. 4. Detection image and data delete processing image

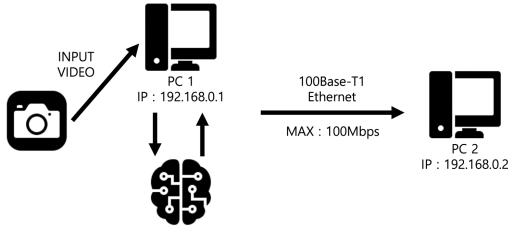


그림 5. 실험 설계
Fig. 5. Experimental Design

그림 4는 원본 영상, 검출 영상과 검출된 객체 외 영역의 데이터를 삭제한 영상을 비교한 이미지다. 이미지는 1080p 해상도, 60 FPS, 12 Mbps의 비트레이트 영상을 100Mbps의 영상 전송 속도로 전송하는 과정의 일부를 추출하였다.

단, 영상은 객체의 이미지 정보만을 포함하고 있으므로, 객체의 Class와 영상 내 객체의 위치 정보를 추가적으로 전송한다. 실시간 데이터 처리를 위해 이 정보들은 영상 데이터와 동시 처리가 필요하다. 따라서, 기존 MPEG-TS 패킷의 길이를 변경할 수 있는 페이로드 길이를 줄여, 줄어든 페이로드 길이만큼 객체의 Class와 영상 내 객체의 위치 정보를 추가한다.

본 연구에서는 그림 6과 같이 MPEG-TS 패킷을 수정하였다. MPEG-TS 패킷의 고정 길이를 가지는 TS 패킷 헤더와 적응 필드는 수정하지 않으며, 페이로드 중 29바이트를 객체의 Class와 위치 정보를 담도록 구성하였으며, 객체 정보와 영상 데이터를 나누기 위해 객체 정보 헤더를 추가하였다. 이때, 전송되는 영상 데이터에서 검출된 객체의 수에 따라 객체 정보를 지니는 Object Payload의 길이와 영상 데이터를 지니는 Video Payload의 길이는 가변적이다.

최종적으로 패킷의 데이터는 {검출된 객체의 Class, 좌표 x1, 좌표 y1, 좌표 x2, 좌표 y2}로 구성된 객체 정보와 {영상 데이터}로 구성된다. 좌표 1은 해당 객체

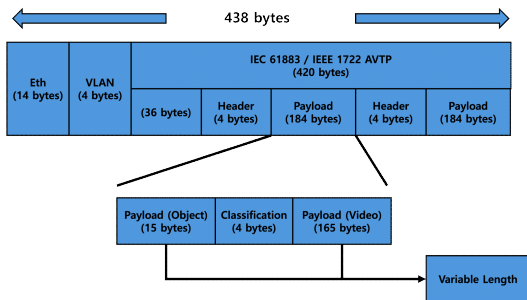


그림 6. MPEG-TS 패킷 설계
Fig. 6. MPEG-TS Packet Design

의 x 좌표와 y 좌표의 가장 낮은 값을 의미하며, 좌표 2는 x 좌표와 y 좌표의 가장 높은 값을 의미한다. 4개의 좌표 데이터를 통해 객체의 영역과 영상 내 위치 정보를 전송한다.

최종적으로 패킷의 데이터 페이로드는 {검출된 객체의 정보, 영상 데이터} 형태로 전송되며, 검출된 객체에 따라 페이로드 내 객체 정보와 영상 데이터의 크기는 가변적으로 변한다.

4.2 실험 결과

표 1은 원본 영상 데이터와 탐지 영상 데이터, 영역 삭제 데이터 순으로 해상도별 데이터 크기를 정리한 표이다.

원본 영상은 YOLOv5 모델을 통해 객체 탐지를 수행한 후, 영상 데이터의 크기가 커진다. 검출된 객체를 영상에 표시하지 않을 경우, 본 실험만큼 많이 차이 나지 않으나, 검출된 객체의 정보를 따로 저장하므로 전송할 데이터가 커지는 것은 동일하다.

하지만, 검출된 객체 외 영역의 데이터를 삭제하여 기존 영상의 크기보다 작아졌다. 1080p에서는 37.8%, 4K에서는 62.5% 데이터 크기가 줄어들었다.

그림 7. 은 1080p와 4K의 영상 원본, 물체를 검출하여 표시한 영상과 제안하는 시스템을 통해 전송한 영상의 전송 시간을 측정한 결과이다. 물체를 검출하여 표시한 영상에서는 원본 대비 3.2 ~ 4.5배 이상 전송 시간이

표 1. 데이터 크기 비교
Table 1. Data Size Comparison.

	1080p	4K
Original	206MB	870MB
Detection	1.22GB	3.65GB
Delete Area	128MB	326MB

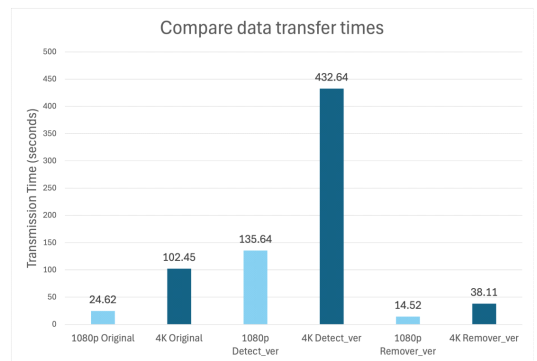


그림 7. 데이터 전송 시간 비교
Fig. 7. Comparison of data transfer times

늘었으나, 제안하는 시스템을 사용한 경우 1080p 영상 전송 시간은 41.02%, 4K 영상 전송 시간은 62.8% 감소하였다.

V. 결 론

본 논문에서는 자율주행차량의 탐지된 객체 외 영역을 제거하는 방법과 객체 인코딩을 적용한 데이터 전송 기법을 통해 자율주행 판단에 필요한 전송 영상의 데이터 크기를 줄이고, 데이터 전송 시간을 단축시켰다. 기존 시스템과 달리 카메라에서 수집된 데이터를 중앙처리장치로 전송하여 객체 탐지를 진행하는 것이 아닌, 카메라와 연동된 객체 탐지 장치를 장착하여 객체 정보와 데이터를 삭제함으로써 전송될 데이터의 크기를 줄였다. 이후, 100Base-T1을 통해 전송될 객체 정보에 객체 인코딩을 적용함으로써, 패킷 데이터의 크기를 줄여 데이터 처리 및 전송 시간 단축이 가능하다. 최종적으로 100Base-T1과 동일한 환경을 구축하여, YOLOv5를 통해 탐지된 객체 데이터에 제안 방법을 적용하였을 때, FHD 환경에서는 37.8%가량 데이터 크기가 줄었으며, 전송 시간은 41.02% 감소하였다. 4K 환경에서는 62.5% 데이터 크기가 줄었으며, 62.8% 전송 시간이 줄어들었다. 본 연구를 통해 기존 자율주행차량의 검출된 객체의 데이터 전송 시간을 개선하였다. 제안 방법을 데이터 처리량이 많은 고해상도 영상에 적용하여 3D 모델링에서 활용할 수 있음을 의미한다.

VI. 향후 연구 계획

본 연구에서는 자율주행 차량의 3D 모델링 프로세스의 데이터로 영상 데이터 내 객체 탐지와 분류가 완료된 데이터를 사용하였다.

수신기에서 원본 고해상도 영상 수신에 불가하다는 문제점이 존재하며, 한정된 전송속도에서 제안 기법이 최적인지 고려되지 않았다.

따라서, 원본 고해상도 영상의 송수신과 제한된 전송속도에서 최적의 기법에 대해 추가 연구를 진행할 예정이다.

References

[1] S. Kim, Y. Kim, H. Jeon, D. Kum, and K. Lee, "Autonomous driving technology trend and future outlook: Powered by artificial intelligence," *Trans. Korean Soc. Automotive*

Eng., vol. 30, no. 10, pp. 819-830, Aug. 2022. (<https://doi.org/10.7467/KSAE.2022.30.10.819>)

- [2] W. Zeng, M. A. S. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surv. & Tuts.*, vol. 18, no. 3, pp. 1552-1571, Jan. 2016. (<https://doi.org/10.1109/COMST.2016.2521642>)
- [3] X. Liu, Z. Nie, D. Li, and H. Yu, "Design of an improved ethernet AVB model for real-time communication in in-vehicle network," *2019 IEEE 3rd Inf. Technol., Netw., Electr. and Automat. Control Conf. (ITNEC)*, pp. 6-10, Jun. 2019. (<https://doi.org/10.1109/ITNEC.2019.8729189>)
- [4] C.-K. Son, S.-C. Oh, E.-R. Kim, S.-R. Kim, and Y.-G. Kim, "A study on automotive diagnostic system using CAN, CAN FD, FlexRay," *J. Inst. Internet, Broadcasting and Commun.*, vol. 16, no. 6, pp. 311-318, Dec. 2016. (<https://doi.org/10.7236/JIIBC.2016.16.6.311>)
- [5] Y. Wu and J. G. Chung, "Efficient CAN data compression algorithm using signal length," *Smart Media J.*, vol. 3, no. 3, pp. 9-14, Sep. 2014.
- [6] PORTER, Donovan, *100BASE-T1 Ethernet: the evolution of automotive networking*(2018), Retrieved May 13, 2024, from <https://www.ti.com/lit/wp/szy009/szy009.pdf>
- [7] IEEE, 802.3u-1995, IEEE standard, 1995. (<https://doi.org/10.1109/IEEESTD.1995.7974916>)
- [8] Redis, *OBJECT ENCODING*(2020), (Retrieved May 13, 2024, from <https://redis.io/docs/latest/commands/object-encoding/>)
- [9] Apple, *Cocoa Core Competencies*(2018), Retrieved May 13, 2024, from <https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/ObjectEncoding.html>
- [10] S.-J. Jang, "자율 주행 자동차 관련 SW기술 동향," *Inf. and Commun. Mag.*, vol. 33, no. 4, pp. 27-33, 2016. (<https://www.dbpia.co.kr/Journal/articleDetail?n>

odeId=NODE06646126)

- [11] J.-S. Oh, K.-I. Lim, and J.-H. Kim, "A research of obstacle detection and path planning for lane change of autonomous vehicle in urban environment," *J. Inst. Control, Robotics and Syst.*, vol. 21, no. 2, pp. 115-120, 2015. (<https://doi.org/10.5302/J.ICROS.2015.14.9006>)
- [12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, Apr. 2018. (<https://doi.org/10.48550/arXiv.1804.02767>)
- [13] C. Chen, Z. Zheng, T. Xu, S. Guo, S. Feng, W. Yao, and Y. Lan, "Yolo-based uav technology: A review of the research and its applications," *Drones*, vol. 7, no. 3, p. 190, Mar. 2023. (<https://doi.org/10.3390/drones7030190>)
- [14] Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, and G. Kuang, "BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images," *Remote Sensing*, vol. 13, no. 21, pp. 4209, Oct. 2021. (<https://doi.org/10.3390/rs13214209>)
- [15] H.-D. Lee, S.-G. Kim, S.-C. Na, J.-Y. Ham, and C. Kwak, "A study on traffic vulnerable detection using object detection-based ensemble and YOLOv5," *J. Korea Soc. Comput. and Inf.*, vol. 29, no. 1, pp. 61-68, 2024. (<https://doi.org/10.9708/jksci.2024.29.01.061>)

김 시 온 (Si-On Kim)



2023년 2월 : 순천향대학교 정보보호학과 학사과정
 2023년 3월~현재 : 순천향대 학교 정보보호학과 석사과정
 <관심분야> 정보보안, 암호응용, 네트워크보안, 딥러닝
 [ORCID:0009-0007-0312-8084]

이 선 영 (Sun-Young Lee)



1993년 2월 : 부경대학교 전자계산학과 (이학사)
 1995년 2월 : 부경대학교 전자정보공학과 (이학석사)
 2001년 3월 : 일본도쿄대학 전자정보공학과 (공학박사)
 2004년 3월~현재 : 순천향대학교 정보보호학과 교수
 <관심분야> 콘텐츠보안, 암호이론, 정보이론, 정보보안
 [ORCID:0000-0002-4686-9436]