

# 컴퓨터 비전 기반 무인 버스 운행시스템

이용한, 김범영, 이신효, 이지훈  
 인하대학교 정보통신공학과  
 e-mail : [yhlee3073@naver.com](mailto:yhlee3073@naver.com)

## Computer vision based unmanned bus operating system

Yong-Han Lee, Beom-Young Kim, Sin-Hyo Lee, Ji-Hun Lee  
 Dept. of Information & Communication Engineering, Inha University

### 요 약

본 시스템은 자율 주행 버스를 위한 시스템이다. 딥러닝(Deep Learning) 기반 컴퓨터 비전 기술을 이용해 차선과 물체 인식을 하여 버스를 제어하는 방식으로 자율 주행을 가능하게 하는 시스템으로 교통비 완화 및 안정성 증대를 기대할 수 있다.

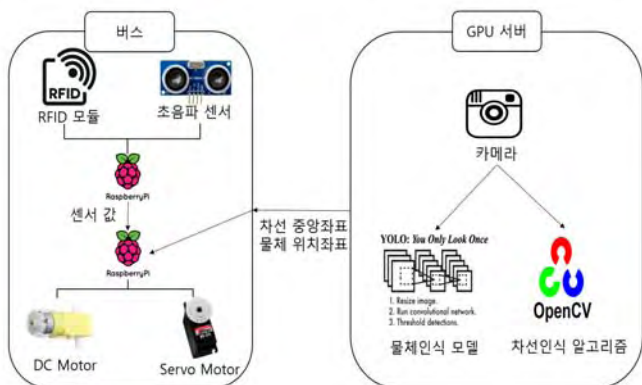
### Abstract

This system is designed for autonomous buses. It controls buses by lane and object recognition using Deep Learning based computer vision technology. Through this system, we can expect to reduce traffic costs and increase stability.

### 1. 서 론

4차 산업혁명이 화두가 된 이후로 인공지능은 꾸준히 우리의 관심을 끌고 있다. 인공지능이 접목된 기술 중 무인 자율 주행 기술은 국내외적으로 제일 큰 관심사이다. 경찰청의 자료에 따르면 연간 교통사고 발생 횟수는 총 232,035건, 그로 인한 사망자 및 부상자가 355,021명이며 대중교통 사고율은 연간 증대되고 있다. 사고 발생은 운전자 과실이 대다수이다. 그중 졸음운전으로 발생한 사건은 2433건, 음주운전으로 발생한 사건은 19,769건으로 비교적 많은 비중을 차지한다. 음주운전과 졸음운전은 비교적 늦은 시간대에 발생한다. 또, 대중교통은 승객들의 이용 시간에 영향을 받지 않고 운행될 필요가 있다. 본 논문에서는 위 조건을 만족시키며 부가적인 측면으로 교통비 완화를 가능하게 하는 “무인 버스 운행시스템”의 시스템 및 구현 방안을 제시한다.

### 2. 시스템 모델



[그림 1 자율 주행 버스 운행 시스템 구성도]

무인 버스의 전방 카메라는 실시간으로 영상을 취득한다. 그리고 이 영상을 바탕으로 GPU 서버는 OpenCV를 이용해 라인 인식을 하고, 딥러닝(Deep Learning) 기반 물체인식 모델인 YOLO (You Only Look Once)를 이용해 물체 인식을 한다. 인식으로 얻은 차선의 중앙 좌표와 물체의 이름과 좌표를 소켓 통신을 이용하여 Raspberry Pi에게 보낸다. Raspberry Pi는 차선의 중앙 좌표에 따라 DC 모터 및 Servo 모터를 제어한다.

또한 인식되는 신호등의 색에 따라 주행 또는 멈추게 되고, 위험 범위 내에 차량이나 사람이 인식되면 속도를 줄이거나 멈춘다.

무인 버스 바닥에 설치된 RFID 리더기와 정류장 앞 도로 바닥에 설치한 IC 카드를 이용해 정류장을 인식한다. 정류장이 인식되면 버스가 멈추고 일정 시간 뒤 다시 출발한다.

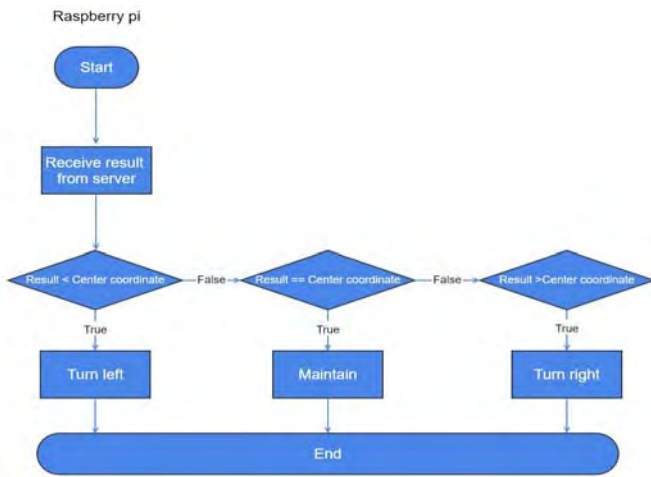
카메라의 사각지대에 있는 장애물 인식 및 안전성 확대를 위해 초음파 센서를 이용하여 근접 장애물을 인식하여 차량 제어를 한다.

### 3. 하드웨어 설계

무인 버스의 구성은 Raspberry Pi3, USB 카메라, PWM, DC 모터, DC 모터 드라이버, Servo 모터, 초음파 센서, RFID로 구성되어 있다.

사용한 PWM은 12-bit PWM Driver이다. 높은 bit 수를 가진 PWM Driver를 이용하여 좀 더 섬세한 회전 Control을 가능하게 한다. 사용한 USB 카메라는 오토포커스 지원, 자체 노이즈 Filtering으로 선명한 도로 환경과 차선을 확인하고, Wide 광각렌즈로 넓은 시야를 확보하여 코너에서 꾸준한 차선 인식을 가능하게 한다.

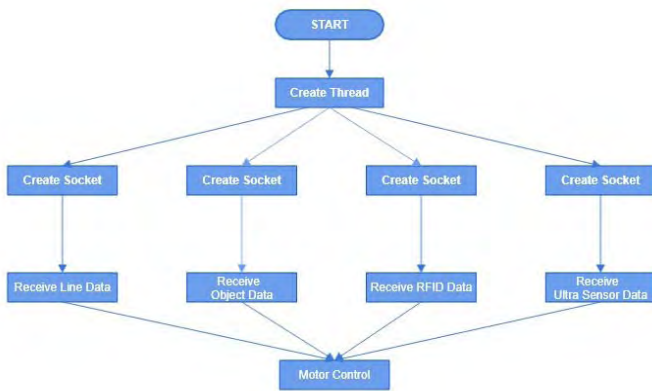
### 3.1 Actuator part 분석



[그림 2 Actuator 구동 알고리즘]

소켓 통신을 이용해 GPU 서버로부터 도로 중심 좌표 값을 받아들인다. 받아들인 좌표 값을 기준 삼아 조향 장치를 구성한다. 도로의 중심 좌표를 벗어나면 PWM의 Duty Rate를 조절하여 Servo 모터의 방향을 변경해 자동차를 도로 중심으로 이동하게 한다. 자동차의 출발, 정지는 PWM을 Enable 신호로 이용한 후, Duty Rate를 통해 속도를 조절한다.

### 3.2. 통신 분석



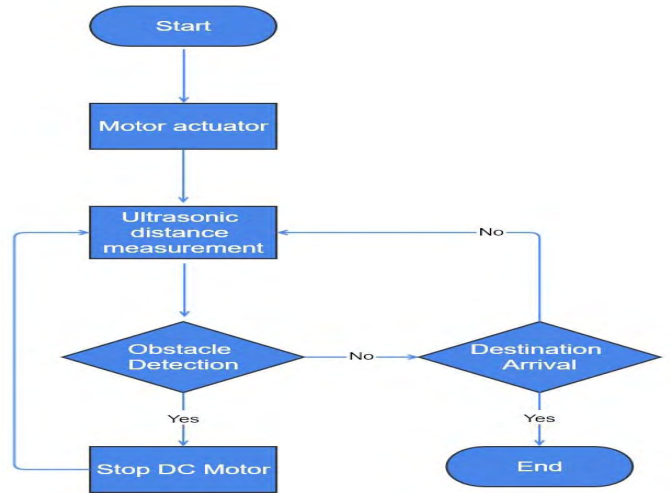
[그림 3 통신 구동 알고리즘 ]

하나의 Raspberry Pi는 라인 및 물체 인식을 실시간으로 할 수 없고, 실시간으로 다수의 센서 값 처리와 모터 제어를 동시에 수행하기에는 성능이 부족하다. 그래서 GPU 서버와 두 개의 Raspberry Pi를 사용하였다.

연산 속도 문제로 인한 Delay 문제를 최소화하기 위해 첫 번째 Raspberry Pi에서는 초음파 센서와 RFID를 처리하고 그 결과값을 소켓 통신을 이용해 두 번째 Raspberry Pi에게 전송한다.

두 번째 Raspberry Pi는 제어 역할을 하며, 멀티 스레드와 소켓 통신을 이용하여 데이터를 받아 제어한다. 총 4개의 Thread로 구성되어있고, 각각 소켓 통신을 통해 라인, 물체, 초음파 센서, RFID로부터 정보를 수신하고 그 정보를 바탕으로 모터와 서보모터를 제어하여 무인 버스 시스템을 구현한다.

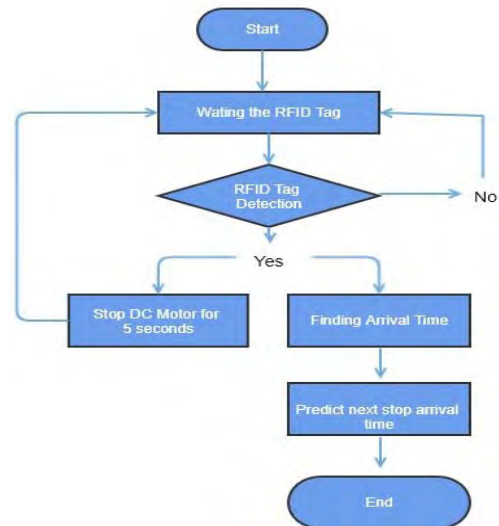
### 3.3. 초음파 센서 분석



[그림 4 초음파 센서 구동 알고리즘 ]

초음파 센서를 이용해 자동차 앞의 거리를 수시로 측정해 장애물을 판단한다. Raspberry Pi와 초음파 센서를 유선 연결하여 거리 측정 간 Delay를 최소화하고, 영상 인식 오류, 카메라 사각지대로 인해 판단하지 못하는 장애물을 초음파 센서로 감지하여 자동차의 안전성을 확대한다.

### 3.4. RFID 분석



[그림 5 RFID 구동 알고리즘 ]

RFID 모듈을 이용해 수시로 통신대기를 하고 버스정류장 도착 시 통신을 이용해 버스정류장 도착에 대한 신호를 전달한다. 이를 통해 보다 정확한 정류장 인식이 가능하며, 더 나아가 버스정류장 도착 시간을 통해 다음 정류장 도착 시간을 예측하여 이용자에게 서비스를 제공한다.

#### 4. 소프트웨어 설계

무인 버스 구현에 사용된 개발 도구는 Python, OpenCV, Keras 등이 있다.

실시간으로 물체와 차선 인식하기 위해 GPU 서버가 무인 버스 내의 카메라를 이용해 찍은 영상을 받아 처리하고 그 결과를 소켓 통신으로 Raspberry Pi에게 전송하는 방식을 사용하였다.

#### 4.1 딥러닝(Deep Learning)을 이용한 물체 인식 알고리즘 분석

GPU가 탑재된 서버는 무인 버스의 카메라를 통해 영상을 받아 물체 인식을 한다.

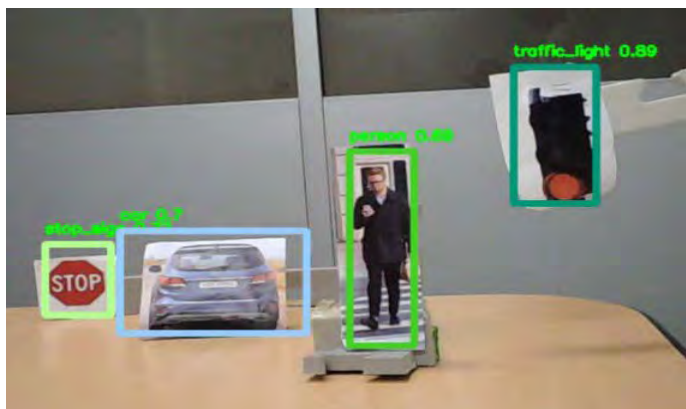
딥러닝 기반 물체인식 알고리즘 중 R-CNN 계열은 처리 속도가 상대적으로 느려 실시간으로 인식이 불가능하다. 그래서 YOLO (You Only Look Once) v2라는 실시간 처리가 가능하고, 높은 성능을 보이는 모델을 사용한다.

4개의 Classes (Person, Car, Traffic light, Stop sign)에 대한 각각 30,000장 이상의 Annotation된 이미지를 학습시켜 얻은 Weights를 저장한다. 그리고 이 Model과 Weights를 이용하여, 이미지 내에 물체의 종류와 위치를 예측한다.

Model의 복잡도와 Training Data의 quality나 quantity에 따라 인식률의 차이가 생기고, GPU의 성능이나 인식할 때 설정할 Threshold에 따라 속도의 차이가 있는데, 본 논문에서 구현한 YOLO 모델은 적은 Class의 수(4개)와 각 30,000장의 학습 데이터에 의해 좋은 인식률을 보이고, Model이 복잡하지 않고 Threshold 값을 높게 사용하기 때문에 상대적으로 화면에 표시되는 속도도 빠르다. (GTX-980M 기준 약 0.38s).

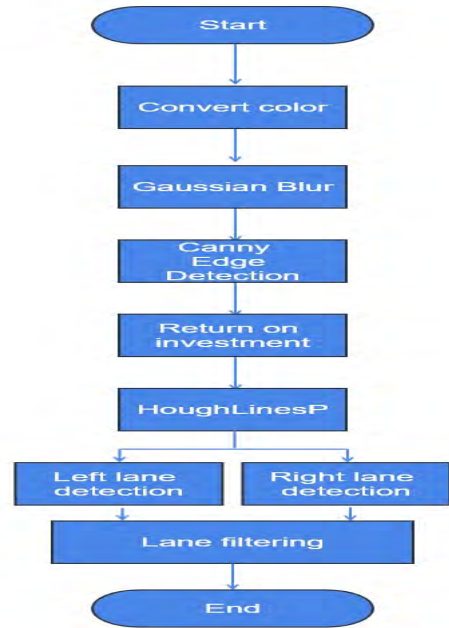
Frame 한 장씩 물체를 인식하기 때문에, 정확하고 안전한 제어 명령을 위해 Frame 4번 연속으로 인식되면 해당 물체가 인식된 것으로 판단하여 멈춤 또는 감속 명령을 내린다. 마찬가지로 인식되었던 물체가 4번 연속으로 인식되지 않으면 재출발 명령을 내린다.

인식되는 물체의 Y좌표로 대략적인 거리를 알아내어 물체가 특정 거리 내에 인식될 때만 멈추거나 감속을 한다. 이때, 신호등을 인식하면, 인식 영역의 RGB값을 추출하여 신호등의 색에 맞게 멈추거나 주행을 한다.



[그림 6 YOLO를 이용한 물체인식 결과]

#### 4.3 차선 추출 알고리즘 분석



[그림 7 차선 추출 알고리즘 순서도]

카메라를 통해 입력받은 영상을 Gray 이미지로 바꾼 뒤 결괏값의 최적화를 위해 Gaussian Blur 필터를 적용한다. 필터링된 영상에 Canny를 적용하여 Edge를 검출, 이진화 처리를 한다.

차선 검출은 영상의 모든 부분을 이용할 필요가 없으므로 영상의 일부를 사용하기 위해 ROI(Region of Interest)를 지정해 준다.

필터링 된 영상에서 직선을 검출하기 위해 HoughLines 혹은 HoughLinesP 함수를 이용할 수 있지만 HoughLines 함수는 연산에 많은 시간이 소요되어 HoughLinesP 함수를 사용하여 직선을 추출한다. 추출된 직선의 각도에 따라 왼쪽, 오른쪽 차선으로 구분하여 차선을 검출한다. (GTX-950M 기준 평균 차선 검출 시간 0.018s)



[그림 8 OpenCV를 이용한 차선 인식 결과]

## 5. 결론 및 향후 연구

본 논문에서는 컴퓨터 비전 기술을 기반으로 자율 주행하는 무인 버스 시스템을 구현하였다. 무인 버스 전방의 카메라로 찍은 영상을 GPU 서버에서 실시간으로 차선 및 물체 인식을 하고, 그 결과를 Raspberry Pi에 전송한다. Raspberry Pi는 그 데이터를 바탕으로 제어를 하여 무인 버스 시스템을 구현하였다. 또한 초음파 센서를 통해 사각지대 탐지 및 운행의 안전성을 확보할 수 있으며, RFID를 이용하여 보다 정확한 버스 정류장 인식을 할 수 있다.

본 논문을 바탕으로 실제로 구현한 무인 버스 운행시스템에서는 무인 버스 내에 GPU 보드가 없기 때문에 GPU가 탑재된 노트북과 카메라를 직접 연결하여서 영상 전송으로 인한 Delay를 최소화시켰다. 또한 Raspberry Pi 두 개를 사용하였는데 이는 연산으로 인한 Delay를 최소화하기 위함이다. Raspberry Pi 하나는 초음파 센서와 RFID를 처리하고, 다른 하나는 멀티 스레드와 소켓 통신을 이용해 차선, 물체, 센서로부터 데이터를 받아 모터와 서보모터를 제어하는 방식을 사용하였다.

향후, 연구 방안으로는 GPU 보드인 Jetson TX2를 사용하여 카메라와 센서로부터 얻은 정보를 무인 자동차 내에서 처리하여 제어하는 방식을 사용하여, 통신으로 인한 Delay를 최소화하여 실시간성을 높일 수 있다. 그리고 Haar Cascade 방식의 물체 인식과 함께 사용하거나 Object Tracking을 구현한다면 보다 정확하고 효율적인 물체 인식이 가능할 것이라고 판단된다. 또한 더 복잡한 Model과 방대한 양의 Dataset으로 다양한 Class를 인식할 수 있는 Model의 구현을 연구할 것이다.

또한 차세대 지능형 자동차에 사용되고 있는 ADAS 기반 기술과 본 논문의 시스템을 결합하여 실제 버스에 적용하면 실제 무인 버스를 개발할 수 있을 것이다.

나아가 딥러닝(Deep Learning) 및 빅데이터를 기반으로 버스 주위의 도로 교통 상황을 판단하여 다양한 지역의 교통 상황 정보를 제공할 수 있다.

## 6. 감사의 글

“본 논문은 2017년 한이음 ICT멘토링 프로젝트의 결과물입니다.”

## 7. 참고문헌

- [1] [국내. 저자 없음] 경찰청.“교통사고의 추세”  
<http://www.police.go.kr/portal/main/contents.do?menuNo=200436>, (2017.09.01 방문)
- [2] [국내. 저자없음] 공공데이터포털, “교통사고 통계현황”  
<https://www.data.go.kr/dataset/15022347/fileData.do>, (2017.09.06.)
- [3] [국내. 저자 없음] 경찰청.“사망자와 부상자”  
<http://www.police.go.kr/portal/main/contents.do?menuNo=200436>, (2017.09.01 방문)
- [4] Taejun Park and Taihon Cho, A Crosswalk and Stop Line Recognition system for Autonomous Vehicles, Journal of

Korean Institute of Intelligent System, Vol. 22, NO. 2, (April 2012), p.154-160

[5] 전국 버스운송사업 조합연합회(부설 한국 운수산업 연구원) “대형 버스교통사고 사례와 그 예방법”, 버스이슈 Vol.2, (May 2014), p.1-4

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016), p.779-788