

# 클라우드 서버 환경에서 Lidar 센서와 Yolo v3 라이브러리를 이용한 360도 영상 인식 시스템 구현

조국한\*, 조현우\*\*, 송영준\*\*\*

## Implementation of 360-degree Image Recognition System Using Lidar Sensors and Yolo v3 Libraries in Cloud Server Environment

Guk-Han Jo\*, Hyun-Woo Cho\*\*, and Young Joon Song\*\*\*

---

이 연구는 금오공과대학교 학술연구비로 지원되었음(2018104128)

---

### 요 약

본 논문에서는 Lidar 센서와 360도 카메라를 활용하여 클라우드 환경에서 동작하는 영상인식 시스템을 새롭게 구현하였다. 360도 카메라를 사용한 블랙박스과 Lidar에서 수신된 데이터들을 분류 및 분석한다. 시스템 구성의 복잡도 감소와 클라우드 서버에서의 효율적인 데이터 관리와 데이터 신뢰성 향상을 위하여, Lidar에서 움직임이 감지될 경우에만 블랙박스에서는 서버로 영상데이터를 전송하도록 시스템을 구성하였다. 이렇게 전송된 영상데이터들은 10초단위로 여러 개의 프레임으로 나뉘어 저장되며, 저장된 영상 데이터들을 서버 내부에서 You Only Look Once v3를 적용하여 분석하였다. 분석이 완료된 영상 및 이미지 데이터들은 Bounding Box로 인식된 물체가 표시된다.

### Abstract

In this paper, we implement the new image recognition system being operated in the cloud environment using Lidar sensors and 360 degree cameras. We analyze and classify data received from dashcam with 360 degree camera and a Lidar sensor. To reduce the system complexity and to efficiently manage the data in the cloud server as well as to improve the reliability of data, the video data received from the dashcam with 360 degree camera is transmitted to the cloud server only if the Lidar sensor detects some movement at the same time. These transmitted image data are stored through multiple frames in 10 seconds, and then the saved image data are analyzed using You Only Look Once v3 algorithm in the cloud server. The images and image data that have been analyzed are displayed as the objects in the bounding boxes.

### Keywords

cloud servers, lidar, Yolo v3, image recognition system, dashcam

---

\* 금오공과대학교 전자공학과 박사과정  
- ORCID: <https://orcid.org/0000-0002-4384-2852>  
\*\* 금오공과대학교 전자공학과 석사과정  
- ORCID: <https://orcid.org/0000-0003-4504-0965>  
\*\*\* 금오공과대학교 전자공학부 교수(교신저자)  
- ORCID: <https://orcid.org/0000-0002-9736-602X>

· Received: Jan. 09, 2020, Revised: Feb. 04, 2020, Accepted: Feb. 07, 2020  
· Corresponding Author: Young Joon Song  
Dept. of EE, Kumoh National Inst. of Tech., Korea  
Tel.: +82-54-478-7486, Email: [yjsong@kumoh.ac.kr](mailto:yjsong@kumoh.ac.kr)

## 1. 서론

최근 클라우드 서버(Cloud server)를 활용한 공유 기술과 딥러닝은 특정분야에서만 사용되는 기술 아닌 여러 가지 방면에서 중요한 기술로 활용되고 있다. 이러한 발전은 여러 가지 분야가 독립적으로 각자의 역할을 수행하기 위하여 동작하는 것이 아니라 클라우드 서버라는 가상 서버를 이용하는 것을 뜻한다. 클라우드 환경은 각각의 장치마다 서로 데이터를 공유하고 협업하여, 하나의 새로운 서비스 및 기능들을 구현하는데 중요한 역할을 하고 있다. 이에 맞물려 현재는 차량 내부에 장착되는 다양한 제품들도 점점 출시되고 있다. 본 연구에서는 이러한 차량 내부 장치들 중에서도 블랙박스를 이용하였으며, 블랙박스 중에서도 360도 촬영이 가능한 블랙박스를 이용하여 연구를 진행하였다. 또한 현재 자율주행과 모션인식 기술의 핵심인 Lidar 센서를 블랙박스와 접목하여, 차량 내부 보안 및 인식 성능 향상에 필요한 기술들을 연구하였다.

본 연구에서 사용한 서버는 AWS(Amazon Web Services) 클라우드 서버 EC2(Elastic Compute cloud)이다. AWS 서버 EC2는 자체적으로 인스턴스를 구성하고, 구성된 인스턴스 내부에서 개발된 프로그램 및 소스 코드들이 동작하는 구조로 되어있다[1]. 이렇게 구성된 서버로 카메라에서 수집된 이미지나 영상을 전송한다. 이때 수집된 영상 및 Lidar 데이터들은 TCP 소켓(Socket)을 사용하여 전송한다. 서버는 360도 카메라를 통해 수집된 영상 및 이미지를 Yolo(You only look once) v3 라이브러리를 사용하여 분석한다[2]. 이와 같이 본 연구에서는 기존에 활용성이 높은 Lidar 및 블랙박스와 AWS 클라우드 서버 EC2 간의 협업을 통한 새로운 시스템 구성을 제안한다.

본 논문의 구성은 다음과 같다. 2장은 Yolo v3 알고리즘 및 라이브러리와 세팅에 관하여 기술한다. 3장은 Lidar 센서의 구성과 활용방법에 대하여 설명하고 4장에서는 전체적인 시스템을 구성하는 구성도와 그에 대한 필요성 및 중요성에 대하여 설명한다. 5장에서는 Yolo v3 라이브러리 및 Lidar 센서를 활용하여 구현된 결과들을 설명하고, 마지막으로 6

장에서는 향후 연구 계획 및 개선 방향에 대하여 기술한다.

## II. 영상인식 알고리즘과 클라우드 서버 세팅

### 2.1 Yolo v3 라이브러리

영상인식을 진행할 때 기존에 연구들은 CNN을 이용하여 객체를 인식하고 결과를 도출한다. 본 연구에서 제안한 새로운 시스템을 테스트하기 위해 Yolo v3 라이브러리를 이용하여 객체 검출을 수행한다. Yolo v3는 사용이 용이하고 이미지 인식 성능이 뛰어나 비교적 빠른 처리시간이 필요한 시스템에 최적화되어 있다. 이는 1-state detection으로써 처리과정이 타 CNN(Convolutional Neural Network) 알고리즘에 비해 간단하고 그리드(Grid) 방식으로 클래스 분류가 이루어진다. 이와 같은 장점 때문에 처리 속도도 빠르며, 높은 mAP를 보인다. 하지만 기존 타 CNN 알고리즘에 비해 정확도 및 성능 면에서 떨어진다는 단점이 존재한다. 기존 R-CNN이 느린 이유는 Proposal 수가 많으며 그로인한 오버헤드가 많기 때문에 영상이나 이미지를 처리함에 있어 상당한 시간이 소요된다. Yolo v3는 이러한 시간적 소모를 줄이기 위해 제안되었다[2][3].

### 2.2 AWS 클라우드 서버 EC2 세팅

본 연구에서는 AWS 클라우드 서버 EC2를 이용하여 서버를 구축하였으며, EC2는 웹에서 서버를 생성하는 것이 가능하다. 또한 S3를 사용하여 부가적으로 EC2 인스턴스를 보완할 수 있다. S3의 기능은 EC2 서버에 원하는 데이터를 홈페이지를 통해 업로드 할 수 있으며, Mount를 통해 서로의 파일 및 데이터를 공유 할 수 있다. 그러나 AWS 서버를 성공적으로 구축하고 활용하기 위해서는 아래의 4가지 항목을 우선적으로 고려해야한다.

- AWS 클라우드 서버 EC2의 인스턴스 유형
- EC2의 볼륨 크기
- EC2 내부에서 동작하는 OS
- 네트워크 및 보안키 설정

첫 번째 AWS 클라우드 서버 EC2의 인스턴스 유형을 정해주는 과정이다. EC2는 인스턴스를 생성할 때 다양한 유형의 인스턴스가 존재한다. 이러한 인스턴스들은 각각의 특화된 기능 및 성능이 존재하며, 기본적으로 T3의 유형을 무료로 제공한다. 이외에 컴퓨팅 최적화, 범용, 스토리지 최적화, 메모리 최적화, 컴퓨팅 가속화 등의 여러 가지 인스턴스 유형들이 존재한다. 본 연구에서는 컴퓨팅을 최적화하기 위하여 C4 인스턴스 유형을 지정하였다. 다른 성능이 필요하다면 인스턴스 유형별 기능 확인하고 서버 인스턴스를 구성할 수 있다.

두 번째는 EC2의 볼륨 크기를 설정하는 과정이다. EC2에서 설정된 볼륨의 크기는 서버의 용량과 동일하며, 생성 후에 서버 관리 홈페이지를 통해 볼륨 크기를 변경하여 사용이 가능하다. 볼륨 크기를 변경한 내용이 적용되기 위해서는 서버 인스턴스를 다시 시작 하여야한다.

세 번째는 EC2 내부에서 본인이 작업할 공간의 OS를 설정해주는 과정이다. 이 과정은 인스턴스를 생성할 때 설정할 수 있으며, Window, Linux, Unix 등의 다양한 OS들을 선택하여 운영체제를 구성할 수 있다.

인스턴스를 구성하고 OS가 설치되었다면, 이제 EC2에 접속이 가능하다. 그러나 접속을 위해서는 매번 키페어를 다운받아 접속할 때마다 입력하여야만 한다. 이러한 방법은 접속의 불편함과 동시에 센서와 카메라 연동 시 개발에 어려움을 줄 수 있다. 이 문제를 해결하기 위해서는 서버 인스턴스로 항상 접속이 가능하도록 암호로 접속하는 방법으로 변경하여야 한다.

네 번째는 네트워크 및 보안키 설정이다. 보안키 설정은 서버 인스턴스를 구성하고 실제 코딩 작업 및 장치 간 연결에 중요한 역할을 담당한다. 이 설정 과정이 없으면 장치 간 통신 및 연동이 되지 않고, 또한 개발자의 작업에 상당히 어려움이 따르게 된다. 이를 해결하는 방법은 그림 1에서 설명한다.

그림 1은 AWS 클라우드 EC2의 비밀번호 접속 세팅 방법이다[3]. EC2의 접속 방법을 변경하기 위해서는 먼저 키페어를 이용하여 Putty나 다른 접속 방법으로 ssh 형식으로 접속해야한다.

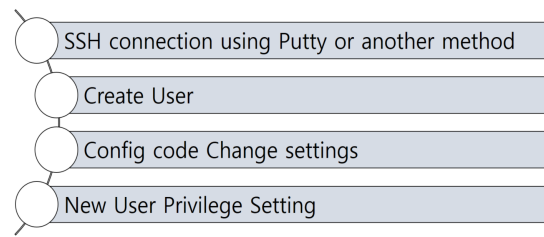


그림 1. AWS Cloud EC2 비밀번호 접속 세팅  
Fig. 1. AWS cloud EC2 internal setting

이후 서버 내부에서 작업 할 때 사용할 유저를 생성하고 접속 시 입력하는 비밀번호를 세팅 한다. 유저생성이 완료되면 서버 시스템 내부에 존재하는 config code내 암호 접속 허용 부분을 no에서 yes로 변경한 후에 /PasswordAuth를 추가한다. 이후 재접속하게 되면 비밀번호로 접속이 가능하게 된다[4].

### III. Lidar 센서

본 논문에서 Lidar 센서는 YDLidar F4Pro를 사용하였으며[5], 시리얼통신 이용하여 움직임을 인식하는 코드를 구현하였다. Lidar의 시작 및 종료 명령어는 16진수로 이 명령어를 코드 작성 시 시작부분과 종료부분에 각각 넣어 Lidar로 전송하여야만 Lidar가 동작한다. 본 논문에서는 Matlab을 이용하여 구현 하였으며, fwrite를 이용하여 명령어를 전송할 수 있다. 하지만 Matlab에서는 16진수는 10진수로 변환하여 명령어를 전송하여야 한다.

그림 2는 Lidar 센서가 데이터를 받아서 전송해주는 패킷의 형태를 나타낸 것이다. 이러한 패킷 구조를 인지하고 있어야만 패킷의 데이터들을 가공 및 변환 할 수 있다. 그림 2의 패킷은 YDLidar F4 Pro의 전송 패킷이다. 먼저 PH는 패킷의 헤더를 나타낸다. CT는 패킷의 타입이며 이 부분은 Lidar 센서 데이터를 사이클 단위로 수집하는데 중요한 역할을 한다.

PH	CT	LSN	FSA	LSA	CS	S1
----	----	-----	-----	-----	----	----

그림 2. Lidar 센서의 패킷 구성  
Fig. 2. Packet configuration of Lidar sensors

그 이유는 실제 데이터가 측정되면, Lidar가 측정을 시작한 위치에 대한 기준이 필요하다. CT는 Lidar가 한 바퀴의 측정이 끝나게 되면 Zero 패킷으로 나타난다. 이를 활용하면 Zero 패킷이 수신되었을 경우 Lidar가 한 바퀴의 측정이 완료되었다는 신호라 볼 수 있다. LSN은 샘플의 양 즉 데이터의 양을 나타내는 부분이며, FSA와 LSA는 스캔시작 각도와 END 각도를 나타낸다. 실제 이 값들은 16진수 값으로 되어 있다. 이 두 개의 범위가 Lidar의 렌즈가 한번 측정할 때 측정되는 각의 범위를 측정하는 값이며 수식을 이용하여 각도의 형태로 변환한다.

그림 3은 Lidar 센서를 제어하는 코드 및 데이터 수신 및 가공과 움직임 인식까지 구현한 코드의 구성을 나타내었다. Main에서는 Lidar를 직접적으로 동작시키는 명령어들을 fwrite를 통해 입력한다. 또한 데이터를 가공 및 변환하는 함수와 오류를 정정하는 함수를 호출하여 실행한다.

Data processing and transformation은 데이터를 가공하고 변환하는 함수를 포함하고, 그림 3과 같은 패킷을 각각 분류한다. 분류된 데이터들을 거리 및 각도로 변환하는 수식을 적용하여 수집된 데이터들을 나타낸다. 또한 0~1초대의 사물들의 위치 및 거리가 저장된 테이블을 만들고 4-5초 테이블과 비교하여 움직임이 있으면 움직임이 있다고 판단한다. Error correction은 실제 측정하면서 값이 일정하게 측정되지 않거나 움직임이 없음에도 값이 다르게 수신되는 오류들을 수정하기 위해 구현한 함수이다. 이 함수는 내부에 테이블 간 비교를 통하여 움직임을 판단하는 역할이 포함되어있다.

이러한 코드가 동작한 이후 Lidar 센서가 움직임을 감지하였을 경우 블랙박스에서는 서버로 영상데이터를 전송 하게 된다. 이후 영상데이터는 서버에서 Yolo v3를 이용하여 영상인식을 수행한다. 블랙박스에서 서버로 영상 데이터를 보낼 때는 소켓통신을 이용하여 데이터를 전송한다. 소켓통신 규격은 포트 7000번을 통하여 전송된다. 이때 EC2 클라우드 서버 내부 설정에서 인바운드 규칙에 포트 7000번을 추가하여야만 소켓통신이 정상적으로 동작한다.

#### IV. 장치와 클라우드 서버 간 시스템 구성도

서버에서 펌웨어를 실행하면 Lidar와 전 방향 카메라가 동시에 동작되도록 구현하였다. 본 연구에서는 물체의 거리 및 범위만을 인식하여 데이터를 전송하는 이차원 Lidar를 이용하여 연구를 진행하였다. Lidar를 사용하면, 카메라로 사물을 인식하기 힘든 야간 상황에서도 효과적으로 사용되어 질 수 있게 된다. Lidar에서 우선적으로 움직임이 감지되면, 블랙박스에서 영상데이터를 서버로 보낸다. 이는 블랙박스에서 지속적으로 영상을 서버에 업로드 할 필요가 없고, Lidar에서 우선적으로 움직임이 인식된다면 서버로 영상데이터를 전송하게 되는 장점이 있다. 이러한 시스템 구성으로 대용량 영상 데이터를 서버에 지속적으로 전송하는 현상을 방지할 수 있게 되고, 데이터 전송 시간 단축 및 수집 데이터의 신뢰도가 향상되는 효과를 기대할 수 있다. 또한 클라우드 서버의 저장 장치를 효과적으로 관리할 수 있게 된다.

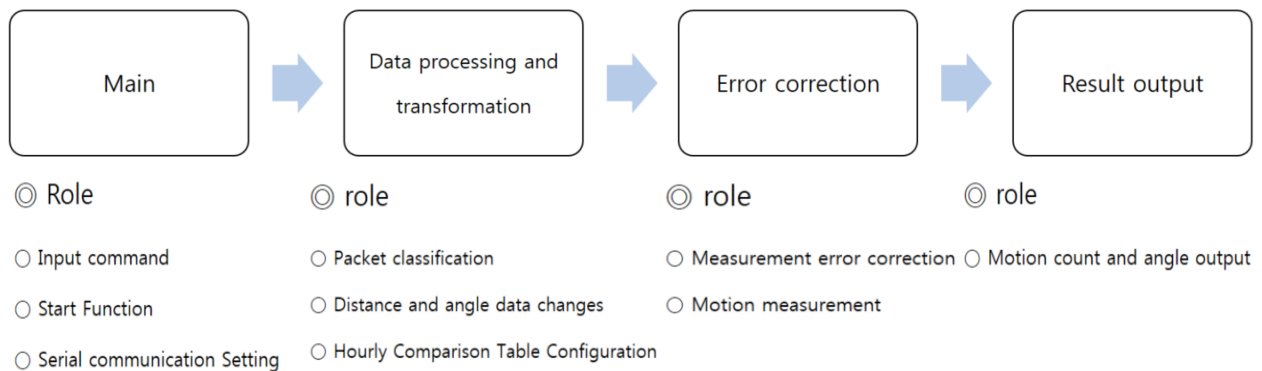


그림 3. Lidar 동작 및 움직임 감지 코드 내부 함수의 역할 및 구성  
Fig. 3. Lidar source code configuration

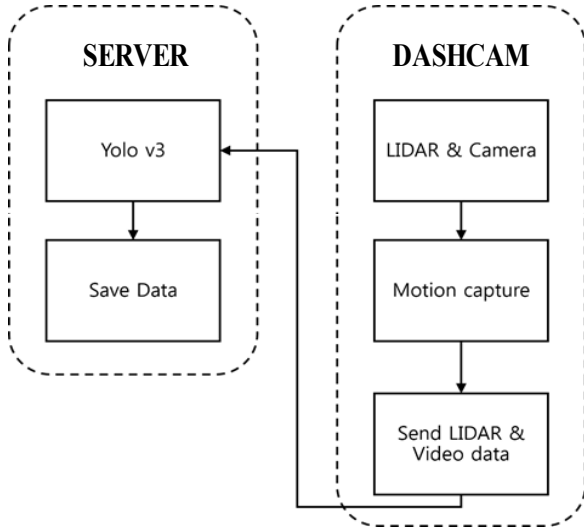


그림 4. 클라우드 서버환경에서의 영상인식을 위한 전체적인 시스템 구성도

Fig. 4. Diagram of image recognition system using AWS cloud server



그림 5 Yolo v3를 이용한 영상 인식 결과(인식 성공)

Fig. 5. Image recognition result using Yolo v3 (recognition success)

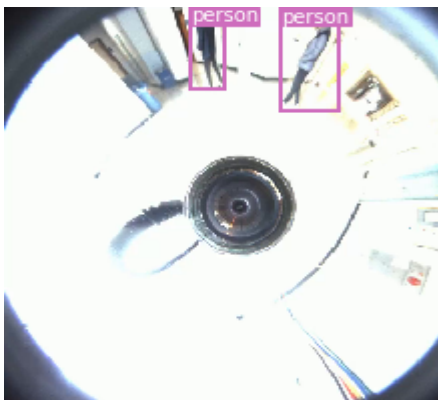


그림 6 Yolo v3를 이용한 영상 인식 결과(인식실패)

Fig. 6. Image recognition result using Yolo v3 (recognition failure)

그림 4는 본 연구에서 제안하는 새로운 시스템 구성도이다. Lidar에서 움직임이 감지되면, 360도 카메라의 영상데이터와 Lidar 센서의 데이터를 서버로 연동규격에 따라 7000번 TCP 포트에 전송한다. 이러한 연동 규격은 개발 구성에 따라 달라질 수 있다. 이렇게 전송된 데이터들은 EC2 서버 내부에 작성된 Yolo v3 라이브러리 코드와 360도 이미지 데이터를 평면 이미지 데이터로 변환하는 알고리즘을 이용하여 10초의 360의 영상을 13개의 평면 프레임으로 변환하여 물체를 인식하게 된다. 인식된 물체가 사람인 경우 영상 속 인식되는 사람의 수를 저장하도록 하였다.

차량의 소유자이자 제안하는 시스템을 이용하는 사용자에게 차량 주위에 사람이 접근하였다는 경고 알림을 주어 차량 도난, 차량 안 귀중품 도난, 차량 파손 등에 대비할 수 있도록 경고 알림을 보내는 부분도 추가할 것이다.

## V. Lidar 센서와 Yolo v3를 이용한 영상인식 결과

클라우드 환경에서 블랙박스 및 Lidar를 이용하여 시스템을 구성한다. 이를 통하여 블랙박스 및 Lidar에서 움직임이 발생한 영상 즉 필요한 360도 영상만을 송신하여 서버 내에서 영상인식이 올바르게 동작하는지를 알아본다. 그림 5는 Lidar를 통하여 움직임이 인식 되었을 경우 서버에서 수신된 데이터를 Yolo v3[6][7]를 이용하여 영상인식을 수행한 결과이다. 사람만 인식되도록 Yolo v3를 학습시켜 진행하였다[8]. 사람 2명의 영상에 들어오자 사람이 Bounding Box를 통해 2명 모두 인식된 것을 볼 수 있다. Yolo v3를 통해 인식된 영상 데이터는 10초 단위로 13개의 프레임에서 인식된 물체에 Bounding Box가 표시된 이미지 데이터로 출력된다. 그림 5은 영상인식이 적용된 10초 단위 13개 이미지 데이터 중의 하나이다.

그러나 이러한 360도 영상은 실제 영상 인식을 진행할 때 많은 문제점이 발생한다. 그림 6을 살펴 보게 되면 인식된 한명 뒤에 또 다른 사람이 나타났음에도 인식이 되지 않고 왼쪽에 촬영된 사람만 인식되어 Bounding Box가 찍워진 결과를 볼 수 있

다. 이는 렌즈 특성상 왜곡이 심하게 일어나 인식 오류가 발생하는 것을 알 수 있다. 따라서 360도 영상 및 이미지는 평면에서 촬영된 영상에 비해 인식률이 저하되는 현상이 나타난다. 또한 영상의 광도도 인식의 정확도를 결정하는데 상당히 중요하기 때문에 빛의 세기를 조절해야 한다. 하지만 사용한 360도 카메라에서는 광도 조절이 어려웠다.

360도 카메라의 왜곡 현상을 해결하기 위하여 360도 영상을 평면 영상으로 펴는 작업 후 영상인식을 수행하여 인식률을 향상시켰다. 그림 7과 8은 360도 이미지 데이터를 평면 이미지 데이터로 변환하는 Omnidirectional Camera Calibration[9][10]을 적용하여 360도 영상을 평면으로 변환 후에 Yolo v3를 이용하여 영상인식을 수행한 결과이다. 이 수행 결과로 360도 영상에서의 물체에 대한 왜곡 발생이 감소하여 기존에 40%이하로 떨어지는 인식률을 100번 (1회당 10초 단위)의 인식률 테스트를 진행한 결과 밝기에 따라 50~70%로 인식률이 개선되는 것을 확인하였다. 그러나 360도 이미지 데이터를 평면 이미지 데이터로 변환하는 과정이 추가되어 실행시간이 360도 이미지 데이터를 이용하였을 때보다 증가하게 되었다. 또한 360도 이미지 데이터를 평면 이미지 데이터로 변환하여도 완벽하게 왜곡을 해결할 수 없었다. 따라서 360도 영상의 왜곡을 줄이는 방법에 대한 추가적인 연구가 필요하다고 생각된다.

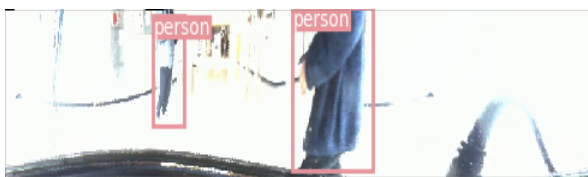


그림 7 평면 알고리즘을 적용한 영상 인식 결과 1  
Fig. 7. Image recognition results using omnidirectional camera calibration algorithm 1

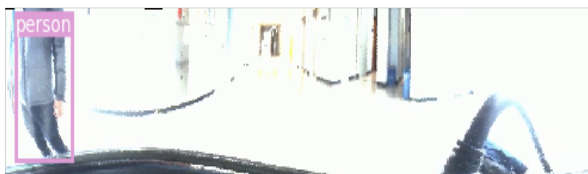


그림 8 평면 알고리즘을 적용한 영상 인식 결과 2  
Fig. 8. Image recognition results using omnidirectional camera calibration algorithm 2

## VI. 결론 및 향후 과제

폭발적인 데이터양의 증가로 새로운 예측데이터 및 인식데이터를 추출하는 작업은 일반적인 개인용 하드웨어에서 처리하기 힘들게 되었다. 이러한 빅데이터를 처리 및 공유하기 위하여 현재는 클라우드 및 관련 자원을 빌려와 사용하는 방식이 일반화되고 있다.

본 논문에서는 센서 데이터 및 블랙박스에서 촬영되는 많은 양의 데이터를 서버를 통해 분석하는 새로운 시스템의 구성을 제안하였다. 서버에서는 이러한 분석의 결과를 통해 특정 물체를 인식하고 이를 사용자에게 알려주는 시스템을 구현하였으며, Yolo v3 알고리즘을 이용하여 서버 내에서 영상인식이 가능한 스마트 블랙박스 시스템을 구현하였다. 또한 본 시스템을 구성하여 필요할 때만 데이터를 전송하였기 때문에 클라우드 서버가 전송받는 데이터는 블랙박스의 모든 영상데이터를 수신받을 필요 없이 움직임이 발생한 부분만 받을 수 있다. 이를 통하여 서버의 과부하 및 부담을 덜어 줄 수 있다. 이렇게 수신된 영상들은 Yolo v3로 분석을 진행하였고, 360도 왜곡된 영상을 Omnidirectional Camera Calibration을 적용하여 인식률을 향상시켰다.

이는 실제 차량 뿐 만 아니라 공공기관 및 개인 주택 등과 같은 보안 시스템 구성에도 기여할 수 있을 것으로 예상된다.

향후에는 영상의 밝기 등 영상의 질을 향상시켜 더 나은 인식률을 얻을 수 있도록 연구를 진행할 예정이다. 그리고 360도 카메라의 왜곡 현상을 해결할 방법에 대한 연구도 진행할 예정이다. 또한 Lidar 센서와 블랙박스를 포함한 다른 여러 영상 촬영 장치 및 센서들을 이용한 시스템을 구성하여 차량을 위한 보안 시스템이 아닌 여러 상황에서 사용가능한 보안 시스템으로 발전시켜 나갈 예정이다.

## References

- [1] AWS Cloud Server EC2 Amazon EC2 Instance Types, <https://aws.amazon.com/ec2/instance-types/>. [accessed : Nov. 10, 2019]
- [2] Seungbo Shim and Sang-Il Choi, "Development on

Identification Algorithm of Risk Situation around Construction Vehicle using YOLO-v3", Journal of the Korea Academia-Industrial cooperation Society, Vol. 20, No. 7, pp. 622-629, Jul. 2019.

- [3] Tian, Yunong, Yang, Guodong, Wang, Zhe, Wang, Hao, Li, En, and Liang, Zize, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model", Computers and electronics in agriculture, Vol. 157, pp. 417-426, Feb. 2019.
- [4] AWS Cloud Server EC2 Password access via ssh, <https://aws.amazon.com/ko/premiumsupport/knowledge-center/ec2-password-login/>. [accessed : Nov. 20, 2019]
- [5] YDLidar Sensor, <http://www.ydlidar.com/download/>. [accessed : Nov. 1, 2019]
- [6] Seungbo Shim and Sangil Choi, "Development on Identification Algorithm of Risk Situation around Construction Vehicle using YOLO-v3", Journal of the Korea Academy Industrial Cooperation Society, Vol. 20, No. 7, pp. 622-629, Jul. 2019
- [7] Gukhan Jo, Hyun Woo Cho, Se-min Oh, Jong-Chun Shin, Dong-min Bea, and Young Joon Song, "Image Analysis and Object Recognition Technology using YOLO V3 Algorithm and LIDAR Sensor", Proceedings of Korea Institute Of Communication Sciences, pp. 30-31, Jan. 2019
- [8] Ivašić-Kos, M., Krišto, M., and Pobar, M, "Human detection in thermal imaging using YOLO", Proceedings of the 2019 5th International Conference on Computer and Technology Applications, Istanbul Turkey, pp. 20-24. Apr. 2019
- [9] Joko Hariyono, Wahyono, and Kang-Hyun Jo, "Accuracy Enhancement of Omnidirectional Camera Calibration for Structure from Motion", ICCAS 2018, Institute of Control, Robotics and Systems, Gwangju, South Korea, pp. 589-592, Oct. 2013.
- [10] Xuanrui Gong, Yaowen Lv, Xiping Xu, Zhaoguo

Jiang, and Zhuang Sun, "High-Precision Calibration of Omnidirectional Camera Using an Iterative Method", IEEE Access, Vol. 7, pp. 152179-152186, Oct. 2019.

### 저자소개

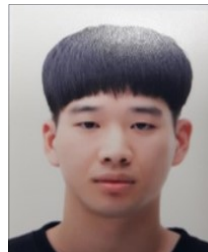
조 국 한 (Guk-Han Jo)



2016년 2월 : 금오공과대학교  
컴퓨터공학과 공학사  
2019년 2월 : 금오공과대학교  
전자공학과 공학석사  
2019년 3월 ~ 현재 : 금오공과  
대학교 전자공학과 공학박사  
과정 재학

관심분야 : 머신러닝, 채널코딩

조 현 우 (Hyun-Woo Cho)



2018년 8월 : 금오공과대학교  
컴퓨터공학과 공학사  
2018년 9월 ~ 현재 : 금오공과  
대학교 전자공학과 공학석사  
과정 재학  
관심분야 : 머신러닝, 채널코딩

송 영 준 (Young Joon Song)



1987년 2월 : 한양대학교  
전자통신공학과 공학사  
1994년 2월 : 한양대학교  
전자통신공학과 공학석사  
1999년 2월 : 한양대학교  
전자통신공학과 공학박사  
1994년 10월 ~ 2002년 2월 :

LG전자 책임연구원

2002년 3월 ~ 현재 : 금오공과대학교 전자공학부 교수,  
IEEE senior member

관심분야 : 머신러닝, 채널코딩, 이동통신 시스템