

프로젝트 완료 보고서

- OpevCV, PyTorch를 활용한 과일 및 당도 분류 -

팀: 과즙팡팡

팀장: 전종배

팀원: 김민성, 손효정

----- 목 차 -----

1. 프로젝트 개요

- 1-1. 추진배경
- 1-2. 목적 및 목표
- 1-3. 업무분담
- 1-4. 개발일정

2. 설계

- 2-1. 개발환경
- 2-2. DB
- 2-3. 클래스 다이어그램
- 2-4. Processing model
- 2-5. 유스케이스 다이어그램

3. 구현

- 3-1. 데이터 라벨링
- 3-2. HW
- 3-3. SW

4. 결과

- 4-1. 구현결과
- 4-2. 개선사항

#. 참고자료

1. 프로젝트 개요

1-1. 추진배경



그림 1) 과일 상품 선별 작업



그림 2) 비파괴 당도 측정 작업



그림 3) 비파괴 당도 측정 과일 선별기

시중에는 다양한 과일 상품이 판매되고 있다. 각각의 과일들은 상품성 평가 기준에 맞춰서 선별 작업이 진행된다. 과일은 당도와 경도, 크기와 모양, 품질 균일성을 기준으로 삼아 과일을 선별한다.

농장에서는 자동화 과일 선별기를 구매하여 작업하는 하는 경우가 많다. 하지만 자동화 과일 선별기는 가격대가 높아 매출이익이 큰 대규모의 농장에서 사용되고 있으며, 소규모의 농장에서는 수동으로 상품을 선별하고 있다.

또, 시중에는 과일의 당도를 측정하는 장비가 있으나, 이는 수작업으로 일일이 모든 과일의 당도를 측정해야하므로 많은 시간과 인력이 필요하다. 최근에는 과일의 당도를 비파괴¹⁾로 측정하는 장비가 출시되었지만 출시시기가 그리 오래 되지 않아 아직은 가격대가 높게 측정되어 있다.

꼭 필요한 기능인 과일 분류와 당도 분류를 포함해서 소규모 농장에서도 사용이 가능한 저렴한 장비를 제작하는 것을 목적으로 이 프로젝트를 기획하게 되었다.

1) 과일의 당도를 측정하는 방식은 파괴와 비파괴로 나뉜다. 파괴는 과일에서 과즙을 짜내서 액체 내에 당을 무게로 측정하는 것이고, 비파괴는 과일에 근적외선을 쏘고 다시 반사된 빛을 측정하여 반사되어 나온 근적외선의 세기가 낮을수록 해당 과일 속에 당 성분이 많다고 예측한다.

1-2. 목적 및 목표

프로젝트에서 중요하게 생각한 키워드는 '자동화', '데이터 분류', '데이터 시각화'이다.

과일 분류 및 당도 측정을 수작업이 아닌 자동화하여, 사람이 아닌 기계로 불필요한 인력자원의 낭비를 줄이고, 보다 효율적인 작업 환경을 구축하는 것을 목표로 한다.

기본적으로 선별과 측정에 무게를 둔다. 더 나아가서는 해당 과수원에서 재배된 과일의 데이터를 모아, 향후 재배 방향에 도움을 줄 수 있는 데이터로 가공하고 시각화하여 사용자가 데이터를 관리하기 쉽도록 도와주며 과수원의 매출 성장에 기여하는 환경을 구축한다.

1-3. 업무분담

No	구분	성명	담당 업무
1	팀장	전종배	모델 학습(과일), 서버 통신 장비 설계 및 제어 프로그래밍
2	팀원	손호정	영상처리, 모델 학습(당도)
3	팀원	김민성	데이터 시각화, 디자인

1-4. 개발일정

No	내용	2020.12.08 ~ 2020.12.23											
		8일	9일	10일	11일	14일	15일	16일	17일	18일	21일	22일	23일
1	과일분류 오픈소스 수집, 알고리즘 토의	■	■	■									
2	장비 설계 토의, 과일 분류, 당도 모델 학습	■	■	■	■								
3	장비 설계			■	■	■	■						
4	제어 프로그램 개발					■	■	■	■	■	■		
5	장비 시뮬레이션								■	■	■		
6	미흡 및 보완점 분석									■	■		
7	보완									■	■	■	■
8	보고서 작성											■	■

2. 설계

2-1. 개발환경

No	부품명	모델명
1	컨베이어 벨트	서보DC모터-중형
2	TowerPro 서보모터	MG996R
3	모터드라이버 모듈	L298
4	Raspberry Pi	Raspberry Pi 4 Model B 8GB
5	카메라	RPI 8MP CAMERA BOARD
6	부저	SM-1205C
7	LCD	SZH-EK101
8	HX711 A/D 저울키트 로드셀 5Kg	Load Cell(5kg), HX711
9	브레드보드용 정전압 AC/DC가변 어댑터	EPX33DVD

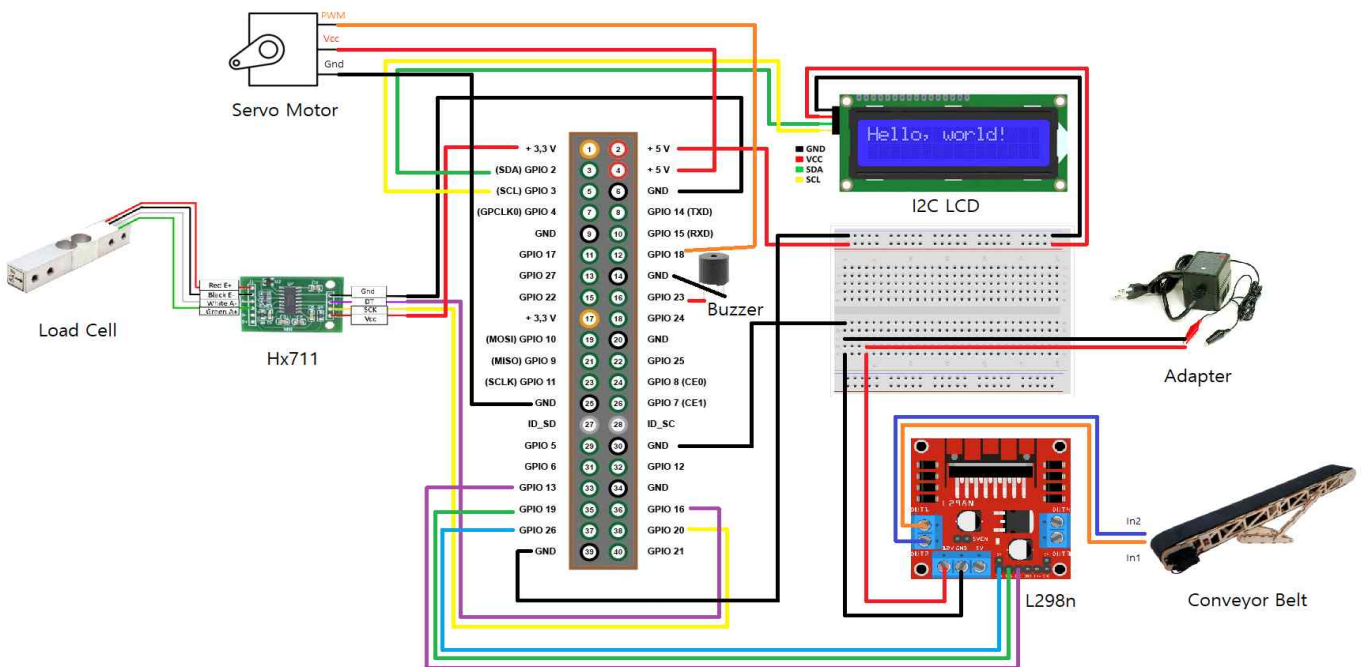


그림 4) 회로도

하드웨어 부품은 총 9가지이다. 각각의 모듈은 라즈베리파이 핀과 브레드보드를 활용하여 흰색 포맥스 판 위에 구현하였다. 핀 넘버는 위의 회로도 그림과 동일하다.

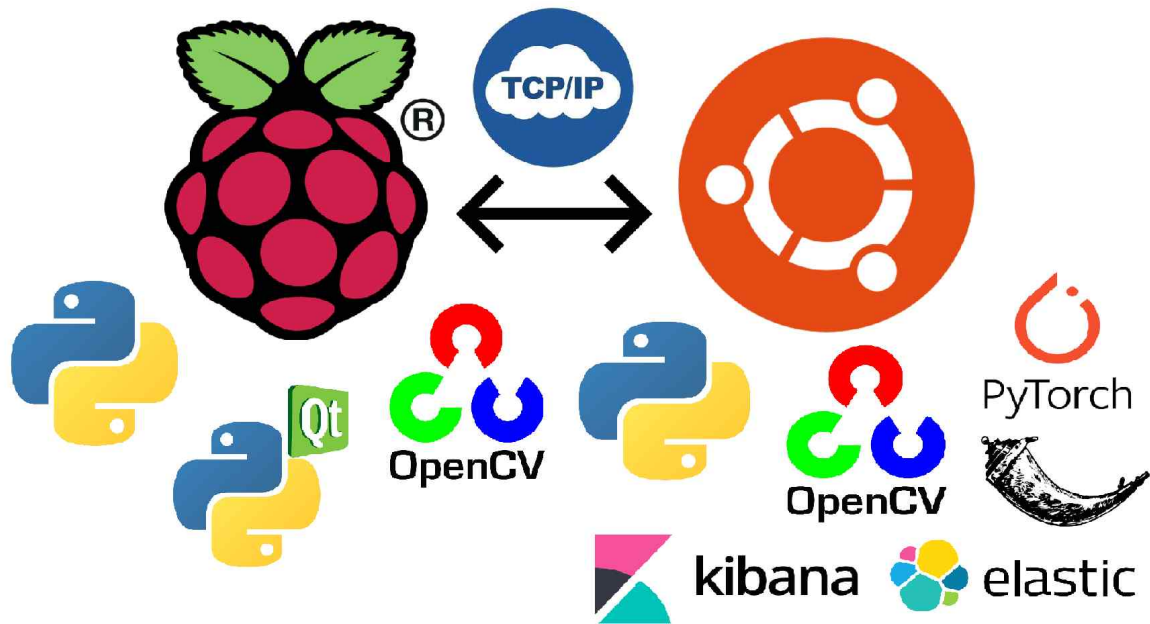


그림 5) 소프트웨어 개발환경

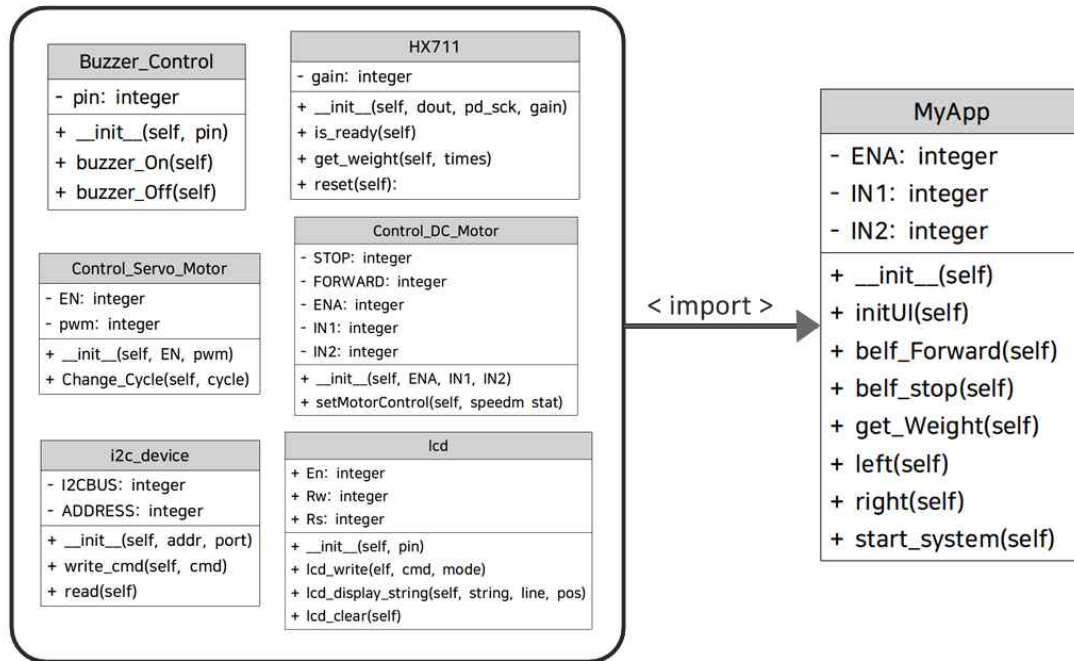
Ubuntu와 Windows 10 환경에서 개발을 진행하였다. Python에서 OpenCV를 이용해 영상처리와 Ubuntu Server에 이미지 저장을 하였고, PyTorch를 이용해 과일 학습을 진행하였다. GUI는 PyQt를 이용해 조작이 가능하도록 만들었다. Database는 Elasticsearch를 사용하였고, Ubuntu Server와 Raspberry Pi간 통신은 TCP 방식으로 하였다. Data Visualize는 Flask Web Server와 Kibana를 사용해 Elasticsearch에 저장된 데이터를 보여주도록 하였다.

2-2. DB

pyelk5
Date type : date, Fields : keyword
Name type : text, Fields : keyword
Sugar_content type:float, Fields : keyword
weight type : integer, Fields : keyword

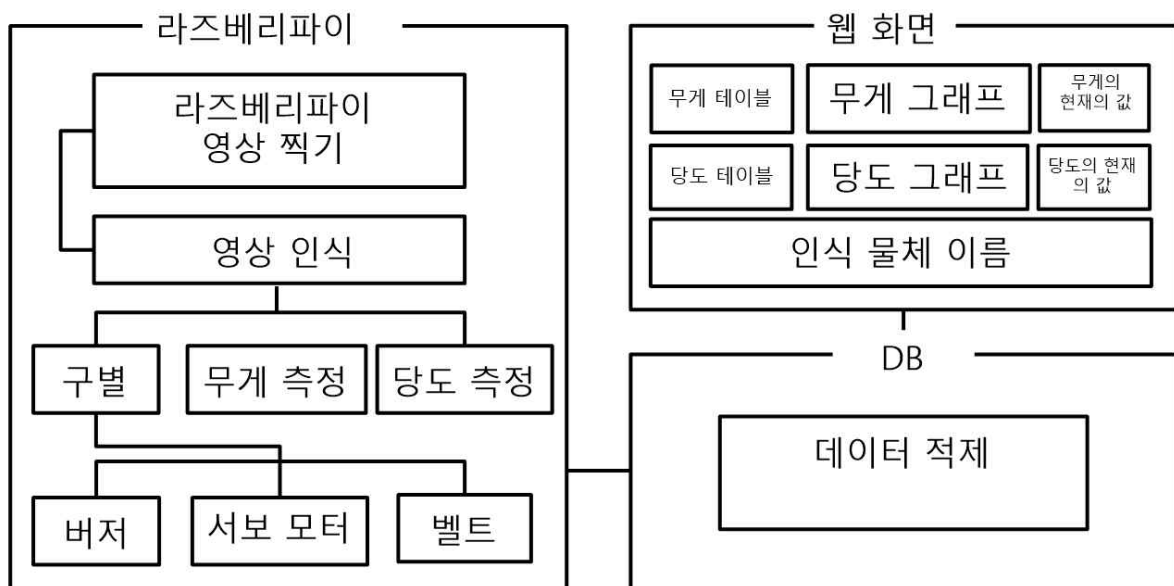
데이터베이스에는 하드웨어 센서에서 읽어온 데이터를 저장한다. 현재 날짜와 시간에 대한 튜플, 굴과 토마토를 구분하는 과일 이름 튜플, 당도 튜플, 무게 튜플로 구성된다.

2-3. 클래스 다이어그램

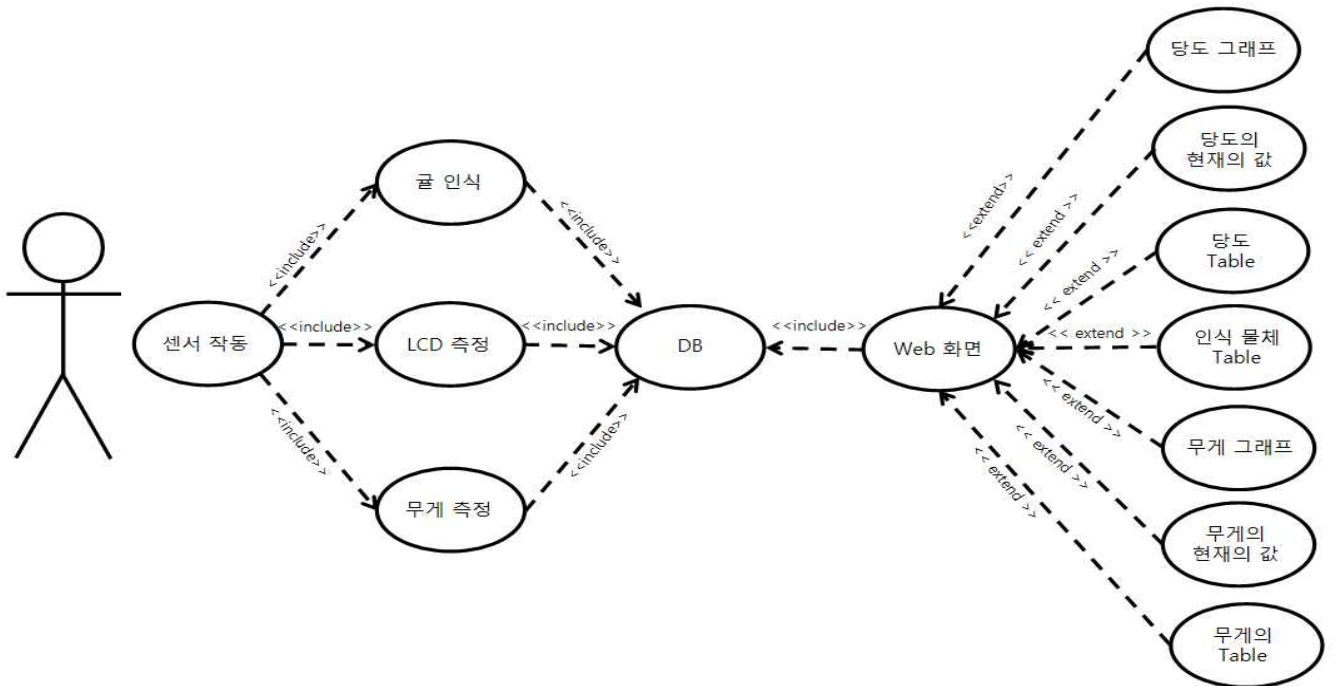


Buzzer_Control, HX711, Control_Servo_Motor, Control_DC_Motor, i2c_device, lcd 클래스는 MyApp 클래스에 import되어 사용된다. MyApp 클래스는 하드웨어를 제어하는 Pyqt로 구성되어 있다. MyApp클래스 이외에 클래스는 오픈소스를 가져와 필요한 변수와 함수를 수정하여 사용하였다.

2-4. processing model



2-5. 유스케이스 다이어그램



3. 구현

3-1. 데이터 라벨링



그림 6) 데이터 라벨링 - 과일 종류(굴, 토마토)

과일 종류 분류 기능을 구현하기 위해서 굴과 토마토 이미지를 크롤링했다. 크롤링한 이미지는 각각 Tangerine 폴더와 Tomato폴더에 나뉘어 저장하고 학습 데이터와 검증 데이터를 만들어 이미지를 학습시키는데 사용하였다.



그림 7) 데이터 라벨링 - 당도

당도 측정 기능을 구현하기 위해서 0부터 9까지 숫자값을 저장하는 폴더를 생성하였다. 각각의 폴더에는 LCD화면을 찍은 사진들이 들어있다. 0을 표시하는 LCD 이미지는 0폴더에, 9를 표시하는 LCD 이미지는 9폴더에 나누어 저장하였다. 이후 파이썬 코드를 통해 0~9폴더를 불러와 LCD이미지에서 필터 처리, 경계값 처리 등 영상처리를 하고 숫자값만 잘라내어 데이터 라벨리를 하였다. 숫자 이미지들 train데이터, test데이터로 나누어 npy파일을 생성하였고 이후 이미지를 학습시키는데 사용하였다.

3-2. HW

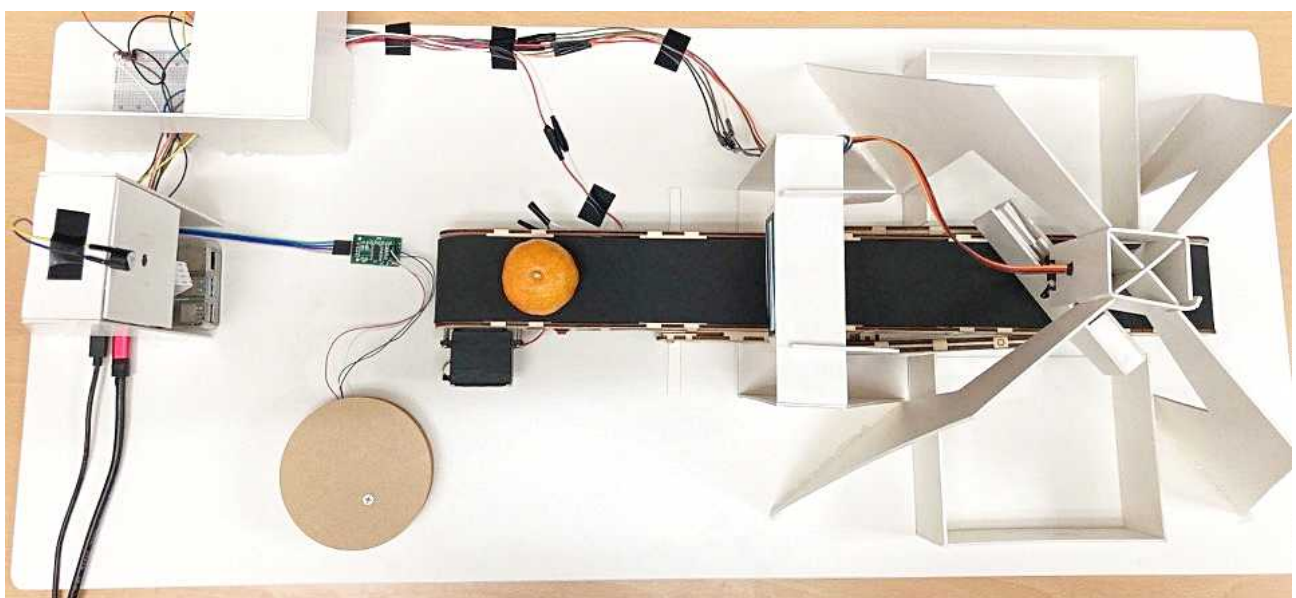
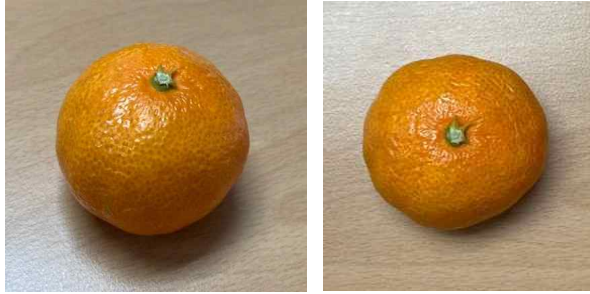


그림 4) 하드웨어 개발환경 - 전체 모형

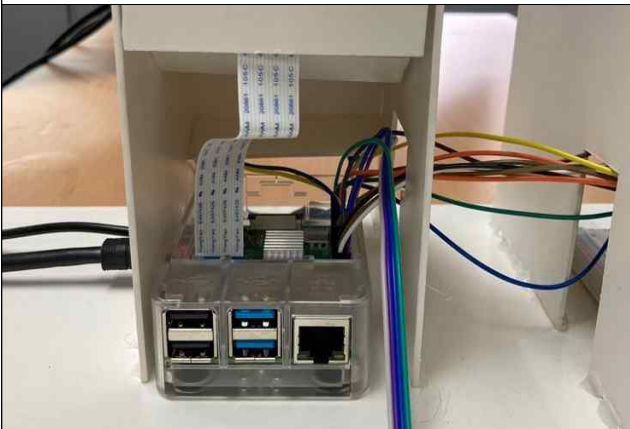
귤



토마토



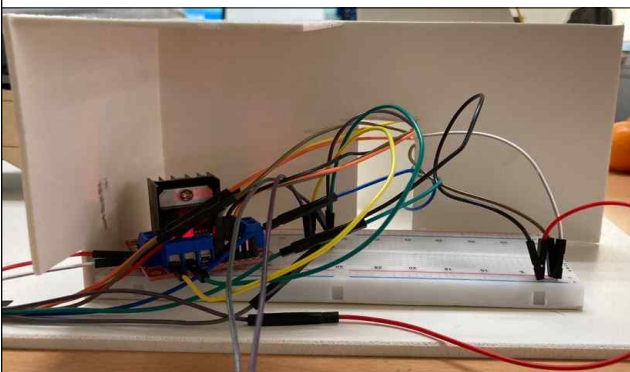
라즈베리파이



카메라



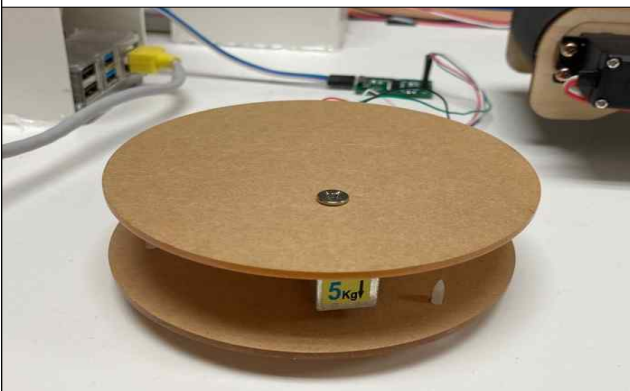
브레드보드



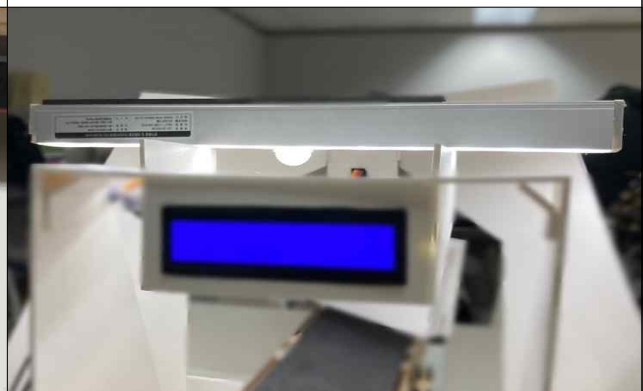
부저

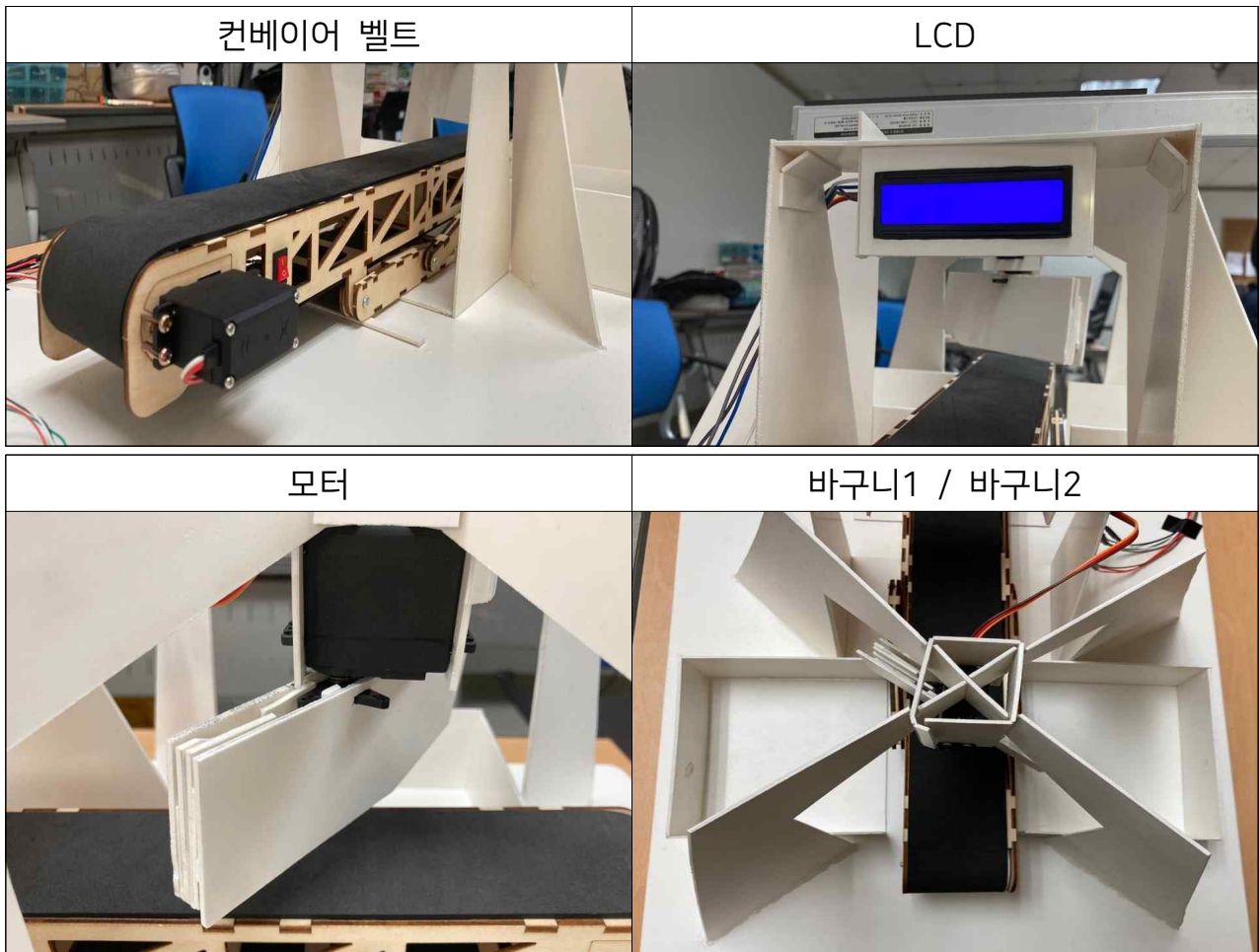


무게센서



LED





3-3. SW

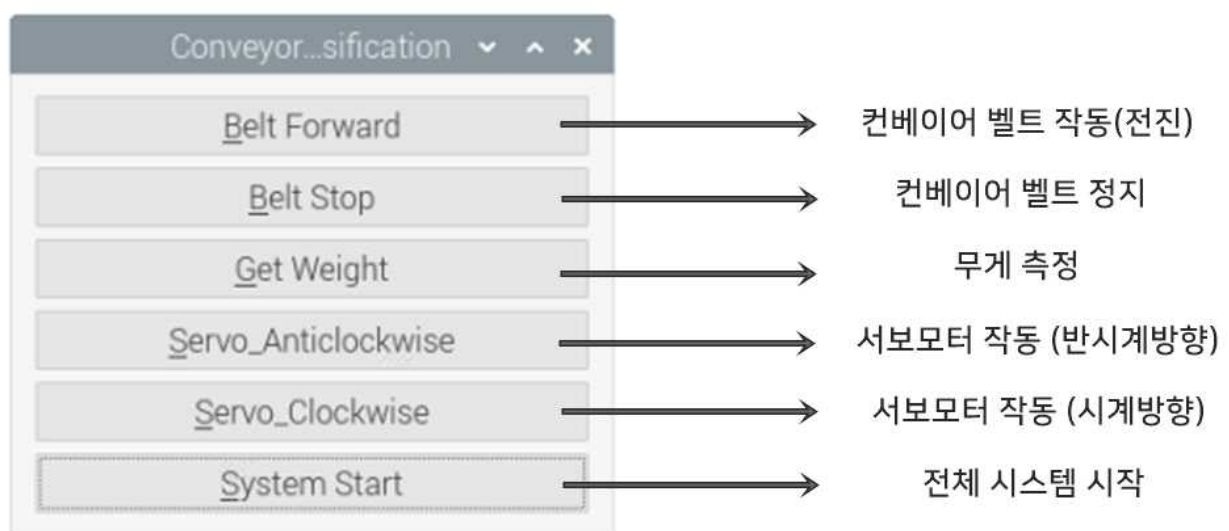


그림 8) GUI - 하드웨어 제어(무게센서, 컨베이어 벨트)

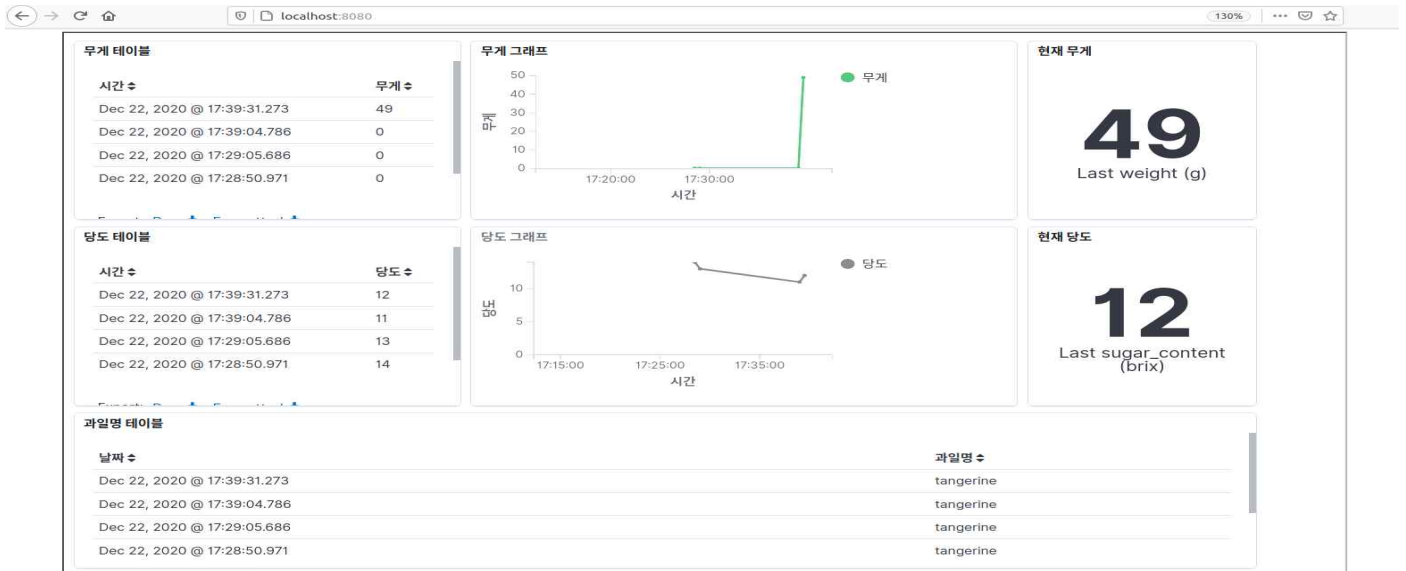


그림 9) GUI - 소프트웨어(데이터 시각화)

무게 센서와 당도 인식 기능을 통해 얻어진 데이터를 Elasticsearch 매핑을 통해 정보화하여 저장하였다. 저장된 정보를 Kibana 프로그램을 통해 시각화하였다. 테이블은 무게 테이블, 당도 테이블, 과일 종류 분류 기능을 통해 알게 된 과일명을 표시해주는 과일명 테이블이 있다. 그래프는 무게, 당도가 있다. 15분 단위로 초기화돼 표시되며 측정된 시간마다 마크되어 표시된다. 현재 무게와 현재 당도를 표시해주는 Metric도 구현하였다.

4. 결과

4-1. 결과

구현 결과 설계했던 대로 작동하였고, 측정된 값이 서버로 전달되어 데이터를 시각화하는 부분까지 잘 보여주었다. 하지만 학습 모델이 조건과 환경에 따라 정확하게 귤이랑 토마토를 구별해내지는 못하였고, LCD 화면을 이용해 당도 값을 영상처리로 가져오는 코드를 연결하려 했으나 충돌이 발생해 연결하지 못했다. 따라서 위와 같은 문제점들을 개선해야 할 필요성을 느꼈다.

4-2. 개선사항

1. 학습 데이터셋의 부족

- 학습 데이터가 크롤링한 이미지를 바탕으로 하였으나 해당 과일 하나만 있는 사진이 많지 않고, 데이터셋 조차도 깔끔하지 못해 실제 과일을 촬영해 분류해 본 결과 정확하게 분류하지 못하는 일이 많이 발생했다. 학습 모델을 바꾸거나, 데이터셋을 만들거나 하는 등의 보완점이 필요하다.

2. 카메라 성능 향상

- 이미지를 추출할 때 카메라 성능을 높은 것을 사용한다면 숫자인 영역과 숫자가 아닌 공백 영역을 잘 구분하여 보다 쉽게 이미지 처리가 가능할 것으로 보인다.

3. 조명

- 시간 경과에 따라 개발환경의 밝기가 다른데 이를 조명을 통해 일정한 환경을 유지할 수 있도록 한다.

4. 로봇팔

- 현재는 무게를 측정하고 컨베이어 벨트로 옮기는데 사람의 손을 사용한다. 이를 로봇팔로 대체하면 프로젝트 목표로 잡았던 자동화 기능이 더 강화될 것이다.

#. 참고자료

[1] 컴퓨터 비전(VISION)을 위한 전이학습(TRANSFER LEARNING)

Author: Sasank Chilamkurthy

https://tutorials.pytorch.kr/beginner/transfer_learning_tutorial.html

[2] Arduino Uno로 HX711 무게센서 앰프와 5kg급 로드셀을 사용해 무게 측정값 출력해 보기

<http://blog.daum.net/rockjy99/2434>

[3] Solution : PyTorch에서 RuntimeError: freeze_support() Error

<https://aigong.tistory.com/136>

[4] I2C LCD Library

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

[5] [모듈] L298N 모터 드라이버

<https://code-for-fun.tistory.com/27>

[6] [아두이노 강좌] 14. 아두이노로 서보모터 제어하기

<http://wiki.vctec.co.kr/opensource/arduino/servocontrol>

[7] 모두를 위한 머신러닝/딥러닝-CNN

<https://www.youtube.com/watch?v=Em63mknbtWo&feature=youtu.be>

[8] (Python) 라즈베리파이3 에서 Ubuntu로 webcam 화면 TCP socket 통신

<https://j-remind.tistory.com/58>