

과제 수행 결과 보고서

2019학년도 AD 프로젝트 보고서

택배 배달大作전

국민대학교 소프트웨어융합대학

소프트웨어학부

창업연계공학설계입문

6분반 4조

20191684 최창연

20191672 조현민

20191691 황영서

20191678 최영락

20163660 주 가

담당교수: 한광수

2019 년 12월 15일

택배 배달大作전

1. 서론

택배는 우리의 삶에서 많은 부분을 차지한다. 최근에는 인터넷 쇼핑이 대중화되어 택배의 중요성이 더욱 부각된다. 국토교통부는 2018년 기준 우리나라의 연간 택배 물량이 약 25억 4300만 건 이라고 발표했다. 이를 1인으로 환산하면 일주일에 한 번 정도이다. 여기에 통계에는 잡히지 않는 배달음식, 사설 퀵서비스 등을 합한다면 택배 유동량은 더 많아질 것이다. 몇 년 전부터 자동차를 자율주행으로 만드는 것이 대세가 되었다. 물론 택배도 예외는 아니다. 하지만 택배는 일반 자동차와 신경을 쓸 부분이 다를 수 있기에 우리 조는 택배 차량에 맞게 자율주행 프로그램을 짜기로 했다.

우선 택배의 기본 요건인 짐을 나르는 것은 짐이 담길 컨테이너에 갈고리를 달고, xycar의 뒤에 갈고리와 대응하는 홈을 만들어놓을 것이다. xycar가 주행을 시작할 때, 뒤로 조금 가게 해 차와 컨테이너를 연결시킨 뒤 출발시킬 것이다.

그 다음으로 xycar의 센서 중 카메라를 이용하여 표지판을 인식시킬 것이다. 최대한 여러 색을 준비하여 표지판 당 행동을 정해놓고 그것을 실행시킨다면 특정한 역할을 수행시킬 수 있을 것이다.

또한 IMU센서를 이용해 현재 차량의 기울기나 흔들림을 파악할 수 있다. 이를 통해 과속방지턱이나 오르막길 등의 경사가 바뀌는 지점이 있으면 그 구간을 IMU센서로 인식해 속도를 줄여 물건의 안전을 확보할 수 있을 것이다.

마지막으로 가장 중요한 차선 따라가기. 카메라로 차선을 인식해 좌우 차선의 위치 혹은 인식 유무로 지향점을 설정하여 최대한 차선을 나가지 않고 주행할 수 있도록 프로그램을 설정했다.

2. 연구의 목적, 내용

2.1 연구의 목적

요즘 AI 자동화 시대가 열리면서 택배 사업 또한 그 변화의 바람을 맞기 시작했다.

대표적으로,

- 아마존의 무인 드론 배달 서비스
- 무인 배송 서비스 kiwi Bot

등이 있다.

이 시대의 흐름을 따라가며 자동 택배를 상용화를 실현하기 위해 테스트 모듈로서 작동시켜보기위한 Xycar 무인 자동 배달 택배운반차량을 만들어보고자 하였다.

2.2 연구의 내용

Xycar(무인 택배 운반 차량)의 자율주행(Line Tracing)기술, 택배(상차장)인식 기술, 돌발 상황 감지 및 대처(초음파센서의 거리감지) 기술, 각 기능을 수행할 이미지 or 색상 인식(OpenCv) 기술, 경사의 기울기에 따른 속도조절(IMU 센서) 등을 조합하여 배송 출발지로 부터 배송 도착지까지 안전하게 배송을 완료할 Xycar를 개발하였다.

2.2.1 연구의 목표

Line Tracing: Xycar는 배송 출발지로부터 배송 도착지까지 가는 도로의 차선을 인식해 차선안에서 자율주행을 안전하게 수행할 수 있도록 한다.

OpenCv Image Recognition: 각 Image마다 해당하는 Image의 기능을 수행할 수 있도록 OpenCv를 이용하여 Deep Learning의 과정을 거쳐 구현한다.

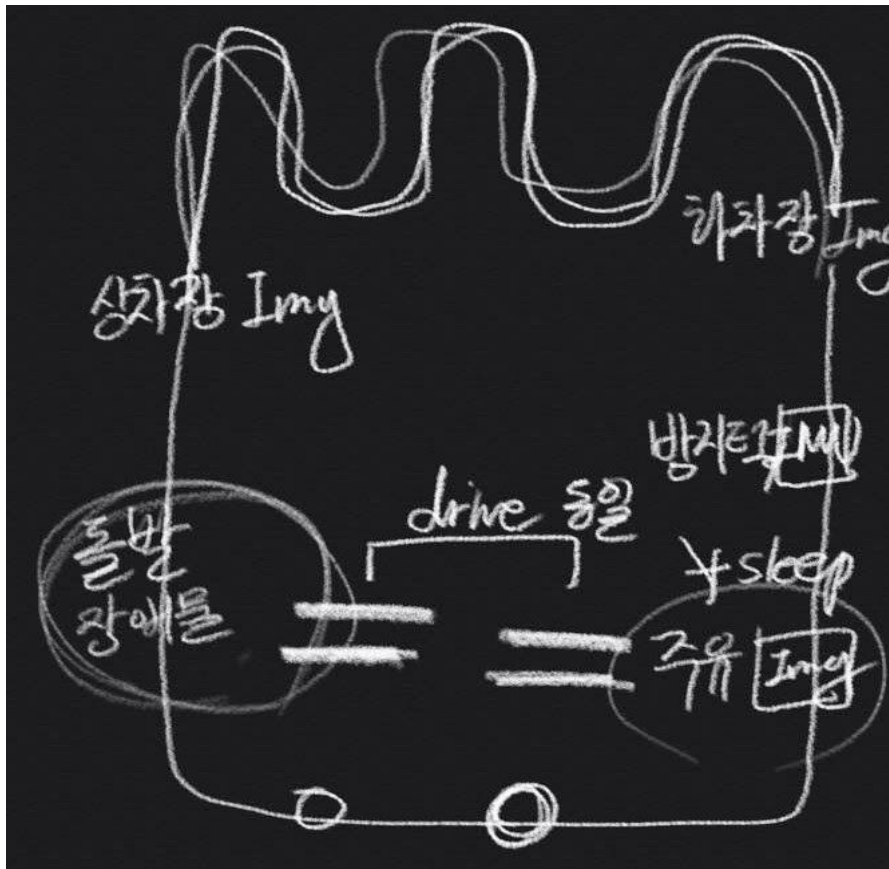
OpenCv Color Recognition: 각 Color마다 해당하는 Color의 기능을 수행할 수 있도록 OpenCv를 이용하여 Deep Learning의 과정을 거쳐 구현한다.

Ultrasonic: Xycar 주변에 있는 물체들 간의 거리를 초음파를 이용해 검출하여 일정 거리 이하가 되면 물체를 피해갈 수 있도록 구현한다.

IMU - Principal Axis: Roll Axis, Pitch Axis, Yaw Axis의 세축 중 Pitch Axis의 Pitch 값을 이용해 차의 기울기의 변화를 검출하여 차의 기울기 때마다 해당하는 기능을 수행할 수 있도록 IMU Sensor를 이용하여 구현한다.

2.2.2 연구의 방법

2.2.2.가 : 연구 과제 설정 :



자율 주행 스튜디오의 0자 트랙의 한 부분에서 출발하여 각 지점들의 과제들을 해결하여 마지막 목적지까지 도달한다.

자율 주행 : 자율 주행 스튜디오에 Xycar이 주유나 주차 등 어떤 다른 행동을 해야하지 않는 이상 그려져 있는 차선을 벗어나지 않고 가도록 한다.

표지판 인식 : 각 지점들마다 어떤 표지판을 세워 두어 빨간색 표지판은 정지, 노란색 표지판은 어린이 보호구역(서행), 파란색 표지판은 주행(차선 옆으로 잠시 빠졌다가 대기 후 다시 진입), 초록색 표지판은 목적지(오른쪽으로 빠져서 정차)로 하여 Xycar이 해당 표지판을 지나갈때 인식하여 해당 행동을 하게 한다.

장애물 인식 : 특정 지점에 장애물이 설치되어 있어 장애물을 만나면 비켜가도록 한다.

경사면 주행 : Xycar이 경사면을 만나서 올라가고 내려갈때 서행을 하도록 한다.

2.2.2.나 : 과제 해결 방법 :

자율 주행 : Xycar의 카메라의 영상을 받아와서 OpenCv를 통하여 분석하여 바닥에 있는 흰색 선만을 받아와서 선의 움직임에 따라 움직이게 한다.

표지판 인식 : Xycar의 카메라 영상을 받아와서 OpenCv를 통하여 각각의 표지판들을 Tensorflow 등을 통하여 학습시켜 각각의 표지판들을 구별할 수 있게 한다. 그리하여 주행 중에 각각의 표지판들을 보았을때 해당하는 특정 행동을 취하게 한다. 이 방법을 사용하지 못할 경우 OpenCv를 통해 각 표지판의 색깔 범위를 설정하여 Xycar의 카메라의 특정 영역에서 어떤 색깔 범위에 있는 픽셀이 그 영역의 대다수를 차지하게 된다면 표지판을 인식한것으로 하여 특정 행동을 취하게 한다.

장애물 인식 : Xycar의 Ultrasonic을 통하여 앞의 장애물이 있다면 앞의 Ultrasonic 센서의 값이 낮아질 것이므로 정면의 센서가 측정한 값이 일정 수치 이하로 떨어지면 우로 나가서 좌측 센서가 측정한 값이 다시 아무것도 측정 되지 않아 높아질 때까지 앞으로 가다가 다시 코스로 들어와 원래 주행을 이어나간다.

경사면 주행 : Xycar의 IMU 센서를 통하여 횡축에 해당하는 y축의 변화량인 pitch값이 일정 이상 변화하면 속도를 느리게 가도록 한다.

2.3 연구의 한계점

- Deep Learning:

기본적으로 raw data를 받고 그것을 가공하는 과정에 있어서 Filter의 역할을 해줄 수 있는 Deep Learning기술을 잘 구현하지 못하였다. 이에 따라 data값이 튀면서 생기는 오류들, 미세한 오류부터 큰 오류까지 등의 오류를 처리하는데에 많은 어려움이 있었다.

- IMU Sensor:

IMU Sensor가 Pitch data를 읽어들이는데에 있어서 값을 받는데에 딜레이가 심했다. 이에 따라 연구의 목표중 하나인 '경사로를 올라가면서 바로 속도가 줄어드는 상황'에 대한 구현이 잘 이루어지지 못했다. 반응속도가 느린탓에 경사를 어느정도 올라간 후에 속도가 줄어들기 시작했다.

- 후방 카메라의 부재:

전방카메라로 차선을 인식하여 라인트레이싱을 하는 것과 같이 후방카메라로 물체를 인식하여 후방카메라의 가운데에 물체가 오게끔 하도록 코드를 구현하여 택배박스가 스스로 물체를 찾아가게 만들려고 했으나, 후방 카메라가 없어서 자동으로 택배를 찾아가게끔 구현하지 못하였다.

3. 연구의 결과

3.1 자율주행

무인배달과 드론배달의 핵심 기술은 자율주행을 통해 원하는 목적지까지 가는 것이다.

- > 드론이 아닌 차량이기에, 라인 트레이싱을 통해 자율주행을 할 수 있게 했다.

3.2 장애물 인식

장애물을 초음파 센서로 인식하여 피해간다.

- > 장애물의 거리를 잘못 계산할 수 있기에 필터를 이용하여 인식한다.
- 또한, 기본적으로 라인 트레이싱을 할 때 앞쪽의 장애물을 피해간다.
- > 라인 트레이싱 코드부분에서 장애물이 라인을 가리게 되면, 라인을 잃은 것으로 간주, 피해가고 다시 라인을 찾아간다.

3.3 표지판 인식

각 색상 별로 표지판에 대한 행동을 코드로 구현했다.

> 잠시 정지

- >> 빨간색
- >> 주행을 정지시키고, 몇 초후 서행 후 원래의 속도로 주행한다.

> 주유소

- >> 파란색
- >> 주행 방향의 오른쪽으로 빠진 후, 주유할 시간 동안 정차한다.
- >> 그 후에 다시 주행도로로 복귀한다.

> 어린이 보호구역

- >> 노란색
- >> 주행 중에 이 구간에서는 서행을 한다.
- >> 노란색을 인식하지 못하면 원래 속도로 주행한다.

> 목적지

- >> 초록색
- >> 목적지의 주차장에 정차한다.

3.4 경사면 안전 주행

Pitch 값의 변화에 따라 속도를 조절하도록 코드를 구현했다.

> 그러나 경사면을 제작하지 못했다.

> 코드의 출력 내용을 토대로 바퀴의 속도를 관찰했을 때 알맞게 실행된다.

4. 결론.

프로젝트를 수행하기 전에 우리는 3가지 의의를 생각했다.

첫째, 배달 시에 일어날 수 있는 교통사고 및 각종 접촉사고

기본적으로 우리의 실습 범위 내에선 사고를 예방하기 위해 여러가지 방법으로 구현을 해냈다. 그러나 실제 도로나 상황에서는 어떠한 일이 일어날지 그 누구도 모르기에 현실의 데이터와 접목시켜서 더 광범위한 사고예방 시스템을 만들어 낼 수 있도록 노력해야 한다.

둘째, 배달 시간 지키기와 안전하고 효율적인 주행

배달 시에 빠른 속도로 주행하면서, 안전까지 보장하는 것은 매우 어려운 일이다. 그러나 현대 사회에서는 그 두가지를 충족시켜야 한다는 매우 어려운 과제가 존재한다.

그렇기에 지금은 단순히 주행 속도를 빠르게 설정하는 방법밖에 없지만, 앞으로는 실시간으로 받아오는 정보를 자율주행 자동차의 접목시키고, 더 나아가 드론을 통해 무거운 물건을 배달할 수 있는 세상이 올 수 있도록 개발할 것이다.

셋째, 밤샘운전의 위험성 낮추기

근본적으로 자율주행은 밤샘운전의 위험을 없앨 수 있을 것인가를 생각해야한다. 물론 사람이 운전하지 않기에 없앨 수 있을 것이라고 생각하지만, 밤샘운전의 위험성은 사람의 졸림 뿐만 아니라 어두움으로 인한 대처능력의 부족이라고 생각한다.

그렇기에 자율주행에서 뛰어난 대처능력이 필요할 것이다. 현재로서는 그러한 능력을 요하는 인공지능을 자동차에 탑재하는 것이 가장 해결책에 가까워 보이지만, 앞으로 좀 더 많은 시간과 노력을 투자한다면 더 나은 해결책을 얻을 수 있을 것이다.

