

YOLOv4와 OpenCV 모델을 이용한 자율 주차 시스템

이충녕, 김종민, 남수진, 채승호*

한국공학대학교

lc9902130509@tukorea.ac.kr, *shchae@tukorea.ac.kr

YOLOv4 and OpenCV Based Automatic Parking System

Chungnyeong Lee, Jongmin Kim, Sujin Nam, Seong Ho Chae*

Tech University of Korea

요 약

본 논문에서는 운전자의 편의를 위한 YOLOv4 및 OpenCV 기반 자율 주차 시스템을 제안하고 프로토타입을 구현한다. 제안된 시스템은 YOLOv4를 활용한 주차 공간 탐색 및 인식 기능과 OpenCV를 이용한 거리 계산 및 자율 주차 기능으로 구성된다. 본 시스템은 운전자에게 웹 기반의 원격 차량 제어가 가능하게 함으로써 사용자의 편의성을 제고하고, 운전자의 주차 소요시간 절약과 함께 자율 주차를 가능하게 함으로써 주차고통비용을 효과적으로 감소시킨다.

I. 서 론

국토교통부의 자동차 등록대수 현황 보고에 따르면, 2022년 현재 전국 차량 등록대수가 2507만 180대로 2015년에 비해 약 19% 이상 급격히 증가한 것으로 나타났다[1]. 이에 따라, 부족한 주차 공간으로 인해 발생하는 '주차고통비용'(운전자의 스트레스, 시간 낭비, 교통혼잡 비용 등을 통칭)이 연간 38조~41조 원에 달할 것으로 추정하고 있다. 이러한 문제는 국내 자동차 등록대수 증가 추세와 함께 더욱 가속화될 것으로 전망되고 있으며, 이를 효과적으로 해결하기 위한 방안에 대한 연구의 필요성이 대두되고 있다. 이에 따라, 본 논문에서는 '주차고통비용'을 효과적으로 감소시킬 수 있는 자율 주차 시스템을 제안하고 프로토타입을 제작한다.

II. 제안 시스템 구성



그림 1. 주차 구역 인식 알고리즘 구성도 및 동작

그림 1은 제안하는 자율 주차 시스템의 전체 구성도 및 동작을 보여준다. 제안된 자율 주차 시스템은 주차 공간 탐색 및 인식, 거리 계산, 주차 수행으로 총 3개의 하부 시스템으로 구성된다. 제안된 시스템의 차량은 Raspberry Pi 4B와 Raspbot AI 비전 로봇 카[2]를 활용하여 구성하고, 서버는 사전에 수집한 주차 공간 사진 데이터를 통해 입력받은 영상 데이터 중 주차 공간을 찾고 해당 공간의 주차 가능 여부를 판단하는 모델을 YOLOv4[3] 활용하여 학습시킨다.

각 하부 시스템의 구체적인 세부 동작 및 기능은 다음과 같다.

1) **(주차 공간 탐색 및 인식 기능)** 그림 2는 제안하는 주차 공간 탐색 및 인식 알고리즘을 보여준다. 차량은 서버로부터 360도 회전이 가능한 Raspberry Pi의 카메라v2 모듈을 장착하고 주변 영상 정보를 획득한다. 운전자가 차량 외부에서 원격으로 신호를 송신하면, 차량은 획득한 영상 정보를 서버로 전송한다. 서버는 사전에 학습시킨 인공지능 모델을 기반

으로 전송받은 이미지로부터 주차 공간을 인식한다. 만약 주차 공간 인식에 성공하고, 주차 가능 공간이 있다면 서버에서는 계산된 모든 결과(주차 가능 공간 수, 주차 가능 공간 좌표 등)를 차량에게 전송한다.

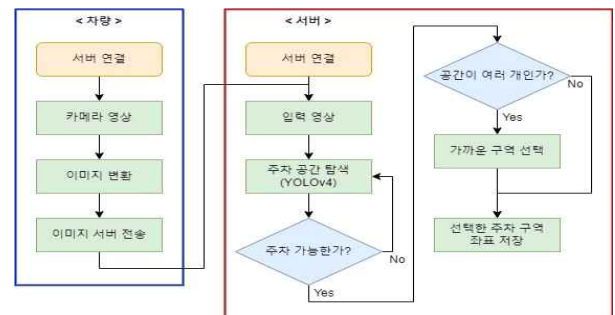


그림 2. 주차 공간 탐색 및 인식 알고리즘

2) **(거리 계산 기능)** 거리 계산 단계는 차량의 카메라 각도와 서버로부터 받은 주차 구역 좌표를 토대로 차량의 목표 지점(주차 가능 공간)을 설정하고, 해당 목표 지점까지의 방향 및 거리의 계산을 수행한다. 차량은 OpenCV[4]를 기반으로, 캐니 에지(Canny Edge)를 통해 주변 주차 공간의 주차선을 인식하는 영상처리를 수행한다. 해당 영상처리 결과와 서버로부터 받은 정보를 이용하여 실시간으로 위치를 조정하면서 차량은 목표 지점으로 이동한다. 차량의 이동은 파이썬 기반 모터의 PWM 신호 제어를 통해 수행된다.

3) **(주차 기능)** 주차 단계는 차량이 안전하게 주차 공간에 주차를 수행하는 단계이다. 차량이 주차 공간에 진입하면 차량의 카메라 방향은 정면을 향하도록 전환한다. 차량은 실시간으로 양옆 주차선과 끝 주차선을 인식하며 주차 공간의 중앙을 OpenCV를 통해 계산한다. 해당 중앙 계산 값을 통해 차량이 중앙을 유지하며 주차할 수 있도록 차량은 실시간 위치를 조정한다. 끝 주차선과 차량 사이의 거리를 통해 일정 거리만큼 가까워지면 주차를 완료한다.

모든 과정을 수행하는 중, 차량의 초음파 센서를 활용하여 이동 중인 차량과 보행자 등 위험 요소를 인지한다. 위험 요소가 초음파 센서를 통해

감지가 되면, 차량은 즉시 정차되며 위험 요소가 제거될 때까지 대기한다. 추가로 파이썬의 웹 프레임워크인 Flask[5]를 통해 웹 서비스를 구현하여, 운전자에게 차량 원격 제어 GUI를 제공한다.

III. 구현결과

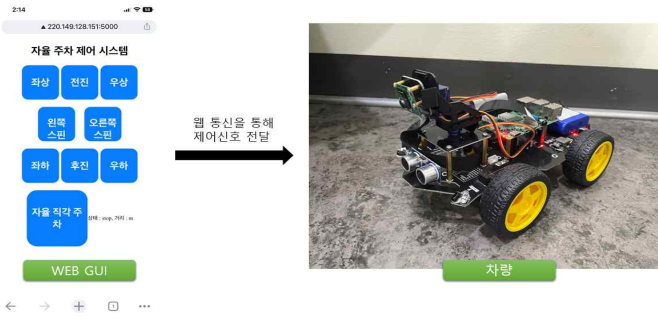
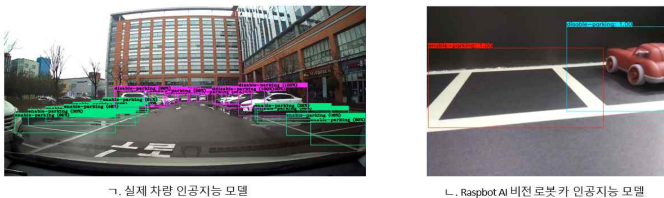


그림 3. 웹 GUI를 통한 차량 제어

그림 3은 운전자에게 제공하는 웹 GUI와 실제 제작한 프로토타입 구현 결과물을 보여준다. 운전자는 해당 웹 GUI를 통해 무인 상태로 원격으로 차량 제어(좌상, 전진, 우상, 왼쪽 스텔, 오른쪽 스텔, 좌하, 후진, 우하, 자율 직각 주차)가 가능하게 구현하였다. 해당 기능을 통해 운전자는 목적지에 도착 후, 차량에게 원격으로 자율 주차 신호를 보내면 주차가 가능하게 함으로써 주차에 소비되는 시간을 상당히 절약할 수 있다.

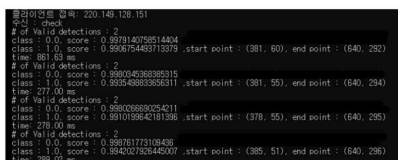


ㄱ. 실제 차량 인공지능 모델

ㄴ. Raspbot AI 비전 로봇 카 인공지능 모델

그림 4. 인공지능 모델 결과

그림 4는 실제 영상(ㄱ. 블랙박스 영상, ㄴ. Raspbot AI 비전 로봇 카의 Raspberry Pi 카메라v2 영상)에서의 주차 공간 및 주차 가능 여부를 학습 모델을 통해 검출한 결과를 보여준다. 주차 공간은 색에 상관없이 직사각형 공간에 해당한다. 해당 주차 공간 중 주차 가능한 곳은 ㄱ은 초록색 직사각형, ㄴ은 빨간색 직사각형에 해당하며, 주차 불가능한 곳은 ㄱ은 분홍색 직사각형, ㄴ은 하늘색 직사각형에 해당한다. 본 논문에서는 Raspbot AI 비전 로봇 카를 사용했지만, 추후 연구 확장성을 위해 실제 차량에 해당하는 인공지능 모델도 추가로 학습을 진행하였다. 모델 학습에 사용한 훈련 데이터는 모두 직접 촬영해 수집하고 YOLOv4에서 제공하는 데이터 라벨링 툴을 통해 훈련 데이터를 제작했다. 라벨링은 툴을 통해 영상 데이터에서 주차 공간이 직사각형 모양의 박스 안에 포함되도록 영상 데이터에 박스를 그리고, 해당 공간의 주차 가능 여부를 라벨링하는 방식으로 진행했다. 제작된 훈련 데이터를 통해 입력받은 영상 중 주차 가능 공간을 찾고, 해당 공간의 주차 가능 여부를 판단하는 모델을 학습시켰다. 실제 차량 모델 경우 800여 장, Raspbot AI 비전 로봇 카인 경우 300여 장의 사진 데이터를 사용하였으며, 최종 인공지능 학습 모델의 평균 손실 값은 실제 차량 및 Raspbot AI 비전 로봇 카 각각 0.3895, 0.4159로 산출됐다.



ㄱ. 서버 계산 과정



ㄴ. 차량이 보낸 영상 데이터

그림 5. 주차 공간 인식 및 탐색 결과

그림 5는 서버가 차량으로부터 받은 영상 데이터를 토대로 주차 공간 탐색 및 인식 계산 결과를 차량으로 보내는 과정을 보여준다. 계산은 사전에 학습된 인공지능 모델을 통해 수행한다. 그림 5를 보면, 서버에서는 주차 공간을 두 곳으로 판단했고, 그중 주차 가능 공간 한 곳(class 1.0에 해당), 주차 불가능 공간(class 0.0에 해당) 한 곳으로 판단하였다. 주차 공간을 판단할 때는 정확도가 80%(score 0.80) 이상인 기준을 통해 해당 기준을 만족한 공간만을 주차 공간으로 판단하였다. 좌표 데이터의 경우에는 직사각형 모양으로 주차 공간을 판단하기 때문에 두 점의 좌표(직사각형의 좌측 상단과 우측 하단의 꼭짓점)를 전송하였다. 차량으로부터 전송하는 영상 데이터는 MJPG-streamer[6]를 활용해 전송했다.

그림 6은 주차 가능 공간에 자율 주차하는 과정을 보여준다. 해당 그림에서 주차 공간은 초록색 직사각형 공간이다. 그림 6의 차량은 서버로부터 주차 가능한 공간(초록색 직사각형 공간)이 존재하고 해당 공간 관련 정보를 받는다. 이를 통해 차량은 해당 공간에 자율 주차를 수행한다. 1단계~3단계는 차량이 주차 가능 공간에 진입하는 과정을 보여주고 있으며, 4단계~5단계는 영상 처리를 통해 차량 위치 조정 후 주차하는 과정을 보여준다. 6단계는 차량의 주차 완료 단계를 보여주고 있다.

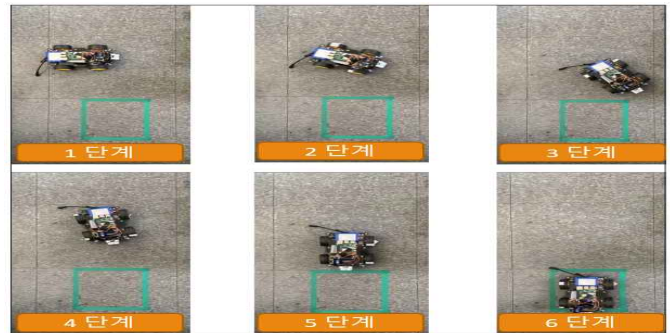


그림 6. 자율 주차 수행 과정

IV. 결론

본 논문에서는 운전자에게 편의를 제공하고자 YOLOv4와 OpenCV 모델을 활용해 자율 주차 시스템을 설계하고 프로토타입을 제작하였다. 차량을 제어하는 데 있어 본 시스템은 운전자에게 웹 GUI를 통해 편리한 접근성을 제공한다. 제안된 시스템을 통해 운전자는 주차하는데 소요되는 시간을 크게 절약하고 스트레스로부터 해방된다. 따라서 운전자의 삶의 질을 개선하고 주차고통비용을 크게 절감할 수 있다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지능정보화혁신 인재양성(Grand ICT연구센터) 사업의 연구결과로 수행되었음. (IITP-2022-2020-0-01741)

참 고 문 헌

- [1] <https://news.mt.co.kr/mtview.php?no=2022060122162687735>
- [2] <https://www.icbanq.com/P012114433>
- [3] <https://github.com/AlexeyAB/darknet>
- [4] <https://opencv.org/>
- [5] <https://flask.palletsprojects.com/en/2.2.x/>
- [6] <https://github.com/jacksonliam/mjpg-streamer>