

# 라즈베리파이와 Yolo를 이용한 운전 도우미 앱 개발

연건모, 류건우, 정어진, 이성욱\*

16106 경기도 의왕시 철도박물관로 157 한국교통대학교 데이터사이언스과

## Developing a Driver Helping App Using Raspberry Pi and Yolo

Kunmo Yeon, Geonwoo Ryu, Eojin Jeong, Songwook Lee\*

Major in Data science, Korea National University of Transportation,  
Uiwang, 16106, Korea

### ABSTRACT

Identifying a person jaywalking through a smartphone application and camera in real time in the vehicle, or determining whether there is a person when making a right turn, We created the Driver Helper app that can perform the function of guiding the time of the bus-only road.

The Driver Helper app uses Raspberry Pi 4 and PiCamara 2 to stream the front of the vehicle, analyze objects through Yolo, and forward information to the server.

The server receives the information and analyzes the objects, and if the conditions are satisfied, the text-to-speech (TTS) is outputted through a smartphone app.

Users can turn on/off only the features they want throught the app.

Keywords: Yolo, Raspberry Pi, text-to-speech (TTS)

## I. 서론

최근 교통사고의 주요 원인 중 하나는 무단 횡단과 우회전 시 보행자의 인지 부족이다. 이러한 문제는 운전자의 주의력만으로는 해결하기 어려워지고 있으며, 기술의 발전을 통해 보다 효과적인 해결책을 찾고자 하는 노력이 지속되고 있다.

우리는 이러한 문제를 해결하기 위해 차량 내부에서 실시간으로 스마트폰 애플리케이션과 카메라를 이용하여 보행자의 무단 횡단 여부를 식별하고, 우회전 시 보행자의 존재 여부를 파악하며, 버스 전용 도로의 사용 시간을 안내하는 '운전 도우미 앱(Driver Helper app)'을 개발하였다. 이 시스템은 Raspberry Pi 4와 PiCamera 2를 활용하여 차량 전방을 스트리밍하고, Yolo 모델을 통해 객체를 분석한 후, 서버로

정보를 전달하는 방식으로 작동한다. 서버는 수신된 정보를 분석하여 특정 조건이 충족되면 텍스트를 음성으로 변환(TTS)하여 스마트폰 앱을 통해 사용자에게 안내한다. 사용자는 애플리케이션을 통해 원하는 기능만을 선택적으로 켜고 끌 수 있다.

운전 도우미 앱은 운전자의 시야를 넓히고, 보행자를 보다 빠르고 정확하게 인지할 수 있도록 돕는 역할을 한다. 차량이 우회전을 할 때 도로에 보행자가 있는지 여부를 실시간으로 파악하여 운전자에게 경고를 줌으로써 사고를 예방할 수 있다. 또한, 버스 전용 도로의 사용 시간을 안내함으로써 운전자가 효율적으로 도로를 이용할 수 있도록 돕는다. 이러한 기능은 모두 운전자의 안전과 편의를 증대시키는 데 목적이 있다.

\* leesw@ut.ac.kr, Tel: 031-460-0585

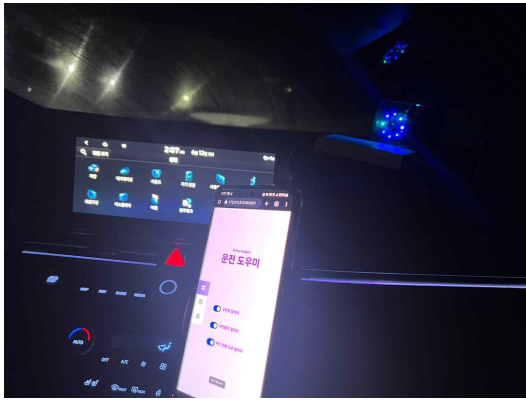


Fig. 1. Example of using the Driver Helper app

## II. 시스템 설계

프로젝트의 시스템 설계는 PiCamera 2를 장착한 RaspberryPi 4와 Yolo를 실행할 Client, 정보를 전달 받아 조건에 맞게 분류하고, 사용자에게 분류된 정보를 전달할 Web Server, 마지막으로 Smart Phone을 통한 제어와 TTS 음성을 출력할 Application 이렇게 네가지 구성 요소로 나눌 수 있다.

RaspberryPi 4와 PiCamera 2를 사용해 차량 전방의 영상을 실시간으로 스트리밍하며, Client로 영상을 전송한다. 이후 Client는 전송받은 영상에서 Yolo 알고리즘을 통해 객체를 분류하고, 조건에 맞게 WebServer로 데이터를 전송한다. WebServer에서는 Node 모듈을 설치하여 GUI를 구성하고, 특정 데이터를 받을 때마다 Application에서 TTS 음성이 출력되도록 한다. Application은 Expo프레임워크를 사용한 React Native로 WebServer의 화면을 앱으로 출력해주는 WebView방식을 사용하였다.

Client 와 WebServer간의 통신은 HTTP 프로토콜을 통해 이루어지며, Express 프레임 워크를 사용하여 Restful API 엔드포인트를 정의하고 미들웨어로 요청을 처리한다.

이를 통해 클라이언트는 특정 경로로 데이터를 전송하고 서버는 이를 받아 필요한 처리를 수행하여 응답을 반환한다.

## III. 시스템 구현

### 1. 데이터 수집

데이터 수집은 Roboflow와 Aihub등을 사용해 이루어졌으며, 수집한 데이터는 버스 전용 도로 표지판(전일제, 시간제), 사람, 신호등, 횡단보도, 차선 등으로 구성된다. 추가로 데이터의 정확도를 높이기 위해 Youtube 영상들을 사용하였다.

## 2. 소프트웨어

### 1) 스마트폰 애플리케이션

개발 환경으로는 Expo프레임워크를 사용해 개발되었으며, 앱은 GUI와 백그라운드에서 실행되는 기능들을 효율적으로 처리할 수 있도록 설계되었다.

주요 기능으로는 실시간 TTS, Yolo 모델을 사용한 객체 분석 등이 있다.

다음 표는 개발을 소프트웨어 도구와 그 용도를 나타낸 표이다.

도구	용도
debian LINUX	라즈베리파이 OS
Visual studio code	Js, expo 개발툴
pycharm	Python 개발툴
putty	원격접속
EXPRESS	node.js 프레임워크
Expo	react native 프레임워크
Roboflow	Yolo 라벨링
Yolo v5	딥러닝 기반 실시간 이미지 검출 시스템
OpenCV	영상 라이브러리
Tenserflow	기계학습 라이브러리
Pytorch	딥러닝 라이브러리
javascript, css, html, node.js, python, react native	개발 언어

### 2) 웹 서버

Web Server는 JavaScript와 Express 프레임 워크를 사용해 개발되었다. 또한 동적인 웹 페이지를 생성하고 렌더링하기 위해 EJS 템플릿 엔진을 사용하였고, Node.js를 통해 서버를 구성하였다.

Web Server에서는 Client로부터 받은 데이터를 분석하고 조건에 맞게 TTS 음성으로 출력시킨다.

### 3) 클라이언트

Client는 실시간으로 객체를 탐지하는 Yolo 알고리즘을 사용하기 위해 Python으로 개발되었다.

Client에서는 RaspberryPi로부터 전송된 스트리밍 영상을 분석해 객체를 탐지하고, 탐지된 객체를 분석해 특정 정보를 WebServer로 전송한다.

## 3. 하드웨어

### 1) 라즈베리 파이 및 파이카메라

처음엔 Client를 따로 두지 않고 RaspberryPi 4에서 Yolo 알고리즘을 사용하여 곧바로 WebServer로 보내는 방식을 사용하였다. 하지만, RaspberryPi 4의 물리적인 한계로 인해 스트리밍되는 영상의 Fps가 너무 낮았고, 사용하는데 문제가 있다 느껴 결국 외부 컴퓨팅 장치를 사용하게 되었고, RaspberryPi 4

는 실시간 스트리밍의 기능만 남기게 되었다.

PiCamera 2는 차량 전방의 영상을 실시간으로 RaspberryPi로 전송한다. RaspberryPi 4는 PiCamera 2로부터 받은 영상을 같은 IP에서 볼 수 있게끔 스트리밍 하게 된다. 이후, 포트 포워딩을 통해 다른 IP에서도 스트리밍 되는 영상을 전송 받을 수 있게 된다.



Fig. 2. Example of Screen with yolo algorithm applied

#### 4. 시스템 통합

운전 도우미 앱은 여러 구성 요소가 유기적으로 결합되어 작동한다. PiCamera 2가 실시간으로 촬영한 영상은 Raspberry 4로 전송된다. Raspberry 4에서는 영상을 ip에 스트리밍하고 포트 포워딩을 통해 이를 다른 ip에서도 접근할 수 있게 한다. Client는 스트리밍된 영상에 접근해, Yolo 알고리즘으로 객체를 분석하고, 이를 HTTP 프로토콜을 통해 Web Server로 전송한다. Expo 프레임워크를 통해 이 웹을 WebView 형식으로 앱으로 출력하고, 사용자는 이를 통해 TTS 음성을 들을 수 있게 된다.

이 앱의 장점은 실시간 객체 탐지와 정보 전달이 가능하다는 점이다. 운전자는 실시간으로 음성 안내를 받을 수 있어 즉각적인 반응이 가능하고, 필요한 기능만을 선택적으로 활용할 수 있어 불필요한 알람을 줄일 수 있다.

### IV. 결론 및 향후 과제

운전 도우미 앱은 운전자의 시야를 넓히고, 보행자를 보다 빠르고 정확하게 인지할 수 있도록 돕는 역할을 한다. 차량이 우회전을 할 때 도로에 보행자가 있는지 여부를 실시간으로 파악하여 운전자에게 경고를 줌으로써 사고를 예방할 수 있다. 또한, 버스 전용 도로의 사용 시간을 안내함으로써 운전자가 효율적으로 도로를 이용할 수 있도록 돕는다. 이러한 기능은 모두 운전자의 안전과 편의를 증대시키는 데 목적이 있다.

정확도를 높이기 위해서 앞으로 더 많은 데이터를 수집하고 모델의 성능을 향상시키는 작업의 필요성이 있다. 또한, 다양한 주행 환경에서의 테스트를 통해 시스템의 신뢰성을 검증하고, 실제로 앱을 사용해 나가면서 피드백을 반영하여 기능을 개선해 나갈 계획이다. 나아가, 음성 인식 및 인공지능 기술을 활용하여 더 많은 기능을 추가하고, 보다 정교한 운전자 보조 시스템을 개발하는 것이 향후 과제이다.

우리는 Raspberry Pi 4로부터 포트포워딩을 통해 다른 ip에서 데이터를 받아들 수 있게끔 구성하였는데, 보안상 안전하지 않다는 문제점과 스트리밍되는 영상의 프레임이 너무 낮다는 을 발견하였다. 우리는 추후에 이러한 여러 문제점 들을 보완하기 위해 노력할 것이다.

### 감사의 글

이 논문은 2024년 한국교통대학교 지원을 받아 수행하였음.

### 참고문헌

1. Raspberry Pi 4, <https://www.raspberrypi.org/>, 2024
2. Yolo v5, <https://docs.ultralytics.com/>, 2024
3. OpenCV, <https://opencv.org/>, 2024
4. TensorFlow, <https://www.tensorflow.org/>, 2024
5. Pytorch, <https://pytorch.org/>, 2024
6. Node.js, <https://nodejs.org/>, 2024
7. putty, <https://www.putty.org/>, 2024
8. AiHub, <https://www.aihub.or.kr/>, 2024
9. expo, <https://expo.dev/>, 2024
10. roboflow, <https://roboflow.com/>, 2024
11. cuda, <https://developer.nvidia.com/cuda-toolkit/>, 2024