
20223오픈소스 개발 프로젝트

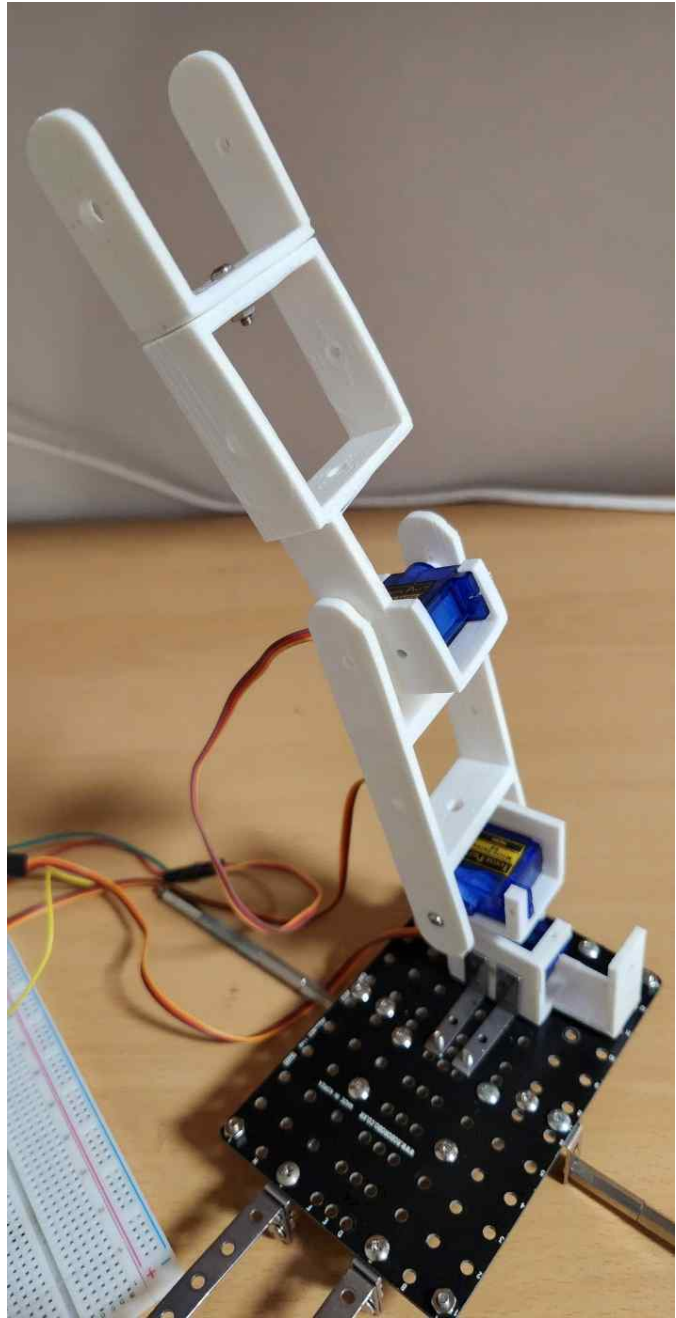
최종보고서

20223 OpenSW Development Project Final Report

과목명: 20223오픈소스개발프로젝트
담당교수: 기석철
학과: 지능로봇공학과
학년: 2학년
학번: 2021042010
이름: 박영준

목 차

1. 과제 제목 & 개요
2. 과제 필요성
3. 개발 목표
4. 프로젝트 설계
 - 하드웨어 설계
 - 개발환경
 - ROS 노드 연결 관계
 -
5. 개발 일정 & 실적
 - 개발 일정
 - 1차 중간 발표
 - 2차 중간 발표
6. 프로젝트 결과
 - 최종 결과물
7. 소감 및 후속 연구 계획
 - 소감
 - 후속 연구 계획



산업용 분류 로봇

Classification Robot

박영준

Park Youngjune

1. 과제 제목 & 과제 개요

이 프로젝트는 산업 환경에서 공장 자동화에 필수적인 분류 로봇을 구현하는 것을 목표로 한다. 분류 로봇이란, 센서를 통해서 양호한 제품과 불량한 제품을 구별하고, 둘을 분류하는 작업을 하는 로봇을 의미한다. 이 프로젝트는 카메라의 센서 정보를 바탕으로 페트병 중에서 불량한 제품을 분별하여, 3 자유도의 로봇팔을 이용하여 불량한 제품을 컨베이어벨트에서 밀어내는 작업을 수행하는 로봇을 구현할 것이다. 이 프로젝트를 진행하는 과정에서 오픈소스(ROS, OpenCV, 등등)를 최대한 활용하므로 오픈소스를 활용하는 능력을 기를 것이다. 또한 로봇팔을 3d 프린터로 제작하기 위한 도면을 제작하면서 3d 프린터를 활용하여 원하는 형태의 기계 부품을 만드는 방법을 배울 것이다.

2. 과제 필요성

무인공장은 우리나라에 필수적이다.

우리나라는 현재 인구는 감소하고 있지만, 고령자의 비중은 급속히 늘어나는 초고령사회에 들어섰다. 내국인의 생산연령인구(15~64세) 구

성비는 2020년 71.5%에서 2040년 55.6%로 감소할 것으로 보인다. 또한 미국의 계속되는 금리 인상에 의한 파장으로 물가가 치솟으면서 기업의 비용 부담이 늘어나고 소비자의 소비력이 구매력이 감소하는 가운데 이미 경제는 여러 도전에 직면해 있는 상황이다. 이러한 위기에서 우리나라의 제조업이 취할 수 있는 선택지 중에는 무인공장을 도입함으로써 인건비를 줄이는 방법이 있다. 앞으로 미래에는 지금보다 더한 경제적인 위기가 아주 많이 있을 것이다. 우리나라가 그 미래를 대비하고 피해를 최소화하기 위해서는 무인공장을 도입하는 것은 기업에 필수적이다.

저렴하게 제작 가능한 로봇팔이 필요하다.

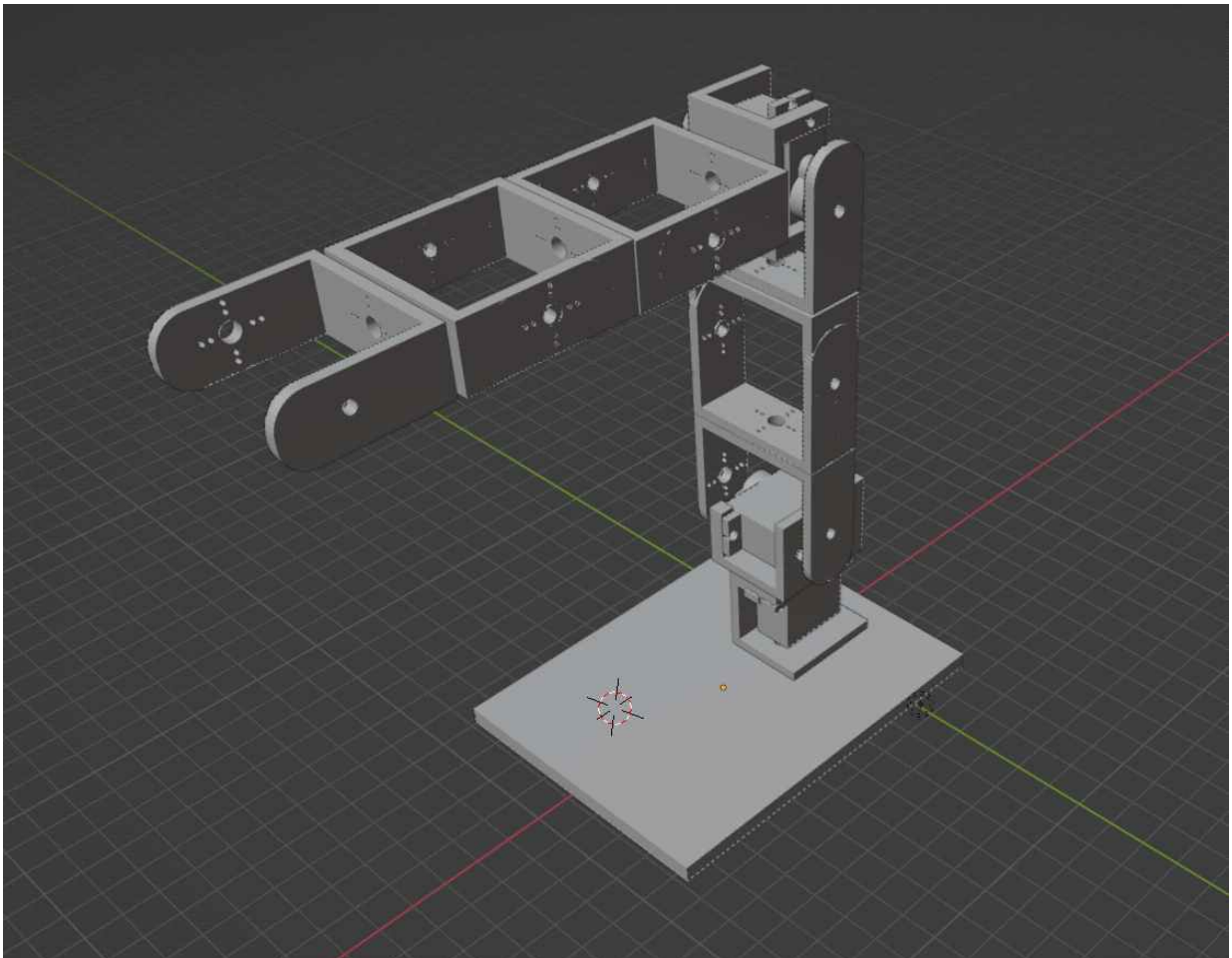
인공지능 기술의 발달로 인하여 인간의 삶의 많은 것을 자동화하는 기술이 연구되고 있다. 이로 인하여 고등 정보기술 교육의 중요성이 점점 높아지고 있다. 정보기술을 이용하여 로봇을 조작하는 실습으로 활용하기 좋은 기계에는, RC 자동차, 로봇팔, 드론 등등이 있다. 그러나 이러한 기계들은 교육에서 활용하기에는 비싼 경우가 많다. 고등 및 대학 교육에서 개인이 각각 로봇팔을 이용한 실습을 진행하기 위해서는 로봇팔의 가격이 아주 저렴해질 필요가 있다.

3. 개발 목표

-5000원 이하의 다목적 로봇팔.

-YOLO를 활용한 객체인지 실습.

-역기구학을 구현하여 로봇의 손 부분의
X, Y, Z를 입력으로 로봇팔을 제어



4. 프로젝트 설계

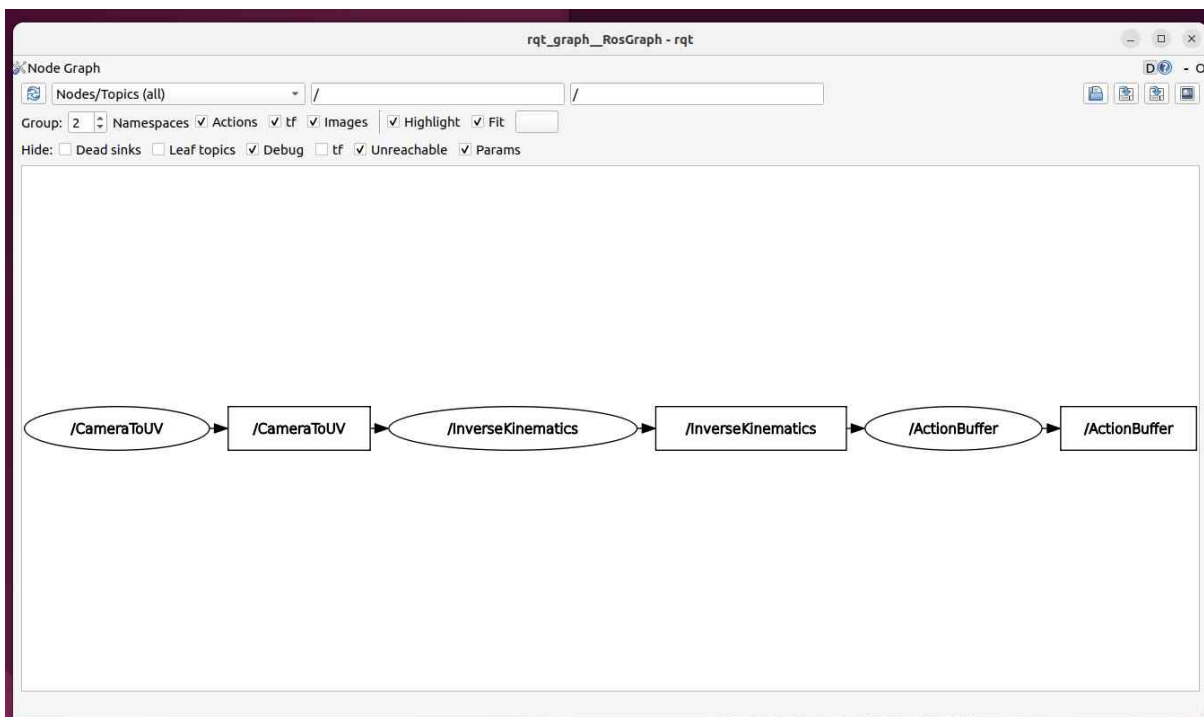
-하드웨어 설계

프로젝트에 사용할 로봇팔의 하드웨어 설계 목표는 로봇팔의 기능을 하는 가장 간단한 형태의 구조를 만드는 것과 동시에, 다목적으로 사용될 수 있도록 하는 것이었다. 이를 위해서 나는 로봇팔의 핵심적인 부품인 모터를 고정하는 브라켓을 설계의 토대로 삼았다. 브라켓과 브라켓을 연결하는 연결부위를 제외한 부분들은 최소화하였다. 로봇팔의 손 부분은 로봇팔의 목적에 따라서 집게를 장착할 수도 있고, 그저 밀쳐내는 행동만 수행할 경우는 빗자루와 같은 형태일 수도 있듯이 다양할 수 있다. 그러므로 나는 로봇의 손 부분에는 나사를 통해서 여러 가지 모듈을 장착할 수 있도록 로봇팔의 손 부분을 설계하였다.

-개발환경

내가 이 프로젝트에서 사용한 개발환경은 UBUNTU 22.04과 ROS-humble을 사용하였다. 이를 사용한 이유는, 첫 번째로 최신 버전의 Ubuntu와 ROS가 이전의 버전과 어떤 차이가 있는지 알고 싶었기 때문이다. 두 번째로는 현재 많은 ROS에 관한 자료들은 Ubuntu 16.04 또는 Ubuntu 18.04를 기준으로 삼고 있다. 하지만

-ROS 노드 연결 관계



< rqt_graph를 사용하여 시각화한 노드의 연결 관계 >

ObjectDetectionCamera.py:

rqt_graph에는 보이지 않으나, 해당 파일을 작동시켜야 로봇이 카메라로부터 정보를 받을 수 있다. 이 파일은 카메라를 통해 YOLOv3를 이용한 객체 인식을 한다. 인지한 객체가 펠트병일 경우에는 펠트병의 위치 u, v를 uv.txt에 저장한다.

CameraToUV:

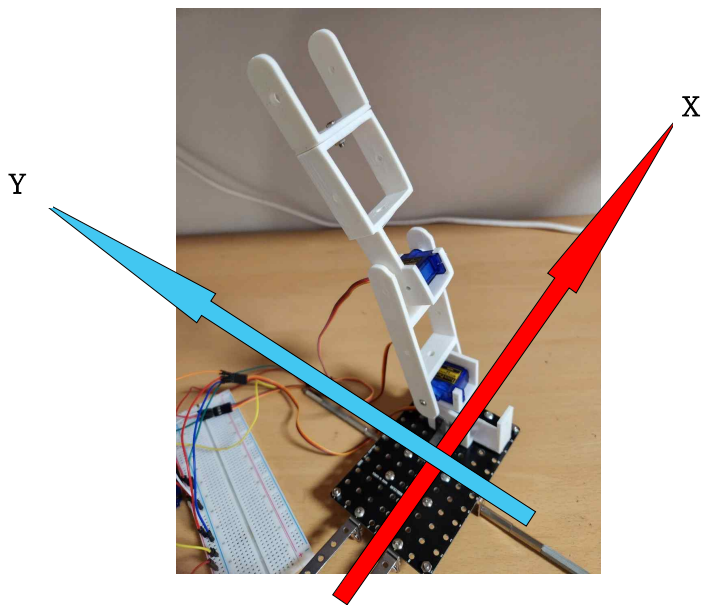
이 노드는 uv.txt로부터 페트병의 위치를 받아온다. 그리고 InverseKinematics 노드로 그 정보를 퍼블리시한다.

InverseKinematics:

이 노드는 페트병의 위치, u v를 이용해서 물체의 위치 x,y,z를 계산한다. 그리고 얻어낸 물체의 위치를 토대로 물체를 밀쳐내는 액션을 생성한다. 그 후, 액션에 해당되는 x,y,z를 역기구학을 이용해서 모터의 목표 제어각도 e1, e2, e3를 구하고 ActionBuffer에 제어각도를 퍼블리시한다.

ActionBuffer:

이 노드는 InverseKinematics로부터 얻은 액션에 대한 제어각도를 순차적으로 로봇팔을 제어하는 아두이노에 보내는 역할을 한다.



< 계산에 활용한 로봇팔의 기저를 기준으로한 좌표계 >

InverseKinematics노드에서 목표제어각도인 e1, e2, e3를 구하는 과정은 다음 수식을 이용한다.

$$\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} v_{11}v_{21}v_{31}x_{camera} \\ v_{12}v_{22}v_{32}y_{camera} \\ v_{13}v_{23}v_{33}z_{camera} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U \\ V \\ Q \\ 1 \end{bmatrix}$$

이때 X, Y는 로봇팔에 대한 물체의 상대위치를 의미한다. 그리고 v1은 카메라의 u에 해당하는 벡터, v2는 카메라의 v에 해당하는 벡터, v3는 q에 해당하는 벡터이다. U, V는 카메라에서 본 물체의 좌표를 의미한다. Q는 물체의 절대 좌표가 0이 되도록 하는 값이다.

위 방법을 통해서 물체의 로봇팔에 대한 상대위치를 알아냈다면, 이를 토대로 물체를 밀어내는 액션을 생성한다. 액션은 로봇팔의 손을 (X+3, Y, 2)→(X-3, Y, 2)의 위치로 이동하는 것이다.

액션이 정해졌으므로 다음은 액션에 해당하는 좌표에 대응하는 모터의 제어 각도를 구해야 한다.

로봇손의 어떤 좌표에 대해서 로봇팔 모터의 제어 각도를 문제를 역기구학이라고 한다. 내가 이 로봇에 적용한 수식은 다음과 같다.

$$\theta_1 = \text{atan2}(y, x)$$

$$d = \sqrt{x^2 + y^2 + z^2}$$

$$\theta_3 = \tan^{-1}\left(\sqrt{\frac{(2l_1l_2)^2}{(l_1^2 + l_2^2 - d^2)^2} - 1}\right)$$

$$\theta_2 = 90 - \tan^{-1}\left(\sqrt{\frac{(2l_1d)^2}{(l_1^2 + d^2 - l_2^2)^2} - 1}\right) - \tan^{-1}\left(\frac{z}{\sqrt{x^2 + y^2}}\right)$$

5. 개발 일정 & 실적

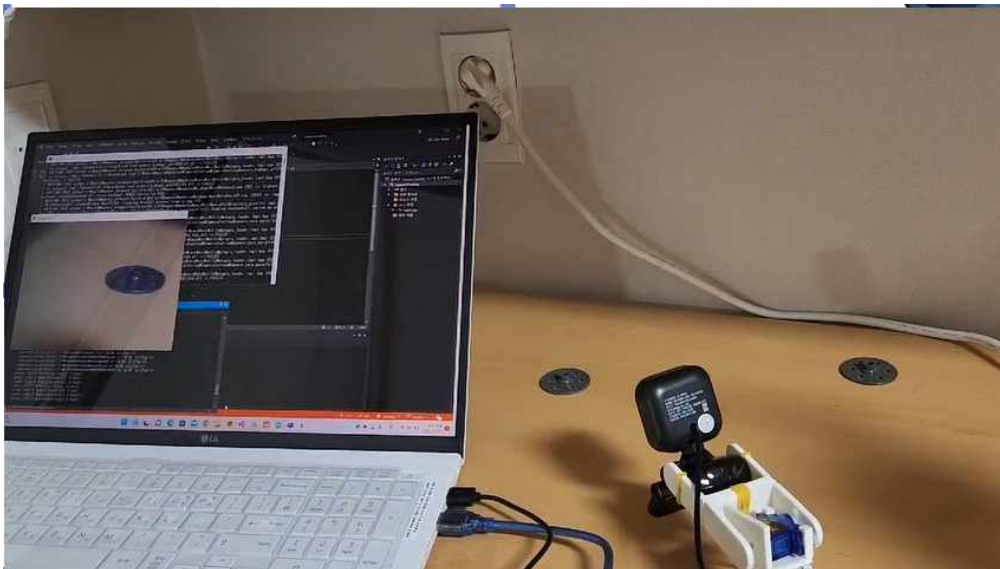
-개발 일정

1차 중간 발표까지 프로젝트에 사용할 하드웨어를 제작한다. 또한 로봇팔에 사용할 아두이노, 웹캠을 마련한다. 그리고 3d 프린터를 이용하여 로봇팔에 사용할 브라켓을 제작한다. 아두이노를 활용하여 인간이 로봇팔을 조종하는 단계까지 개발할 것이다.

2차 중간 발표까지 프로젝트의 개발환경을 구축하고, ROS를 이용하여 카메라와 로봇팔이 움직이는 것이 하나의 시스템이 될 수 있도록 개발한다. 또한 YOLOv3를 활용하기 위한 개발환경을 구축하여 객체를 인식한다.

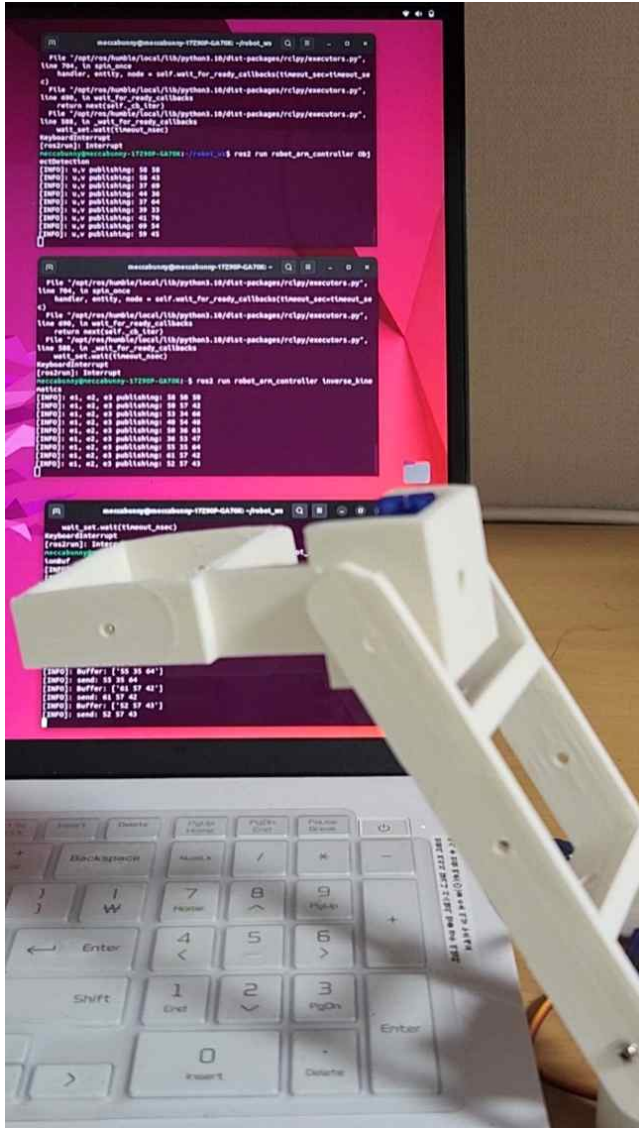
최종 결과 발표까지 카메라로 객체 인식을 하는 프로그램을 로봇팔과 연결한다. 그리고 역기구학 문제를 계산하여 로봇팔이 사용자가 원하는 동작을 취할 수 있도록 한다.

-1차 중간 발표



Cult3D 에서 SG-90 모터에 대한 브라켓의 stl 파일을 얻을 수 있었다. 나는 이 파일을 Blender를 이용하여 로봇팔에 맞게 수정하였다. 수정된 stl 파일은 충북대학교 메이커센터의 큐비콘 3d 프린터를 이용하여 출력하였다. 그리고 Window 11 환경에서 OpenCV를 이용하여 카메라의 정보를 읽는 것을 구현하였다.

-2차 중간 발표



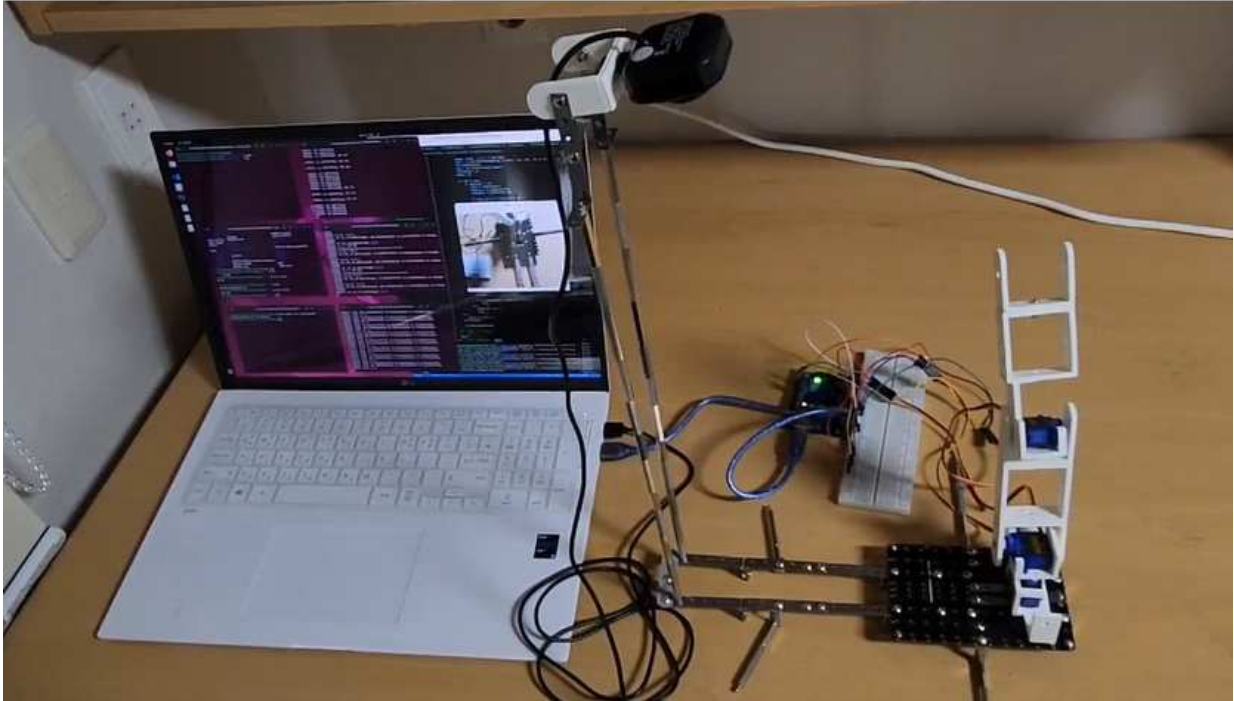
컴퓨터를 듀얼 부팅하여 Ubuntu 22.04를 설치하였다. 그리고 ROS-humble, OpenCV를 활용하기 위한 가상환경을 구축하였다. ROS 노드를 작성하여 카메라의 정보 u, v가 모터의 제어 각도로 변환되는 정보 흐름을 구현하였다.

6. 프로젝트 결과

github 링크:

<https://github.com/MeccaBunny/ClassificationRobot2022>

결과 시연 및 성능 평가



진행과정 및 문제점 해결

로봇팔의 모터를 아두이노로 제어하는 과정에서 정지해야 하는 모터가 무한히 회전하는 오류가 발생하였다. 이 오류의 원인은 한 번에 여러 개의 모터를 제어하는 경우, 모터에 가하는 전력이 부족할 때 발생하는 오류였다. 이 문제를 해결하기 위해 나는 아두이노의 모터를 제어하는 코드를 수정하여, 3개의 모터는 한 번에 하나씩만 순차적으로 움직이도록 하였다.

카메라에서 얻은 물체의 위치 정보 u , v 를 노드에 전달하는 과정에서 나는 txt 파일에 u , v 에 대한 정보를 쓰고, CameraToUV 노드에서 이 txt 파일을 읽음으로 정보를 전달하였다. 만일 이때 uv.txt 파일을 읽기 위해서는 절대 경로를 사용해야 한다. 또한 아래의 코드처럼 “~/ros/” 위치에서 uv.txt 파일을 읽을 수 있다.

```
with open("/home/meccabunny/ros/uv.txt", "r")
as f:

    u = f.readline()
    v = f.readline()
```

로봇의 기저가 무겁게 고정되어 있지 않으면, 로봇팔이 움직일 때, 기저와 기저에 고정된 카메라가 진동하는 문제가 발생하였다. 이를 해결하기 위해 나는 기저와 카메라를 고정하는 부분에 철심을 부착하여 흔들리지 않도록 고정하였다. 또한 기저와 카메라를 연결하는 부분을 3d 프린터를 사용하여 제작하는 것은 비효율적이다. 대신에 기저를 고정하기 쉽도록 설계하는 것이 바람직하다. 나는 기저를 고정하기 위해서 해당 브라켓에 얇은 구멍을 뚫어 그 부분을 철로된 브라켓과 연결하였다.

7. 소감 및 후속 연구 계획

이 프로젝트를 통해서 나는 분류 로봇의 시제품을 만들어 보았다. 처음에 프로젝트를 선정하게 된 이유는 카메라와 로봇팔이 있으면, 다양한 이미지 처리, 인공지능, 기구학 등등의 로봇과 관련된 분야를 실습해 볼 수 있기 때문이었다. 그러나 시중에 판매되는 로봇팔은 나의 기대보다 비쌌고, 또한 특수한 목적의 로봇팔들을 사용하기 위해서는 직접 로봇팔을 제작하는 방법을 알 필요가 있음을 느꼈다. 이 프로젝트에서 로봇팔은 나의 이러한 고민을 해결하기 위해 최대한 저렴하게 목표의 동작을 수행할 수 있도록 설계되었다. 이 보고서가 나와 같은 고민을 한 사람들에게 로봇팔을 제작하는 방법에 대한 가이드라인이 되었으면 좋겠다.

내가 이 프로젝트를 진행한 이유는 앞으로 로봇을 공부할 때 실습으로 사용해보기 위함이었다. 따라서 인공지능이라는 분야를 공부할 때, 카메라를 통한 인식의 결과 확인으로서 로봇에 액션을 가할 것이다. 로봇팔은 전기 신호를 통해서 물리적인 출력을 가하는 아주 기초적인 도구이다. 후속 연구에서 나는 이 프로젝트를 진행하면서 얻은 경험을 토대로 여러 개의 로봇팔을 이용하여 시각 정보를 처리하여 보행하는 보행 로봇을 만드는 것을 구현할 것이다.

참고 자료

- 과제의 필요성(초고령사회): <https://www.bbc.com/korean/news-54506949>
- SG-90 모터 브라켓 STL: <https://cults3d.com/en>
- ROS-humble: <https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>
- OpenCV: https://docs.opencv.org/4.x/d9/df8/tutorial_root.html
- <https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/>
- 표윤석, 임태훈. 『ROS 2로 시작하는 로봇 프로그래밍』. 루비페이퍼, 2021.
- John J. Craig. 『Introduction to Robotics Mechanics and control』. 3rd Edition . Pearson Prentice Hall, 2005.
- 허경용. 『아두이노 Arduino 상상을 스케치하다』. 제이펍, 2014.