포스코 청년 AI & Big Data 아카데미 16기



**User Tracking**

**AI Cart**

**A반 2조**

**강지우 김호준 노소은 조현영 박나희**

**목차**

**1. 프로젝트 소개** ·······························**3**

*1.1. 프로젝트 개요*·····························*3*

*1.2. 추진 배경 및 현황*···························*3*

1.2.1. 추진 배경·····························3

1.2.2. 관련 기술 현황···························4

**2. 구현 과정**··································6

*2.1. 전체 구조도*·······························*6*

*2.2. 하드웨어*································*6*

2.2.1. 카트 도안·····························7

2.2.2. 제작 과정·····························8

2.2.3. 결과································9

*2.3. 음성 인식*·······························*10*

2.3.1. 챗봇 기능 및 아이디어 소개·····················10

2.3.2. 진행 과정·····························12

2.3.3. 결과································13

*2.4. 사용자 Tracking*····························*13*

2.4.1. 기술 및 아이디어 소개·······················13

2.4.2. 진행 과정·····························19

2.4.3. 결과································20

*2.5. 사용자 Detecting*····························20

2.5.1. 기술 및 아이디어 소개························21

2.5.2. 진행 과정······························22

2.5.3. 결과································22

**3. 기대효과 및 개선기회**····························24

*3.1. 기대효과*································24

*3.2. 개선기회*································*24*

3.2.1. 하드웨어······························24

3.2.3. 소프트웨어·····························25

**4. 조원 상호 평가**·······························26

**References**···································29

**1. 프로젝트 소개**

* 1. **프로젝트 개요**

4차 산업 혁명이 대두되면서 여러 분야에서 기술 혁신이 이루어지고 있다. 유통업계도 이 흐름을 놓치지 않고 여러 혁신을 거듭하고 있다. 스마트 카트란 자율주행, 물체 인식 등 사용자의 편의를 위한 기술이 집합되어 있는 쇼핑 카트를 말한다. 스마트 카트의 궁극적 목적은 쇼핑 중 사용자의 편의를 제공하는 데 있다.

본 프로젝트 ‘User Tracking Ai Cart’는 사용자의 뒷모습을 인식해 자동으로 따라다니며 음성으로 물품의 위치를 알려주는 기능을 갖춘 카트이다. 본 프로젝트는 다음과 같은 기능을 가진다.

1) 사용자 tracking

사용자의 뒷모습을 인식하고 카트와 사용자의 거리에 따라 속도를 조절하며 사용자를 따라간다. 거리가 멀면 속도가 빨라지고, 일정 거리를 유지하면 속도가 유지되고, 거리가 가까워지면 속도가 줄어들고, 거리가 너무 가까워지면 후진을 하는 기능을 추가하여 마트를 방문한 고객들의 편의성을 높이기 위해 노력하였다.

2) 음성인식

사용자가 물품의 위치를 물어보면 음성으로 물품의 위치를 알려주는 기능을 추가하였다. 우리 카트의 이름은 ‘Pory’로 정했기 때문에 ‘Pory’라고 부르면 대답하는 것을 디폴트 값으로 넣었다.

3) 물품 detection

카트 위에 카메라를 달아 카트에 담기는 물품을 인식하고 가격을 측정하는 기능을 구현하였다. 각 상품의 이름과 가격 정보 측정 시 10% 세일하는 특정 상품의 경우는 \*0.9를 하여 계산하는 방식을 사용하고 모든 값을 더하여 총 금액이 출력되도록 하여 사용자 편의성을 높였다.

* 1. **추진 배경 및 현황**

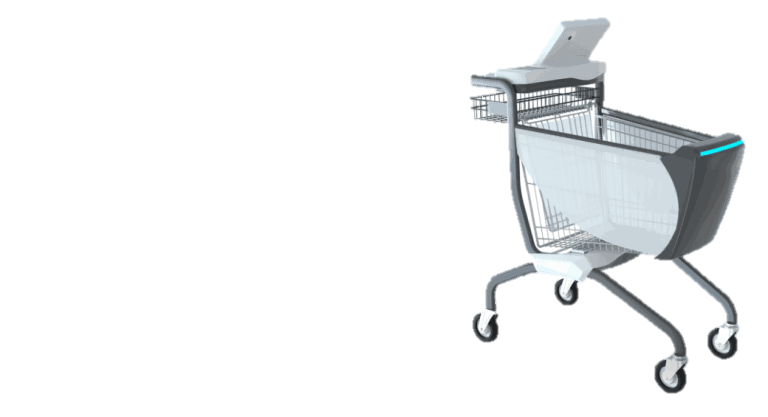
1.2.1. 추진 배경

코로나 19의 영향과 최저 임금의 인상으로 무인 판매 시장이 확대되고 있다. 이에 따라 시장 수요에 대응하여 유연성, 경제성을 지닌 자율주행 카트로 관련 시장을 선점하려는 목표를 수립하였다.

현재 정용진 이마트 부회장이 자율주행 카트 일라이를 선보이며 자사에 스마트 쇼핑 도입을 예고한지 2년이 지났으나 2021년 6월 기준, 실제 상용화된 스마트 쇼핑은 단 한 건도 없다는 사실을 알 수 있었다. 이에 따라 저희는 AI 스마트 카트의 필요성을 느끼고 카트를 제작하게 되었다. AI 카트는 매장에서 고객을 트래킹하며 상품을 자동으로 인식하고 상품위치를 알려주는 시스템을 가진다. 자율 주행 카트에 상품을 적재하고 카트에 음성인식 기능을 추가하였으며 사용자를 트래킹하고 후진 기능을 넣었다.

1.2.2. 관련 기술 현황

스테이플러, 문구, 옅은이(가) 표시된 사진

자동 생성된 설명  텍스트, 바닥, 실내, 시장이(가) 표시된 사진

자동 생성된 설명

**그림1. Eli 그림2. Kroger 그림3. Dash Cart**

AI 카트의 대표적인 예로 이마트의 일라이, 아마존고의 크로거, 아마존의 대쉬카트가 있다. 이 세 가지의 공통점은 사용자 트래킹, 상품 인식 및 자동 계산이다. 여기서 일라이는 3D 카메라와 라이더를 사용하여 장애물을 감지하고 회피하는 기능을 가진다. 크로거는 카트 내부에 딥러닝 센서를 부착하여 제품을 2차 확인하고, 대쉬 카트는 QR 시스템으로 회원 인증을 하는 차별적 기능이 있다.

1. 이마트의 ‘일라이’

앞 뒤의 3D 카메라와 360도 라이더 센서를 장착하여 장애물을 감지, 회피하며, 수집된 정보를 통해 사용자 속도에 맞춰 일정 거리를 두고 트래킹이 가능하다. 음성 인식 기능을 이용하여 매장 내 상품 위치를 검색 할 수 있으며, 더 나아가 카트가 고객을 안내 할 수도 있다. 직접 결제 기능도 있어 계산대에 갈 필요도 없다. 바코드 인식 센서와 무게 감지 센서를 카트 몸체에 장착하여 상품을 고른 즉시 바코드를 읽은 후 추후 합계 금액을 결제한다. 자율 복귀 기능을 가지고 있어 카트 반납도 자동으로 해결된다. 쇼핑이 끝나면 충전을 위해 스스로 복귀한다.

1. Kroger와 Caper의 ‘KroGo’

카트 전면에 대형 터치스크린을 장착해 고객이 플랫폼 내 쇼핑 리스트를 추천받고 프로모션 정보를 제공받는다. 길찾기 기능도 지원된다. 또한 물건을 고를 때 내장된 저울과 컴퓨터 스캐너가 있는 카메라를 이용해 상품을 스캔하고 무게를 재고 합계 금액을 계산할 수도 있다. 쇼핑이 끝나면 카트의 카드리더기에서 직접 결제할 수 있고 마일리지 적립도 가능하다. 매장에 카메라나 센서 같은 별도의 장치를 설치 할 필요가 없어 모든 오프라인 매장에 적용 가능하다는 장점이 있다.

1. Amazon의 ‘Dash Cart’

카메라, 무게 감지기, 컴퓨터, 터치 스크린이 카트에 장착되어 있어서 소비자가 어떤 물건을 몇 개 담고, 무게는 얼마인지를 자동으로 감지하고 마트를 빠져나갈 때 아마존 계좌로 자동 청구된다. 영수증은 이메일로 전송된다. 만약 중간에 카트에서 상품을 빼면 자동으로 결제 금액에서 빠진다. 아마존의 인공지능 비서 ‘알렉사’와 인공지능 스피커인 ‘에코쇼’ 또한 같이 사용이 가능하다.

1. **구현 과정**

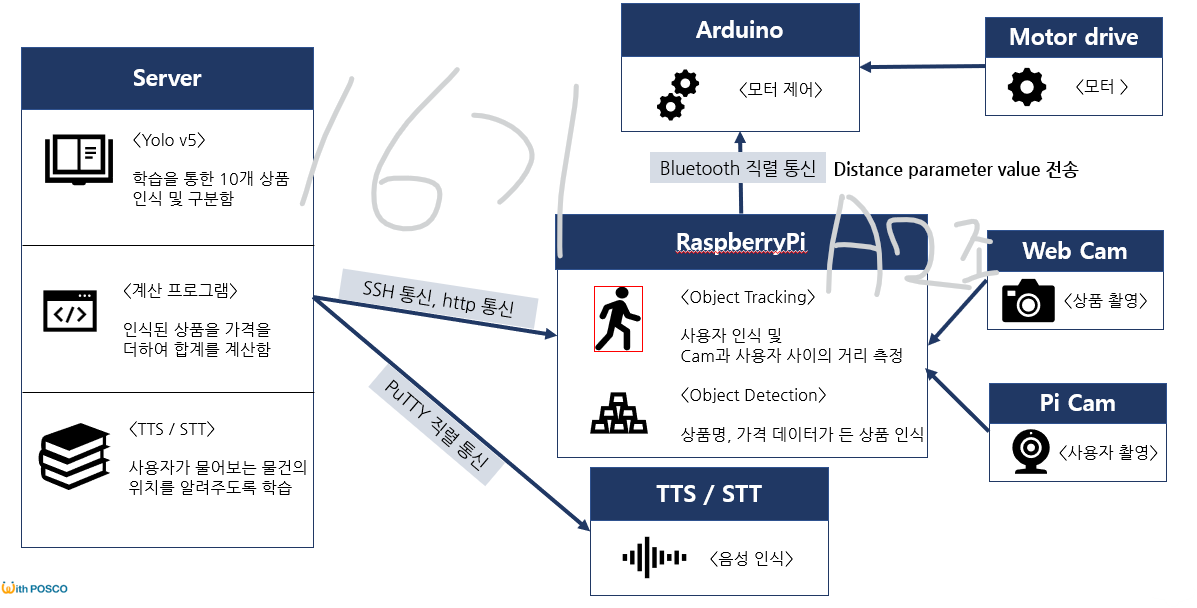
**2.1 전체 구조도 **

그림 4. 구조도

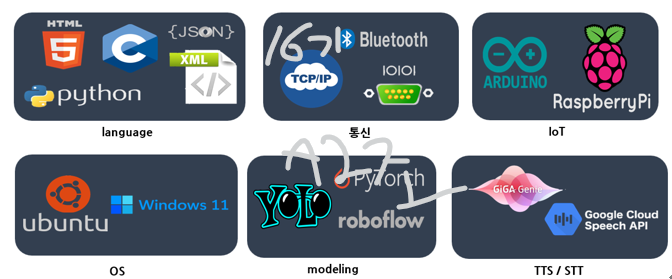


그림 5. 프로젝트 진행 시 사용한 언어 및 기술(tool)

AI 모델링을 통해 사용자를 추적하고 상품을 인식할 수 있도록 했다. 그리고 사용자와의 거리를 측정하여 그 값을 5단계로 나누어 arduino에 넘겨주었다. 거리 값을 넘겨받은 arduino는 사용자와의 거리에 따라 모터를 제어한다. 이 과정을 통해 카트 구동과 카트의 사용자 트래킹을 구현했다. 이 방법에 대한 자세한 내용은 뒤에 자세히 설명했다.

서버와 음성 인식 기능은 PuTTY 직렬 통신으로 연결하고, raspberry와는 SSH 통신, http통신을 사용했습니다. 그리고 raspberry와 arduino 사이는 bluetooth 직렬 통신으로 연결했다.

사실 전기수 코드는 버전이 너무 낮거나 코드에 중간중간 빠진 문장이 있어 전혀 사용하지 못하고 프로젝트를 진행하며 모든 코드는 우리 조가 직접 새로 짰다. 또한 카트 외관에도 빠진 부품이 많았다. 우리는 카트를 새로 제작했는데 완제품으로 한 것이 아니라 따로 되어 있는 부품을 공부해서 각자 구동해보고 또 통신을 어떻게 할 것인지 공부하며 진행했다. 그래서 우리들은 이 프로젝트를 진행하며 문제 해결력을 키울 수 있었다.

**2.2. 하드웨어**

2.2.1. 카트 도안

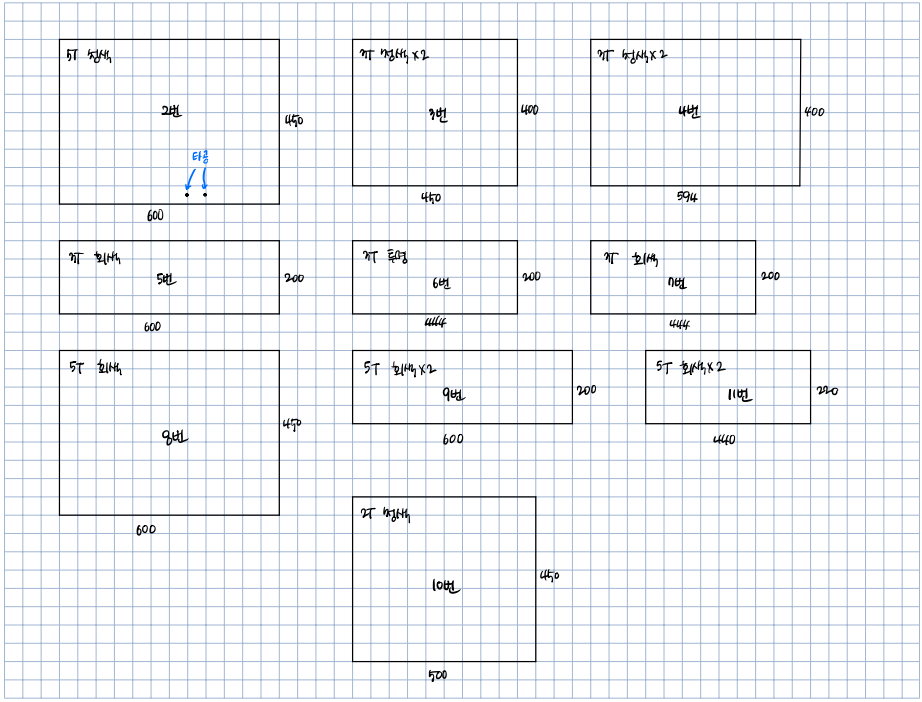


그림 6. 카트 도안

2.2.2. 제작 과정

그림7. 카트 제작 과정

(1)카트 베이스 제작

아크릴 2T, 3T, 5T를 섞어 제작하였다. 기존 8조와 14조에서 사용했던 카트가 많이 상했기 때문에 새로운 재료로 카트를 새로 만들기로 결정하였다. 3D프린터로 제작하려 시도하였으나 우리가 원하는 크기의 카트를 만들기 위해서는 너무 많은 시간이 소모될것으로 예상되어 아크릴로 선택하였다. 모터를 넣을 부분과 카메라를 넣을 부분이 필요했기 때문에 물건을 넣을 부분과 따로 분리하여 도안을 만들었다. 5T의 아크릴로 본체를 제작하려 하였으나 모터가 버틸 수 있는 무게를 초과할 수 있다 생각되어 하중을 버텨야 하는 부분만 5T로 제작하였고 나머지 부분은 얇은 아크릴로 제작하였다. 제일 문제는 아크릴 본드를 구하는 것이였다. 관련 법 땜문에 일반인에게 판매가 중지 되어 판매하는 곳을 찾는 것이 힘들었다. 다행히도 판매하는 곳을 찾아 대량으로 구매해 놓았다.

(2)모터, 모터 드라이버

카트의 무게를 버티기 위해서는 힘이 좋은 모터가 필요하였다. 하지만 속도를 포기하고 싶지는 않았기 때문에 속도와 힘의 균형이 좋은 DC 웜기어드모터를 구매하였다. 총 6개가 필요하였으나 국내 재고의 부족으로 5개를 새로 구매한 뒤에 1개는 기존의 카트에서 가져왔다. 만약 다음 기수가 이 프로젝트를 하게 된다면 주요 물품부터 재빠르게 구매하는 것을 추천한다. 모터 드라이버는 최대로 출력이 안 될 가능성을 가지고 조금 더 가격이 나가는 것을 선택하였다. 최대로 전압, 전류를 공급할 수 있는지 스펙을 확인하고 구매하였다.

(3) 바퀴

여러가지의 바퀴 중 이전 기수가 선택했던 바퀴를 구매하였다. 그 이유는 황동허브의 재고 때문이다. 구매하려고 하였던 바퀴는 키즈카 바퀴로 사이즈가 크고 견고한 것이였다. 하지만 외경이 맞는 황동허브가 국내에 존재하지 않았기 때문에 어쩔 수 없이 기존의 바퀴를 선택하였다.

(4) 부품 고정

알루미늄 프로파일과 다수의 볼트, 나사를 사용하였다. 외관을 아크릴로 선택하였기 때문에 나사로 뚫어 고정 할 수 없을 것이라 생각하여 가장 밑의 바닥은 나무로 구매하였다. 그 나무에 알루미늄 프로파일을 사이즈에 맞게 구매한 뒤에 조립하였다. 드라이버를 이용해 돌리고 전동드라이버를 사용해 구멍을 내여 고정하였다. 한가지 아쉬웠던 점은 나무가 전동 드릴을 버티지 못해 나사를 박는 과정에서 약간의 쪼개짐이 존재했다. 나무의 종류를 잘 고른 뒤에 구매하는 것을 추천한다.

2.2.3. 결과

****

**2.3. 음성인식**

음성 인식 기능은 Google API를 이용하여 구현했다. 사용자의 음성을 인식한 챗봇이 이를 텍스트로 변환하면 해당 질문에 대한 답변을 생성한다. 그 후 챗봇이 답변을 음성 파일로 변환하여 스피커로 출력하는데 이를 구현하기 위해 호출어 인식 기능, 물건 위치 소개 기능을 사용했다. Google TTS 모듈인 gtts와 python STT 모듈인 플레이사운드를 활용하여 여러 번의 테스트 후 음성인식 기능을 구현했다. 그러나 playsound 실행 시 매개변수 오류 문제가 생겨 Google Speech API와 음성을 출력해주는 파이오디오를 활용하여 머신러닝 기술을 이용하여 음성을 분석하고 텍스트로 변환했다.

2.3.1. 챗봇 기능 및 기술 소개

2.3.1.1 챗봇 기능 소개

본 프로젝트에서 마트 ai 챗봇은 고객이 특정 물건의 위치를 물어보면 적절한 답변을 해주는 기능을 구현한다.

사용자의 음성을 인식한 챗봇이 이를 텍스트로 변환하면 챗봇은 해당 질문에 대한 답변을 생성한다. 그 후 챗봇이 답변을 음성 파일로 변환하여 스피커로 출력하면 사용자가 들을 수 있게 된다. 다음은 이를 구현하기 위한 챗봇의 기능을 정리한 것이다.

(1) 호출어 인식 기능

사용자가 원하는 명령을 하려면 일단은 무언가를 불러야 ai 챗봇이 사용자의 명령을 들을 준비를 한다. 따라서 본 프로젝트에서는 사용자가 ‘청년 마트’, ‘안녕’이라는 정해진 호출어를 발화했을 때 ai 챗봇이 이를 인식하고 물건 위치 소개를 시작한다.

(2) 물건 위치 소개 기능

사용자가 호출어를 발화한 후 특정 물건의 위치를 물어보는 문장을 발화했을 때 ai 챗봇이 이를 인식하고 물건의 위치를 음성으로 안내해주는 기능이다.

2.3.1.2 음성 인식 기술

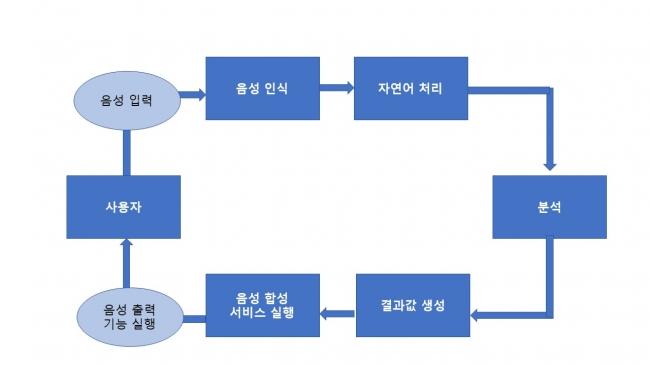


그림 8. 음성인식 구조도

2.3.1.2. STT와 TTS

STT (Speech to Text)

음성인식이란 사람이 말하는 음성언어를 컴퓨터가 해석해 그 내용을 문자 데이터로 전환하는 처리를 말한다. 본 프로젝트에서는 발화 내용을 ai 챗봇이 입력받아 텍스트로 변환할 때 사용한다.

TTS (Text to Speech)

음성합성이란 텍스트를 사람의 음성으로 변환하여 읽어주는 처리를 말한다. 본 프로젝트에서는 챗봇이 물품 위치에 대한 질문을 인식하면 그에 맞는 텍스트 답변을 음성으로 변환하여 출력할 때 사용한다.

2.3.2. 진행 과정

1) 구글 tts모듈과 playsound

구글 tts 모듈인 gtts와 python stt 모듈인 playsound를 활용하여 음성인식 기능을 구현하고자 하였다.

이를 통해 기능구현에 성공하였고 수차례 테스트를 진행했다. 그러나 playsound를 실행할 때 ‘지정한 매개 변수를 드라이버가 인식할 수 없습니다’라는 오류로 인하여 음성출력이 되지 않는 상황이 자주 발생하였다. 구글 검색을 통해 해결방법을 찾아보았고 경로지정 등 다양한 방안을 실행해 보았으나 오류가 해결되지 않았다. 따라서 다른 방안을 모색하였다.

2) Google Speech API와 pyaudio

Google Cloud Platform에서 제공하는 Speech API와 pyaudio를 활용하여 음성인식 기능을 구현하고자 하였다. Google Speech API는 구글의 머신 러닝 기술을 이용하여 음성을 분석하고 텍스트로 변환해주는 기술이다.

사용방법은 다음과 같다.

(1) Google Cloud Platform에 가입한다.

(2) 새 프로젝트를 생성한다.

(3) 프로젝트에서 Cloud Speech API를 검색한 후 사용설정을 불러온다.

(4) 사용자 인증 정보 만들기에서 서비스 계정을 클릭한다.

(5) 계정 이름을 입력한 후, 계정 권한을 소유자로 변경한다.

(6) key 만들기를 클릭한 후, Json 형태로 만들어 컴퓨터에 저장한다.

주의사항은 다음과 같다.

(1) 최초 가입시 카드 정보를 입력하지만 일단 무료로 사용이 가능하다. 하지만 용량을 너무 많이 사용할 경우 유료 전환이 될 수도 있다.

(2) 발급받은 Json형식의 key는 절대 경로를 설정해야 한다.

이때 ‘pip install pyaudio’로는 pyaudio가 설치되지 않아 구글에서 해결 방법을 검색하였다.

해외 사이트에서 해결 방법을 찾아 설치를 완료하였으며 우리 프로젝트에 맞게 코드를 수정하여 실행하였다. 그 결과 STT와 TTS기능이 구현되는 것을 확인하였으나 음성이 끊기는 현상이 발생하여 대안을 모색하였다.

2.3.3. 결과

최종적으로 6기 A반 1조에서 사용했던 KT ai makers kit를 활용하여 음성인식 기능을 구현하였다. 우선 기존의 KT API 코드 파일로는 호출어를 우리가 원하는 ‘Pory’로 변경할 수 없었다. 또한 6기 A반 1조와는 달리 우리 조는 물건의 위치를 알려주는 챗봇이 필요했기 때문에 이에 맞게 새로 코드를 작성하였다.

이를 통해 구현한 구체적인 음성인식 기능은 다음과 같다.

1. 사용자가 ‘Pory’, ‘안녕’ 두가지 호출어를 통해 챗봇을 호출한다.

2. 챗봇이 사용자에게 궁금한 사항이 있나요? 라는 음성을 출력한다.

3. 사용자가 챗봇에게 물건의 위치를 물어본다.

4. 챗봇이 사용자에게 물건의 위치를 알려준다.

이때, 사용자가 말하지 않거나 물건 위치가 아닌 다른 질문을 하는 경우 ‘다시 말씀해주세요’라는 음성을 출력한다.

5. 사용자가 챗봇에게 ‘고마워’라고 말하면 ‘물건 많이 사주세요’라는 음성을 출력함과 동시에 챗봇이 종료된다.

**2.4 사용자 Tracking**

2.4.1. 기술 및 아이디어 소개

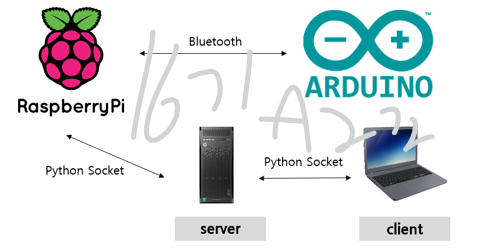
사용자 트래킹에는 2가지를 썼다. 이미지 데이터를 모아 batch size와 epoch을 적절히 조절하여 최적의 정확도를 목표로 학습했고, 그 결과 정확도가 92~100%가 나올 수 있었다. 이를 통해 먼저 사람을 학습해 추적하고, 거리 계산 알고리즘을 이용해서 arduino로 넘겨주었다. 이 때 거리 계산 알고리즘은 다음 방법으로 했다. 1m 거리에서의 사용자의 너비와 높이를 측정한 뒤, 전방에 추적된 사용자 좌표를 입력 받아 cam 사이의 거리를 구했다. 이 때, 너비의 오차율은 사용자 너비에서 높이를 나눈 값으로 구할 수 있다. 너비의 오차율과 높이를 비교하여 더 작은 값을 좌표로 적절히 나누고 반올림해 거리를 구했다. 이 거리를 5개 구간으로 나누고 우측 코드와 같이 raspberrypi에서 직렬 통신을 활용해 arduino로 송신했다. 

그림 9. 연결 방법

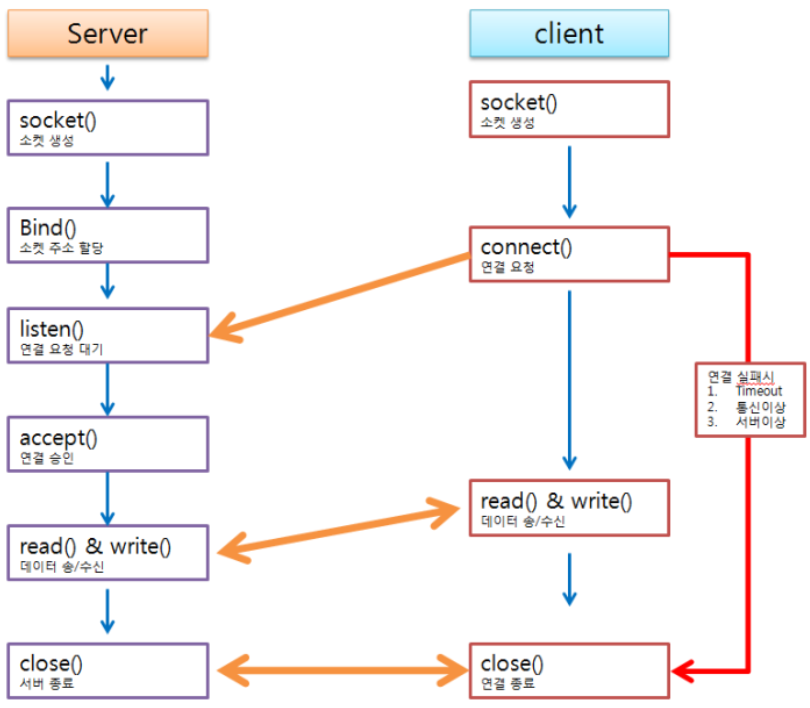


그림 10. Socket 통신

양방향 통신을 위해 socket 통신을 진행했다. 라즈베리 파이의 경우 와이파이와 socket 통신을 지원한다. 이때, 라즈베리 파이에서 워크스테이션 서버로 실시간 영상을 전송하고, 서버에서는 실시간 영상을 통해 카트에 담긴 물품의 정보를 파악해야 한다.

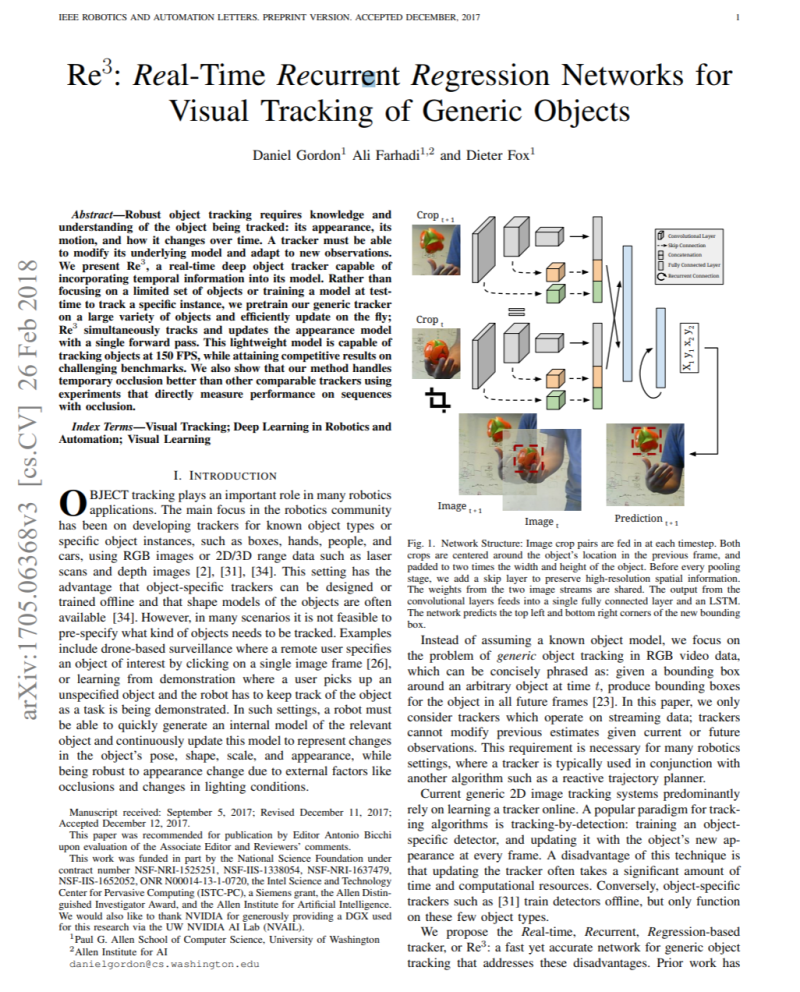
서버와 raspberrypi를 접속시켜 socket 통신을 진행했다.



그림 11. 채팅, 파일, 영상 통신 구현

(1) Re3

 2017년에 소개된 Object Tracking Model.



앞선 선배 기수(8기)의 프로젝트 스터디를 통해 Re3를 알게되었다. 속도가 빠른 편이기 때문에 다른 Tracker에 비해 실시간에 유리하고 모양과 움직임의 변화를 기억하고 학습하여 다른 Tracker에 비해 간섭이 강인한 편이라고 나와있다. Re3를 Tracker로 두고 새로운 모델을 Verifier로 두어, 실시간 Tracking은 Re3로 유지를 하되, 일정 시간마다 Verifier가 어떤 사람이 Tracking 대상인지 확인하고 정정하는 것을 먼저 그대로 따라 구현해보기로 했다.

(참고 문헌:<https://rse-lab.cs.washington.edu/papers/re3-ral.pdf>)

하지만 opencv와 tensorflow 환경 차이로 구현하는데 어려움이 많았다.

특히 tensorflow의 경우 현재 버전보다 downgrade하여 2 미만(1.x)버전이 요구되었고 pycharm을 이용하여 환경 구축을 시도했지만 설치되지 않는 오류가 계속 발생하였다. 또한 가상환경을 만들어 시도해보았만 똑같은 오류가 계속 발생하여 Re3를 대체할 수 있는 Detecting 기술이 요구되었다.

<Re3 예제 코드\_Github>

<https://github.com/danielgordon10/re3-tensorflow/blob/master/tracker/re3_tracker.py>

(2) Azure Kinect Camera

Azure Kinect Camera는 MS사에서 만든 카메라로 컴퓨터비전과 음성 모델을 위해 만들어진 개발자 키트로, 깊이 센서, 마이크, 비디오 카메라, 방향 센서 등이 결합되어 있다. 본 프로젝트에서 8기가 사용했던 카메라를 그대로 사용해보기로 했고 Color, Depth, Infrared Camera를 테스트 해보았다.



카메라 역할 및 이미지 속 사람의 골격 좌표를 얻는데 Azure Camera를 이용하기로 계획했다.

8기와 달리 키트가 발매된 지 오래되어 예제 코드를 구하거나 환경 설정을 하는데는 크게 어려움이 있지는 않았다. 특히 Azure Kinect의 공식 SDK는 C/C++로 지원하고 있으며, Visual Studio 2017 또는 2019 버전을 사용하면 쉽게 Azure Kinect 라이브러리를 내려 받을 수 있다. 또한 opencv의 경우 과거 Windows에 설치해본 경험이 있어서 Visual Studio에 설정하는 일 역시 크게 어려움은 없었다.

 [Azure Kinect DK 설명서]

:<https://docs.microsoft.com/ko-kr/azure/kinect-dk/>

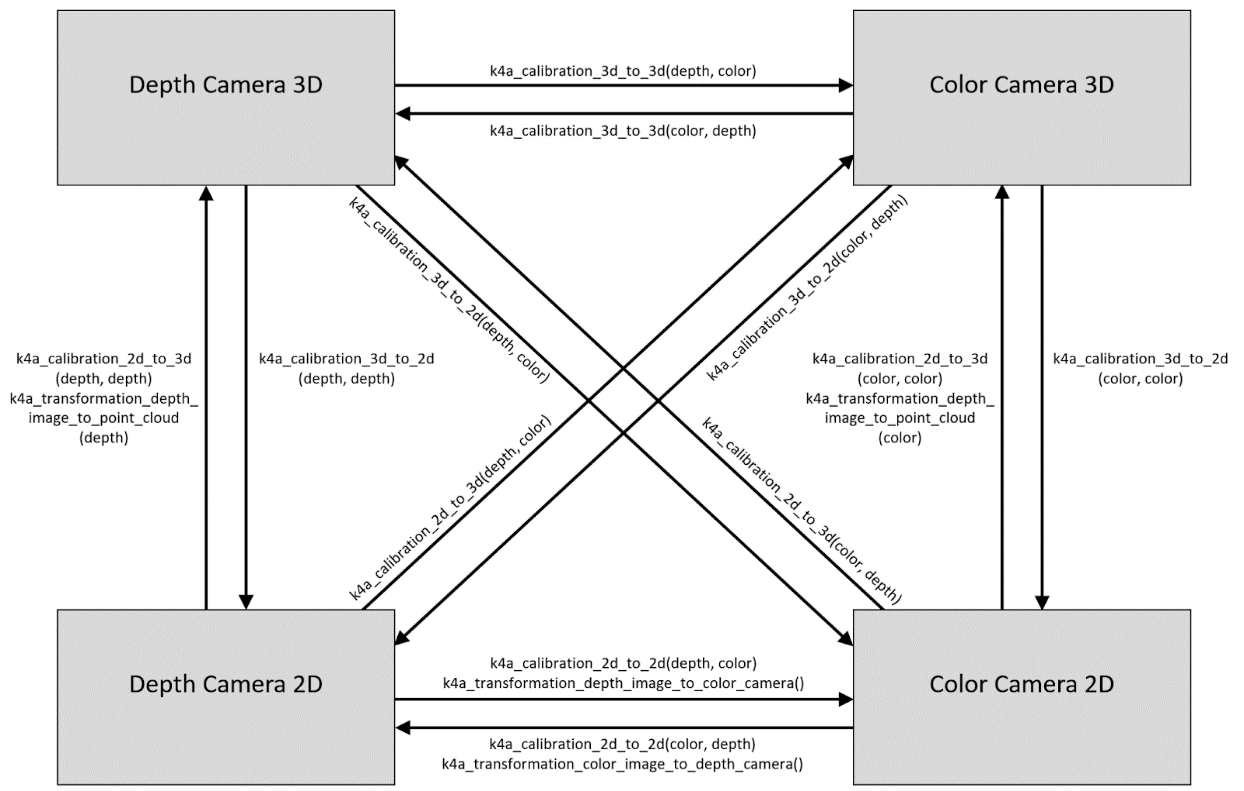
 [Azure Kinect DK 개발환경 구축하기 (Visual Studio C++) :<https://blog.naver.com/post_human/221919701554>]

Calibration 함수는 하나의 포인트가 키넥트 센서의 각 coordinate system 들로 변환할 수 있도록 해준다. 전체 이미지의 변환은 transformation 함수를 이용해서 수행하는 편이 더 효율적이다.

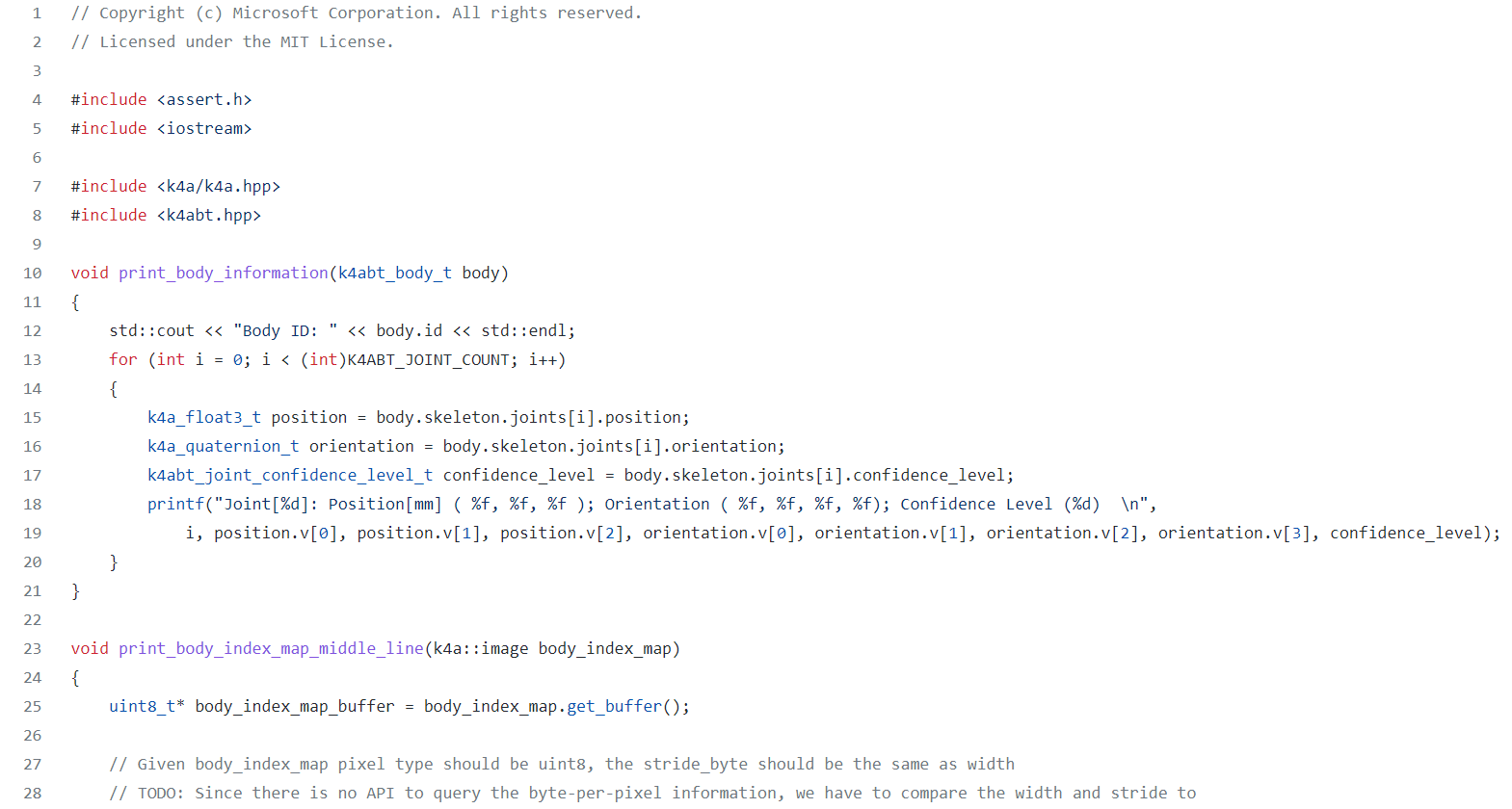
-Retrieve calibration data : 좌표계 변환을 위해서는 기기 calibration을 얻어내는 것이 필요하다. 이 calibration data는 k4a\_calibration\_t 데이터 타입에 저장되어 있으며, k4a\_device\_get\_calibration()함수로 얻을 수 있다.

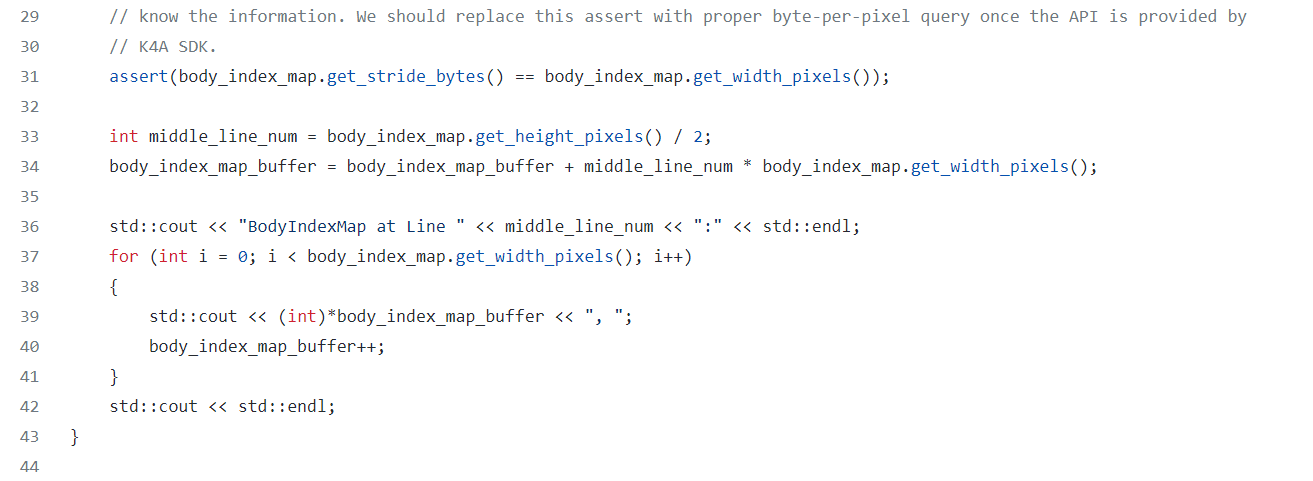
k4a\_device\_get\_calibration함수는 depth\_mode와 color\_resolution을 입력 파라미터로 가진다. 또한 opencv와도 연동 된다.

-Coordinate Transformation Functions : 하단 그림은 여러 좌표계를 보여준다.



<Azure Kinect\_Body Tracking\_Sample Code>





2.4.2 진행과정

Socket 통신을 하기 위해 8기 A반 3조의 코드를 실행시켜보았으나 구현이 되지 않았다.

인터넷 검색, 코드 분석 등을 통해 오류를 해결하고자 했으나 성공하지 못해 직접 코드를 짜 소켓통신을 시도하였다.

관련 도서와 인터넷 서칭을 통해 소켓통신에 대해 공부한 후 소켓통신 예제코드들을 분석하였다. 그리고 이를 바탕으로 window 노트북 클라이언트와 ubuntu 서버 워크스테이션 간에 텍스트를 송수신하는 코드, 이미지를 송수신하는 코드, 실시간 영상을 송수신하는 코드를 직접 짜고 구현에 성공하였다.

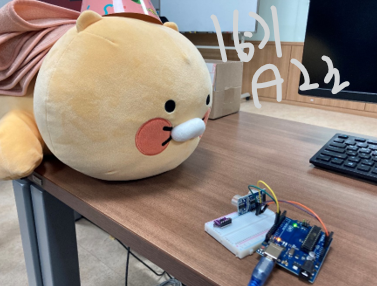


그림 12. 초음파 센서 구현

Arduino 초음파 센서를 통해 물체와의 거리가 기준점 이하로 가까이 다가오면 카트가 급제동하는 기능을 넣고자 했다. 초음파 센서 성능은 2m로 선택하였고, 시리얼 화면을 통해 초음파 센서에 잡힌 물체와의 거리를 실시간으로 확인할 수 있었다.

2.4.3 결과

****

그림 13. 사용자 추적 및 거리 측정 기능 구현

**2.5 사용자 Detecting**

2.5.1. 기술 및 아이디어 소개

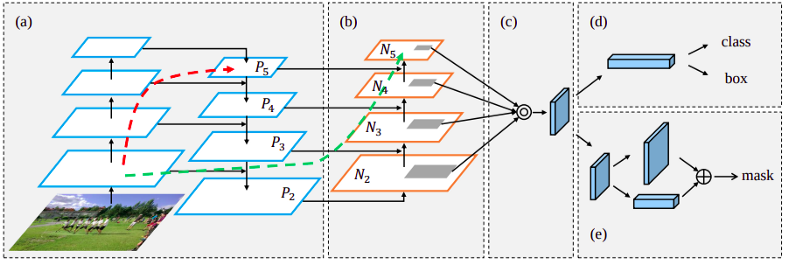
사용자 추적 학습 및 인식에 DenseNet과 Cross Stage Partial DenseNet을 활용했다.



Yolo v5는 CPU에서 빠른 속도를 제공한다. CSPNet은 입력 기능을 동일한 두 부분으로 나눈다. 하나는 전환 블록에 저장되는 그대로 유지되고 다른 하나에는 밀집 블록과 전환 블록이 제공된다.

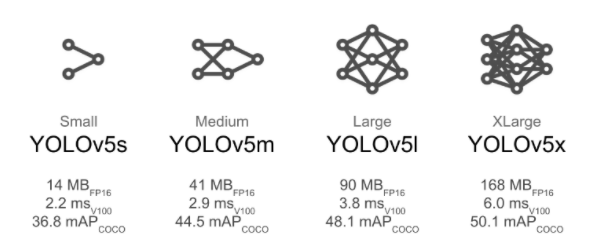
CSPNet 연결은 복잡한 모델을 줄이면서 이전 계층의 정보를 저장하는데 도움을 줄 수 있다.

네트워크가 깊을수록 더 많은 정보가 손실된다. 그래서 작은 물체를 감지하기 위해 FPN(Feature 피라미드 네트워크)을 제안했습니다. PANet(Path Aggregation Network)은 최상위 계층에서 현지화된 정보를 개선하는 FPN의 혁신이다.



(a) FPN 백본. (b) 상향식 경로 보강. (c) 적응형 기능 풀링. (d) 상자 가지. (e) 완전히 연결된 융합

초당 프레임 수가 가장 높은 s를 사용했다. 사용자 Detection의 성능을 측정할 때는 accuracy도 중요하지만 spped도 중요하다. 때문에 FPS를 고려하여 모델링하여 사용자를 인식할 수 있도록 했다.



2.5.2. 진행 과정

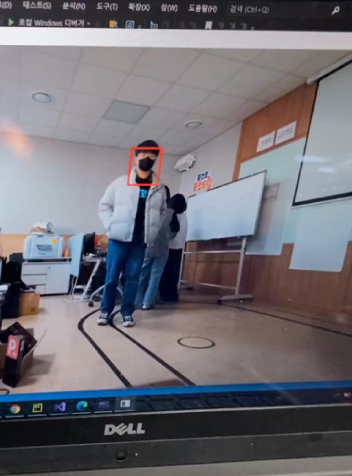


그림 14. re3를 이용한 사용자 추적

처음에는 전기수 코드를 참조했다. 8기 re3 object tracking 코드를 구현하려 했으나, tensorflow version이 너무 낮아 환경 설정이 불가했다. 그러나 그 이후로 사람의 골격 및 신체를 인식하는데 한계가 있었다. 그 다음으로는 14기 GOTURN을 시도했으나, real sense 카메라 전용 library가 필요해 보였다. 이는 Logitech 카메라로 가능해보여 재시도하여 실행시켰으나 사용자의 신체가 완전하게 인식되지는 못했다. OpenCV object tracking을 시도했고 위와 같이 face tracking은 성공했으나 body tracking을 시도하던 중 프로젝트 시간 부족의 한계를 느끼게 되었다.

2.5.3. 결과

Yolo와 ogmentation을 이용하여 학습데이터를 총 약 18,000 개 만들었다. 이 후 모델링을 통해 데이터를 학습시켰다. 학습 데이터 상품은 과자, 커피, 치약, 마스크, 주스, 등 총 10개로 구성되어 있다. 질 좋은 학습 데이터를 2만여 개 가까이 만들면서 학습 정확도를 상당히 높일 수 있었다. 이를 통해 실시간으로 카트에 떨어지며 움직이는 상품 인식이 가능해졌다.



그림 15. 상품이 실시간으로 인식되는 모습

**3. 기대효과 및 개선기회**

**3.1 기대효과**

AI 카트는 단순히 마트에서 사용하는 카트를 넘어 작업장 내에서 사용자를 추적하여 물품을 나르는 일을 할 수 있고, 충돌 방지를 위해 사용자와 일정 거리를 유지하는 기능이 있기 때문에 작업장 내 안전사고 예방에도 적용이 가능하다. 또한, 이사 시에도 활용할 수 있는 등 사용자의 목적에 따라 다양하게 사용될 수 있기 때문에 활용 방안 확장 가능성이 무궁할 것으로 보인다.

**3.2. 개선기회**

3.2.1. 하드웨어

이 프로젝트를 하면서 시간이 더 있었으면 이런 것까지 했겠다 싶은 점들이 있었다. 하드웨어 부분에서는 첫째 원래 시도했던 바퀴 6개짜리 카트를 제작이나 앞 바퀴에 서브 모터를 달아 조향 기능을 주는 것이다. 모터드라이버의 배송문제로 4개의 드라이버를 사용해야 했고 제작하는 과정에서 한 개를 못 쓰게 되었기 때문에 총 3개라는 적은 수의 모터 드라이버를 사용해야 했다. 빠르게 배송을 받았다면 더 다양한 것을 할 수 있었을 것이다. 둘째는 카트 외관의 소재이다. 3D 프린터 교육을 받은 조원이 있음에도 불구하고 우리가 원하는 사이즈의 카트를 제작하려면 너무나 많은 시간이 필요하여 포기해야 했다. 또한 아크릴이라는 소재가 구하기 쉽고 색과 두께가 다양하다는 장점이 존재하지만 무개가 전혀 가볍지 않기 때문에 제작과정에서 카트의 사이즈를 줄여야만 했다. 조금 더 가벼운 소재를 사용한다면 원하는 만큼 높이를 올릴 수 있을 것이다. 셋째로는 모터와 바퀴의 아쉬움이다. Kid’s car의 바퀴를 구매할 예정이었으나 그에 맞는 허브를 구하기 어려웠다. 다음 기수들이 만약 이 카트 프로젝트를 하게 된다면 하드웨어 재료를 미리미리 최대한 빨리 구매하는 것을 추천한다. 마지막으로 디스플레이다. 처음 우리 조는 디스플레이로 물건의 위치를 알려주는 카트를 만들려 하였다. 하지만 코로나 사태로 해외에서 받아야하는 물품들의 배송 기간이 길어졌고 디스플레이도 그 중 한 개였기 때문에 포기할 수밖에 없었다. 카트에 디스플레이로 위치를 알려주고 담은 상품들의 가격들을 미리 알려준다면 조금 더 좋은 스마트 카트가 될 것이다.

3.2.2. 소프트웨어

Kinect 카메라를 잘 사용한다면 거리 계산 알고리즘을 직접 짤 필요 없이 카메라로부터 대상 물체 사이의 거리를 구할 수 있을 것이다. Kinect 카메라로 사용자의 skeleton을 추출하는 것은 성공했으나 추출된 skeleton에서 골격 값 좌표를 받아오는 부분에 많은 시간을 쏟았다. 어깨, 골반 등 골격에 따른 좌표 값을 받아오는데 성공했다면 거리 계산 알고리즘을 짤 필요없이 거리만 바로 가져올 수 있었을 것으로 생각된다. 그리고 터치형 디스플레이를 설치하여 raspberrypi로 코드를 짜, 카트가 정지했을 때 특정 상품의 위치를 음성 인식뿐 아니라 화면에서 지도로 볼 수 있으면 사용자가 찾고자 하는 상품의 위치를 이해하는데 수월할 것으로 예상된다. 마지막으로 우리 조는 arduino에 더 익숙하여arduino와 raspberrypi 모두 이용했는데, 운영체제 설치가 가능하고 더 많은 것을 구현할 수 있는 raspberrypi에 익숙한 사람이 있다면 raspberrypi를 이용해 모터를 구동하면 좋을 것 같다는 생각을 했다. 우린 arduino와 raspberrypi 를 직렬통신 즉, 시리얼 통신을 했는데 raspberrypi 만 사용한다면 통신할 필요가 없기 때문에 제작이 좀 더 수월했을 것으로 예상된다.

**4. 조원 상호 평가**

**강지우 : 통신, 라벨링, 음성인식, 사진찍기**

|  |  |
| --- | --- |
| 김호준 | 비전공자라 교육과정 따라가는 게 힘들었는데 항상 옆에서 많이 도와주고 잘하고 있다고, 할 수 있다고 격려해줘서 고마웠어요. 형!  그리고 취직 준비 때문에 항상 잠도 거의 못 자면서 프로젝트도 열심히 참여하는 형 모습 보면서 저도 더 열심히 하게 됐어요~ 다들 집 가까우니까 서울 가서도 가끔 연락해요! 10주 동안 고생 정말 많았고 나가서도 일 잘 풀리기를 바랄 게요. |
| 노소은 | 내가 낯을 많이 가려서 사람들이랑 잘 못 어울리고 있을 때 먼저 다가와 주고 좋은 친구들도 소개해줘서 고마웠어 누나! 내 생일 챙겨준 것도 정말 고마웠고…ㅠㅠ  프로젝트 진행할 때도 조장으로써 조원들 챙기고 ppt 만들고 모델링 하고, 코딩하고…. 정말 고생 많았어. 진짜~ 우리 집도 가까우니까 과정 끝나고 같이 맛있는 거 먹으러 다니자! |
| 박나희 | 수 |
| 조현영 | 현영아 너의 친화력 덕분에 우리 조가 더 잘 뭉칠 수 있었다고 생각해. 팀 회의할 때도 적극적으로 의견 내주고, 다른 조원 일도 적극적으로 도와줘서 고마웠어!  같이 먹고 마실 때도 네 찰진 드립 덕분에 많이 웃을 수 있었어!  그리고 네가 추천해준 맛집들 다 한번씩 돌아다녀 볼게~ ㅋㅋㅋ |

**김호준 : re3 detecting, 하드웨어 제작, 모터 Arduino**

|  |  |
| --- | --- |
| 강지우 | 지우야! 비전공자로서 말은 못했지만, 많이 힘들었지? 알게 모르게 묵묵히 많이 도와줬는데 고맙다는 말을 매번 못한 거 같아서 미안해.  누나, 동생들 도와줘서 고맙고 특히 많이 부족한 형이랑 프로젝트 한다고 고생 많았어! 지우가 주는 긍정적인 에너지 정말 많이 얻어가는 거 같아서 너무 감사하고 혹시나 형이 필요하면 언제든지 연락해주면 좋겠어! 그럼 이만 줄일게. 지우야~ 정말로 고생 많았다. |
| 노소은 | 개인적으로 소은이 덕을 가장 크게 본 거 같아. 프로젝트 코딩, 발표, PPT, 보고서까지 거의 모든 부분을 책임지면서도 꼼꼼하고 완벽하게 해내는 모습을 보면서 정말 능력이 많은 친구라는 걸 느꼈어. 사실 소은이가 조장을 맡지 않았다면 우리 조의 방향이 어떻게 흘러갔을 지 감도 안 잡히는데,, 전공자로서 그리고 많은 부분에 기여하면서도 겸손하고 섬세하게 계속 팀원들 챙겨주고,, 고마운 마음이 한가득인데, 이 마음을 제대로 표현해보지 못한 게 아쉬움이 많이 남아,, 힘들 때도, 지칠 때도 옆에서 큰 힘을 주지 못해서 미안하고 그래도 다음이라는 게 있다면 그때도 조장이 되어주면 좋겠다:D 많이 투닥거리고 싸우기도 했지만,, 그래도 같은 팀에 네가 있어서 참 든든했어! 소은아 정말 고생 많았어. |
| 박나희 | 고 |
| 조현영 | 동생이지만 가끔 친구 같은? 현영아. 언제나 에너지 넘치고 적극적으로 참여해줘서 너무 고마워!!  오빠, 언니들 도와주느라 힘들었을 텐데 그리고 하드웨어 제작부터 모터가 말썽일 때마다 손까지 다쳐가며 고치느라 정말 고생 많았어!  현영아 빅데이터 발표하는 거 보면서 넌 어디서도 잘하겠다는 생각이 들었는데 조금만 더 놀고 좋은 미래들이 왔으면 좋겠다.  진짜 수고 많았고 서울에서 또 보자~ 고생했어!! |

**노소은 : 통신, yolo모델링, 사용자 detecting 코딩, 사용자 tracking 코딩, 라벨링, 사진찍기, Raspberrypi 코딩, 모터 Arduino**

|  |  |
| --- | --- |
| 강지우 | 항상 옆에서 열심히 도와준 지우야, 고마워. 비전공자지만 맡은 역할에 대해 혼자 공부하고 알아보면서 열심히 해줘서 프로젝트에 전공자 못지않게 큰 도움이 되었어. 특히 음성 인식 부분은 자연어 처리에 대한 지우의 관심과 공부가 큰 장점이 되었다고 생각해. 다른 사람과 문제가 생기면 대화하는 시간을 가지려 노력하고 고민을 들어주려 하는 모습, 맡은 역할은 꼭 책임지고 수행하는 모습 멋있었어.  그리고 중간중간 시간 날 때마다 메모하고 글 쓰는 취미 덕분에 다른 애들 자소서 첨삭도 해주고 그런 건 가봐. 그런 모습 본받고 싶어. 좋은 취미 앞으로도 쭉 이어 가길 바래. 성민이랑 나한테 새벽까지 지우 전공인 통계 과외 해줬던 것도 고마워. 나가면 네가 좋아하는 폴바셋 가자! ㅎㅎ |
| 김호준 | 정말 많이 티격태격했는데 마지막엔 누구보다 친해진 우리 조 유일한 동갑내기 호준아! 팀원들 사이 분위기 메이커(?), 접착제 같은 역할을 정말 잘 해줘서 고마워. 덕분에 우리 조가 끝까지 뭉쳐 나갈 수 있었어. 동영상 제작도 정말 수준급이더라. 내 높은 목표치에 잘해야 한다는 부담도 느꼈겠지만 포기하지 않고 항상 열심히 하고 끝까지 잘해줘서 고마워.  잠도 매일 3시간 or 그 이하로 자면서 취업 준비하고 프로젝트도 진행했는데 미래를 위해 노력하는 열정적인 모습 멋있었어. 말도 잘하고 임기응변 능력도 좋아서 어디서든 잘 할거야. 앞으로 더 잘되길 바래. 다음이라는 게 있다면 다음에도 조장할게. 대신 너도 든든하게 옆에 있어줘. 그리고 우리 반 동갑들 서울에서 모이는 거 알지? 꼭 와! 호준아 정말 고생 많았어. |
| 박나희 | 했 |
| 조현영 | 우리 조 하드웨어는 책임지고 캐리한 현영아 정말 멋지고 고마워. 기계 관련 학과가 아니라 어려움이 많았을 텐데, 관련 학과 친구들에게 열심히 물어보고 부품 알아보고 카트 도안 설계하고 호준이랑 같이 제작하고 마지막 제작 마무리까지 다 하고 정말 고생 많았어. 마지막에 손에 화상도 입었는데 카트 제작은 현영이가 정말 책임지고 맡아 한 점 고맙다는 말 꼭 하고 싶어. 언제나 똑 부러지면서 당차고 (애교 많은?) 현영이를 보면서 많은 걸 배워가. 현영이는 어디서도 사람들과 잘 어울리고 뭐든 잘 할 거야. 난 오늘부터 부지런히 맛집 리스트 추린 거 가야겠다.ㅎㅎ 우리 동네도 가까운데 어진이랑 같이 서울에서 보자! |

**조현영 : 하드웨어 설계, 부품 알아보기, 하드웨어 제작, 라벨링, 모터 Arduino**

|  |  |
| --- | --- |
| 강지우 |  |
| 김호준 |  |
| 노소은 | 어 |
| 박나희 |  |

**박나희 : 사진찍기, 전기수 컨베이어벨트 주석 달기**

|  |  |
| --- | --- |
| 강지우 | 지우야 고생했어 |
| 김호준 | 오빠 하드웨어 하느라 고생했어 |
| 노소은 | 언니 조장하느라 고생했어 |
| 조현영 | 현영아 하드웨어 하느라 고생했어 |

**Reference**

**참조문헌**

**[Arduino, raspberrypi]**

1. 이용진, “정보통신기술 교육을 위한 피지컬 컴퓨팅 학습모델”,한국 사물 인터넷 학회 논문지 Vol. 2, No. 3, pp. 1-6, 2016

**[음성인식]**

1. 강장묵 박정호, “음성인식 기반의 인공지능 서비스 연구”, 한국정보통신학회지 17권 1호

2. 김지환, “딥러닝 기반 음성인식”, 한국정보과학회지 37권 2호

3. 남기훈, “한글단어의 음성 인식 처리에 관한 연구”, 국제문화기술진흥원

4. 김영포, 이한영, “음성 인식률 개선방법에 관한 연구”

5. 이수지, 한석진, 박세원, 이경원, 이재용, “딥러닝 모형을 사용한 한국어 음성인식”, 한국통계학회, 응용통계연구 32권 2호, 2019

6. 구명완, “콘포머 기반 한국어 음성인식”, 한국음향학회지 제 40권 제5호, 2021

7. 김준우, 정호영, “제한된 학습 데이터를 사용하는 End-to-End 음성 인식 모델”, 한국 음성학회지 2020

8. 윤태성, “자연어 처리 기술을 사용한 감정분석 기법에 관한 연구”, 한국교육학술 정보원 2019

9. 박상언, “딥러닝 중심의 자연어 처리 기술 현황 분석”, 한국빅데이터 학회 한국빅데이터학회지 6권 1호 2021

10. 서혜인, 정상근, 김현지, 황태욱, 권오욱, “자연어이해 영역에서의 관계썽 학습을 통한 유사구조 문장 벡터 학습방법”, 한국정보과학회 정보과학회 컴퓨팅의 실제 논문지 제27권 제4호, 2021

**[Object detection]**

1. 이용환, 김영섭, “객체 검출을 위한 CNN과 YOLO 성능 비교 실험”, 반도체디스플레이기술학회지, 2021

2. 이진수, 이상광, 양성일, 김대욱, “딥러닝 기반 객체 인식 기술 동향”, 전자통신동향분석, 2018

3. 이진수, 이상광, 양성일, “객체 인식에서의 속도 향상을 위한 모델 앙상블”, 전자공학회논문지, 2019

4. 김휘, “적응형 셀 크기 HoG 특징 기반 객체 인식 및 추적 기법”, 과학기술연합대학원대학교, 2016

5. 강동연, “딥러닝을 이용한 객체 검출 알고리즘”, 한국정보처리학회 학술대회논문집 26권 1호, 2019

6. 명헌, “회전에 강인한 딥러닝 기반 객체 탐지 방법”, 한국과학기술원, 2020

7. 박태룡, “실시간 다중 객체인식 알고리즘 구현”, 한국전기전자학회 논문지, 2013, vol.17, no.1, pp. 51-56 (6 pages)

8. 최병관, “인공지능 객체인식에 관한 파라미터 측정 연구”, 디지털산업정보학회 논문지, 2019, vol.15, no.3, pp. 15-28 (14 pages)

9. 김동주, 임채원, “스마트폰과 아두이노를 이용한 원격제어 객체인식 이동형 홈 CCTV 개발”, 정보통신학회 논문지, 2020, vol.24, no.11, pp. 1546-1549 (4 pages)

10. 안성준, “3차원 특정점으로부터의 객체 자동인식”, 정보처리학회논문지. 소프트웨어 및 데이터 공학, 2009, vol.16, no.1, 통권 124호 pp. 47-54 (8 pages)

**[Object tracking]**

1. 이동석, 이동욱, 김수동, 김태준, “스테레오 카메라를 이용한 이동객체의 실시간 추적과 거리 측정시스템”, 방송곡학회 논문지, 2009, pp.366-377

2. 이강원, 손호웅, 지형 공간정보체계 용어사전(2016)

3. 황인택, 최광남, “색상변화를 갖는 객체추적 알고리즘”, 멀티미디어학회논문지, 2007, vol.10, no.7, pp. 827-837 (11 pages)

4. 이성호, 배승환, “온라인 다중 객체 추적을 위한 이산 푸리에 변환과 부분 최소 제곱법을 기반한 구별력 있는 외형 학습”, 한국컴퓨터정보학회논문지, 2020, vol.25, no.2, 통권 191호 pp. 49-58 (10 pages)

5. 양용준, 이상구, “OpenCV 기반 파이썬 프로그램에 의한 방송용 카메라의 객체 추적 기법”, 국제문화기술진흥원, 2018, vol.4, no.1, pp. 291-297 (7 pages)

6. 이우주, 이배호, “환경변화에 강인한 다중 객체 탐지 및 추적 시스템”, 전자공학회논문지, 2009, vol.46, no.6, pp. 88-94 (7 pages)

7. 정준용, 이규원, “지역정보와 색 정보의 단계적 적용에 의한 능동 객체 추적”, 정보처리학회논문지. 소프트웨어 및 데이터 공학, 2012, vol.19, no.2, 통권 143호 pp. 107-112 (6 pages)

8. 조지훈, 이상구, “CUDA 기반의 병렬 CAMshift 알고리즘을 이용한 객체추적 시스템”, 한국지식정보기술학회 논문지, 2013, vol.8, no.2, pp. 17-24 (8 pages)

9. 서용호, 박세준, 양태규, “사람 및 얼굴검출에 기반한 이동로봇의 강인한 객체추적 기술 구현”, 한국정보기술학회 논문지, 2010, vol.8, no.12, pp. 21-28 (8 pages)

10. 곽내정, 송특섭, “인체 비율을 이용한 인체의 조인트 자동 검출 및 객체 추적 알고리즘”, 한국 콘텐츠학회 논문지, 2011, vol.11, no.4, pp. 215-224 (10 pages)