

**Intel AI 프로젝트**

# **EOF:EOF**

**권강현, 박도현, 박용근, 우창민**

## [목 차]

<b>I. 서 론.....</b>	<b>3</b>
1 절 개발 개요.....	3
1. 개발 배경.....	3
2. 개발 목표.....	3
3. 작품 개요.....	3
<b>II. 본 론.....</b>	<b>4</b>
1 절 개발 환경 및 개발 도구 설명.....	4
1. 전체 개발 환경.....	4
2. 보드 및 Tools 설명.....	5
2 절 작품 구성.....	6
1. 부품 리스트.....	6
2. 회로도.....	7
3. 시스템 구성도.....	7
3 절 장치별 동작 순서도 및 UML 다이어그램.....	8
1. 전체 순서도.....	8
2. 시퀀스 다이어그램.....	8
3. 클래스 다이어그램.....	9
4 절 개발 과정 및 주요 기능.....	10
1. 계획 수립.....	10
2. 단계별 진행 과정.....	11
3. 주요 기능.....	11
5 절 구현 결과.....	12
1. 완성 사진.....	12
2. 시연 영상.....	12
<b>III. 결 론.....</b>	<b>13</b>
1. 향후 목표.....	13
2. 기대 효과.....	13
<b>IV. 참고 자료.....</b>	<b>13</b>
참고 문헌.....	13
깃허브 링크.....	13

## I. 서 론

### 1 절 개발 개요

#### 1. 개발 배경

전 세계적으로 환경오염이 매우 심각하다. 이에 따라 우리나라는 페트병과 비닐, 유리병과 철 뚜껑을 분류해서 버리는 분리배출제를 2020년부터 시행하고 있다. 실행한지 4년차가 되었지만 실상은 잘 진행되지 않고 있다. 페트병의 라벨을 분류해서 버려도 분류하지 않은 페트병과 섞이면 다시 분류해야한다. 하지만 라벨을 제거하기 불편해서 안하는 사람들이 대부분이다. 결국 분리수거 시스템의 마지막 부분에 사람이 일일이 분류하는 번거로움이 있다.



< 그림 1> 환경부의 분리배출제

#### 2. 개발 목표

##### 1) 아이디어 도출

분리수거 시스템의 마지막에 사람이 하는 일을 영상처리 딥러닝으로 대체하여 효율적인 분류를 하기로 했다.

##### 2) 필요성

재활용 쓰레기를 작업자가 일일이 손으로 분류하는 작업은 번거로운 일이다. 그래서 기계가 대신해서 깨끗한 플라스틱만 구분하면 번거로운 일을 조금 덜 수 있게 된다.

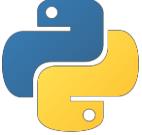
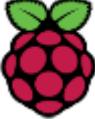
#### 3. 작품 개요

서버(데스크탑) 측에서 클라이언트(라즈베리파이)의 프로그램을 시작하면 클라이언트 백그라운드에서 프로그램이 실행된다. 클라이언트 측에서 페트병을 옮겨놓으면 컨베이어벨트를 따라 페트병이 이동하고, 일정 위치에 도달하면 웹캠에 보이게 된다. 이 웹캠 영상은 실시간으로 서버 측에서 확인할 수 있고, 이 이미지를 가지고 서버측에서 Detection 및 Classification을 실행한다. 해당 결과를 가지고 서보모터의 동작 유무를 결정하게된다. 만약 페트병에 라벨이 있으면 서보모터가 동작을 해서 페트병이 왼쪽으로 떨어지게 하고, 라벨이 없으면 그대로 통과한다. 페트병 분류가 끝난 후 노란색 버튼을 통해 음성녹음을 할 수 있는데, 모델을 바꾸고 싶으면 “모델 바꿔줘”라고 말하면 페트병 모델이 유리병 모델로 변경된다. 한번 더 말하면 다시 페트병 모델로 돌아온다. 만약 어떤 모델이 사용 가능하지 궁금하거나, 하드웨어의 기능이 궁금한 경우 물어보면 스피커를 통해 대답해준다. 서버 측에서는 클라이언트 측 프로그램 시작 및 종료, 레일 동작 및 정지, 모델 변경 등의 기능을 사용할 수 있다.

## II. 본 론

### 1 절 개발 환경 및 개발 도구 설명

#### 1. 전체 개발 환경

	Client Device	Server Main
Device		
Language		
OS		
Library	 	
	 	PyQt5
System		

<표 1> 전체 개발 환경

## 2. 보드 및 Tools 설명

### 1) Raspberry Pi 4B

Raspberry Pi 는 리눅스 운영체제 기반으로 동작하는 싱글 보드로, 다양한 개발도구와 라이브러리를 포함하여 개발 환경을 쉽게 구축할 수 있다. 컴퓨터지만 크기가 작아 휴대성이 뛰어나고 전력 소모 또한 낮은 편이다. 많은 인터페이스를 제공하여 개발, 통신, 데이터베이스 등 여러가지 용도로 사용이 가능하다.

### 2) OTX(Openvino Training eXtension)

OTX란 Openvino Training eXtension의 약자로 CVAT과 Datumaro를 거쳐 dataset이 준비되었을 때, OTX에서 제공하는 pre-trained된 model을 바탕으로 사용자의 dataset을 전이학습시켜주는 툴이다. 사용 가능한 기능으로 build, train, export, optimize, deploy 등이 있다.

### 3) PyQt5

PyQt5란 Qt5 어플리케이션 프레임워크에 대한 파이썬 버전이다. GUI(그래픽 사용자 인터페이스)를 구축하는데 중점을 둔 프레임 워크로 멀티스레딩을 위해서는 PyQt에서 제공하는 qthread를 활용해야한다. 본 시스템에서는 Python 코드를 이용하여 GUI를 구축하였으며, 라즈베리파이에서 웹캠으로 찍은 내용이 서버측 GUI에 출력되며, GUI를 통한 라즈베리파이 제어에 활용하였다.

### 4) Datumaro

Datumaro란 Dataset Management Framework로, CVAT으로 데이터셋 준비가 되었을 때 진행한다. 데이터셋 구축과 변형, 분석과 같은 기능을 지원하는 프레임워크 및 CLI이다. CLI 환경에서 CVAT을 거친 json 파일과 준비된 데이터셋을 가지고 명령어만 치면 사용자 설정 비율에 맞춰 split해준다. Detection은 train:validation:test=7:2:1의 비율로 나누고, Classification은 train:validation=8:2의 비율로 데이터셋을 나눈다.

### 5) CVAT

CVAT이란 Computer Vision Annotation Tool의 약자로 컴퓨터 비전 알고리즘용 데이터 라벨링에 사용되며 데이터셋을 모은 후 Annotation을 쉽게 할 수 있도록 도와주는 무료 오픈 소스 웹 기반 이미지 및 비디오 주석 도구이다. 마우스 드래그 앤 드롭으로 특정 영역을 박스쳐서 Detection이나 세그멘테이션 모델에 사용될 이미지 Annotation을 할 수 있다. 동영상의 경우 interpolation 기능이 있어 키 프레임에 대해서만 annotation을 진행하면 사이의 프레임들에 대해 자동 annotation을 지원한다. CVAT에서 지원하는 AI 모델을 활용해서 자동으로 Annotation을 할 수도 있다. Annotation이 완료되면 export annotations를 통해 annotation된 내용이 담긴 json파일을 원하는 foamat 형태로 얻을 수 있다.

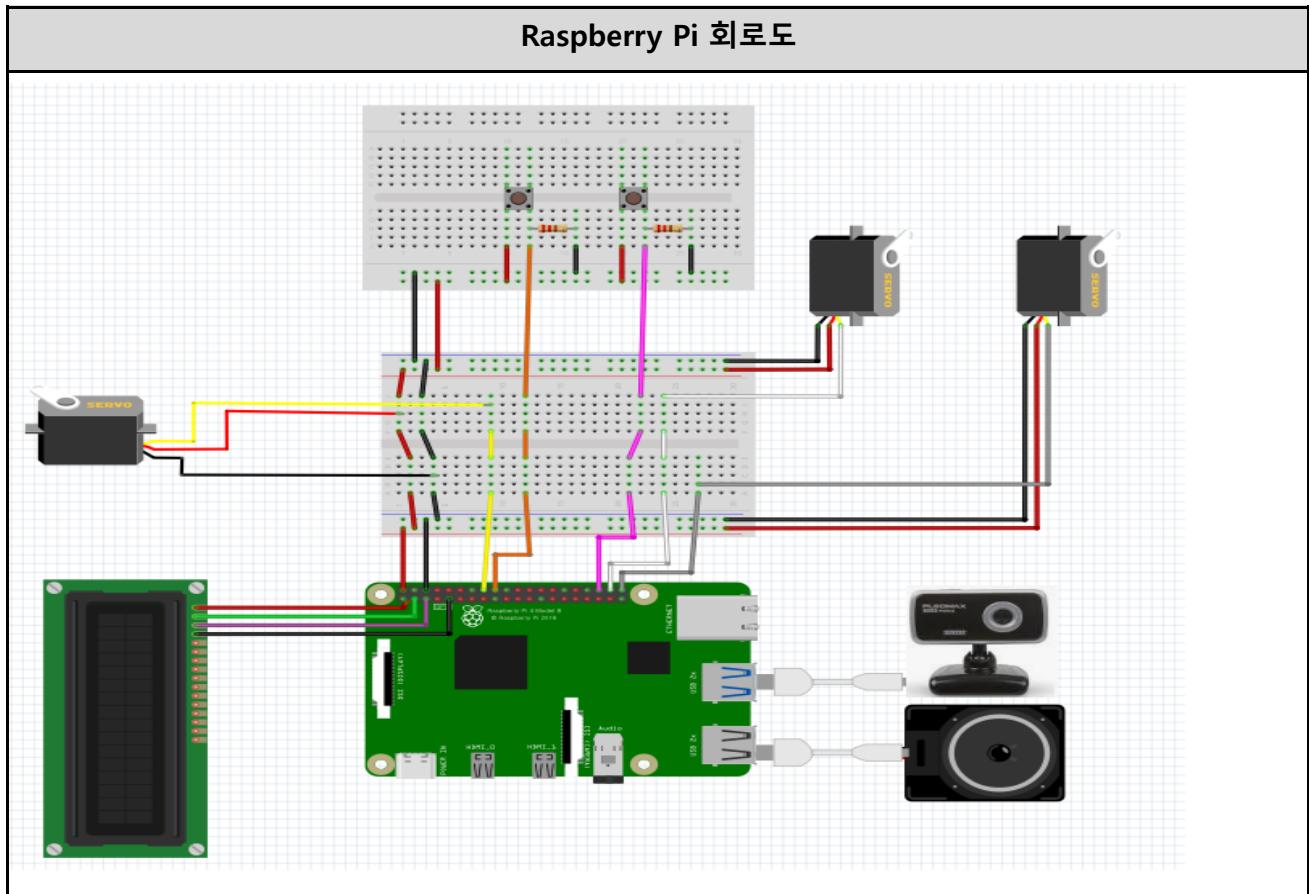
## 2 절 작품 구성

### 1. 부품 리스트

시스템	부품명	이미지	링크 (고유 번호)
영상 촬영 및 음성 녹음	PLEOMAX W-210		<a href="https://bit.ly/3uvM3lf">https://bit.ly/3uvM3lf</a> (112810)
raspberry pi 4 동작제어	Raspberry Pi 4 Model B		<a href="https://url.kr/hov8m9">https://url.kr/hov8m9</a> (12234534)
	SG90 서보 모터		<a href="https://bit.ly/46AX5mT">https://bit.ly/46AX5mT</a> (12503476)
	택트 스위치 (button)		<a href="https://bit.ly/3N51nvM">https://bit.ly/3N51nvM</a> (1322139)
	RC 서보		<a href="https://url.kr/kjetuq">https://url.kr/kjetuq</a> (1382239)
	1602 I2C 캐릭터 LCD		<a href="https://url.kr/ug2fwa">https://url.kr/ug2fwa</a> (12500006)
	Speaker		<a href="https://url.kr/gmykcw">https://url.kr/gmykcw</a>
	power supply		<a href="https://url.kr/s8qah3">https://url.kr/s8qah3</a>

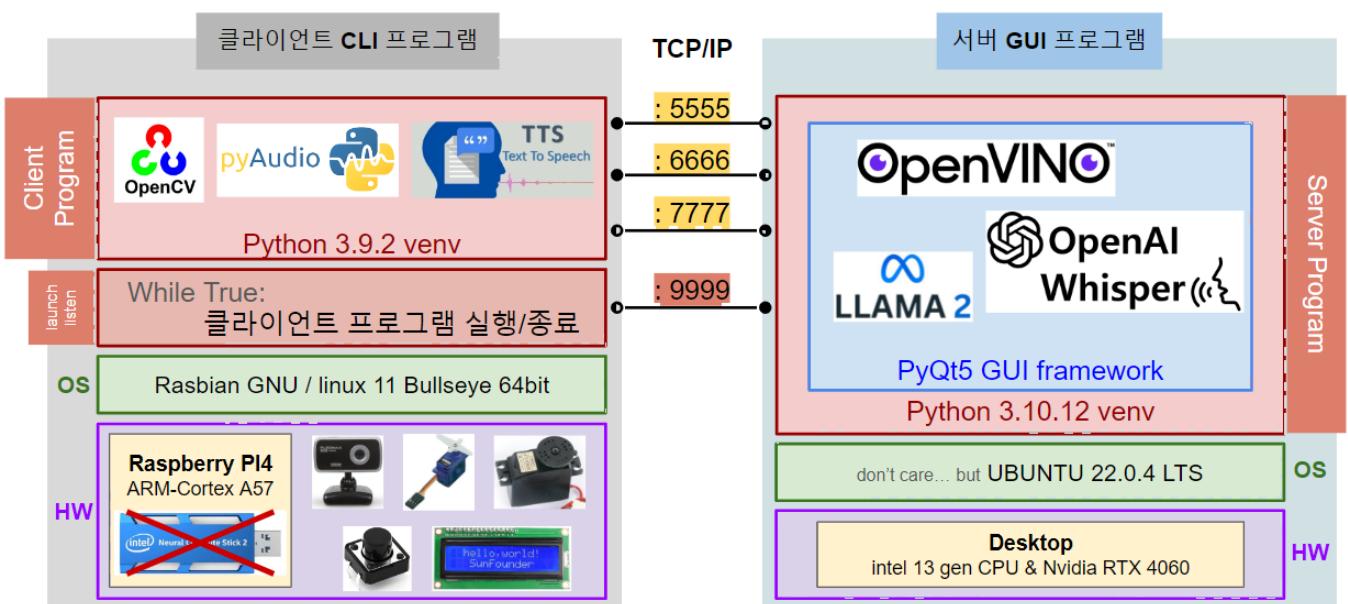
<표 2> 부품 리스트

## 2. 회로도



<표 3> 하드웨어 회로도

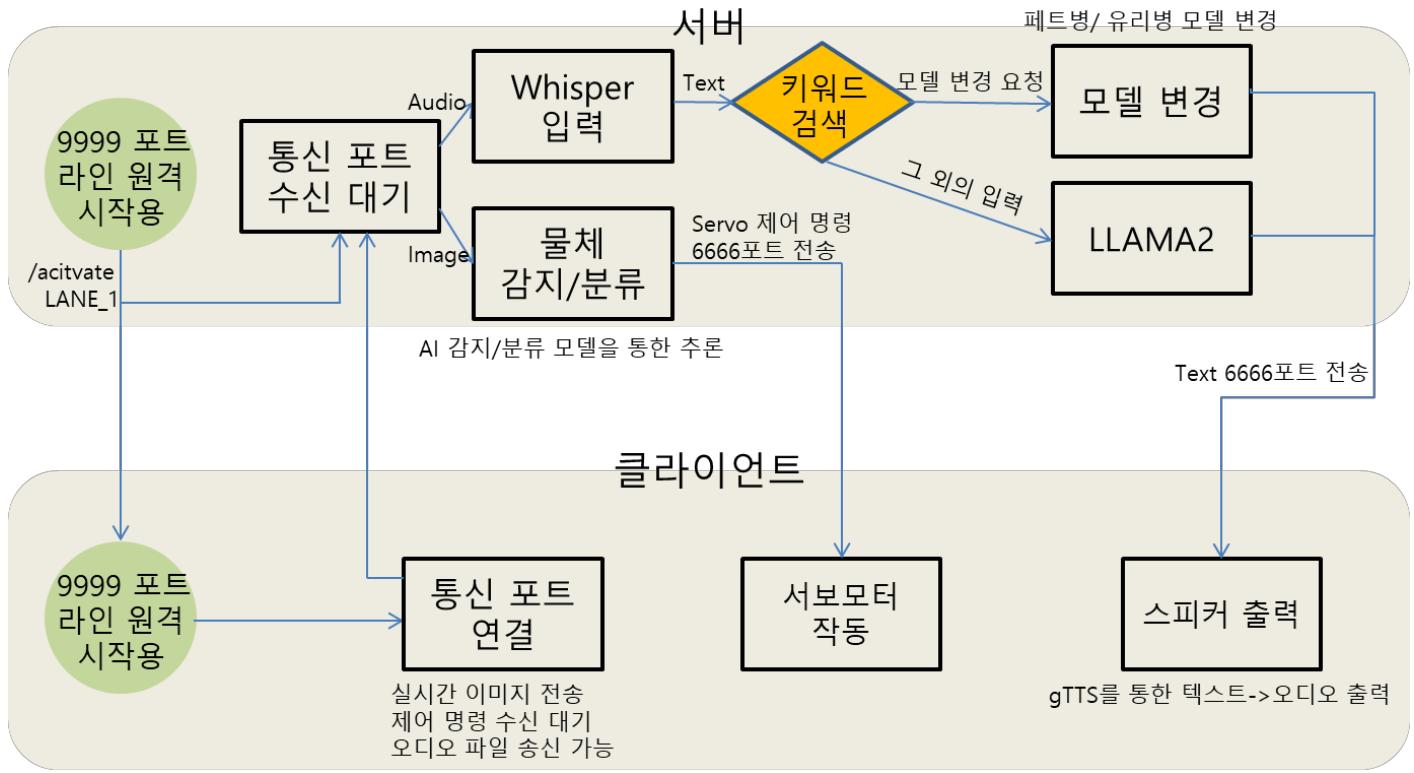
## 3. 시스템 구성도



<그림 2> 전체 시스템 구성도

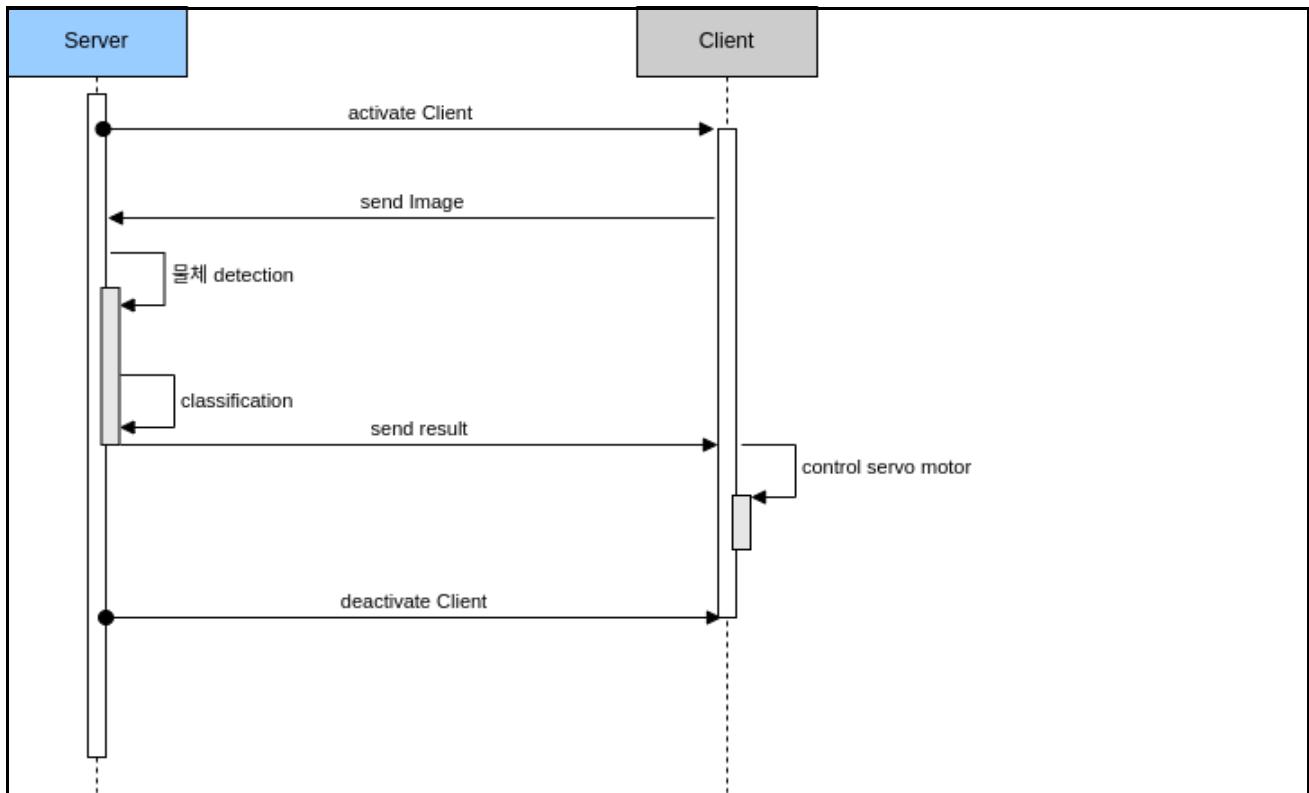
### 3 절 장치별 동작 순서도

#### 1. 전체 순서도

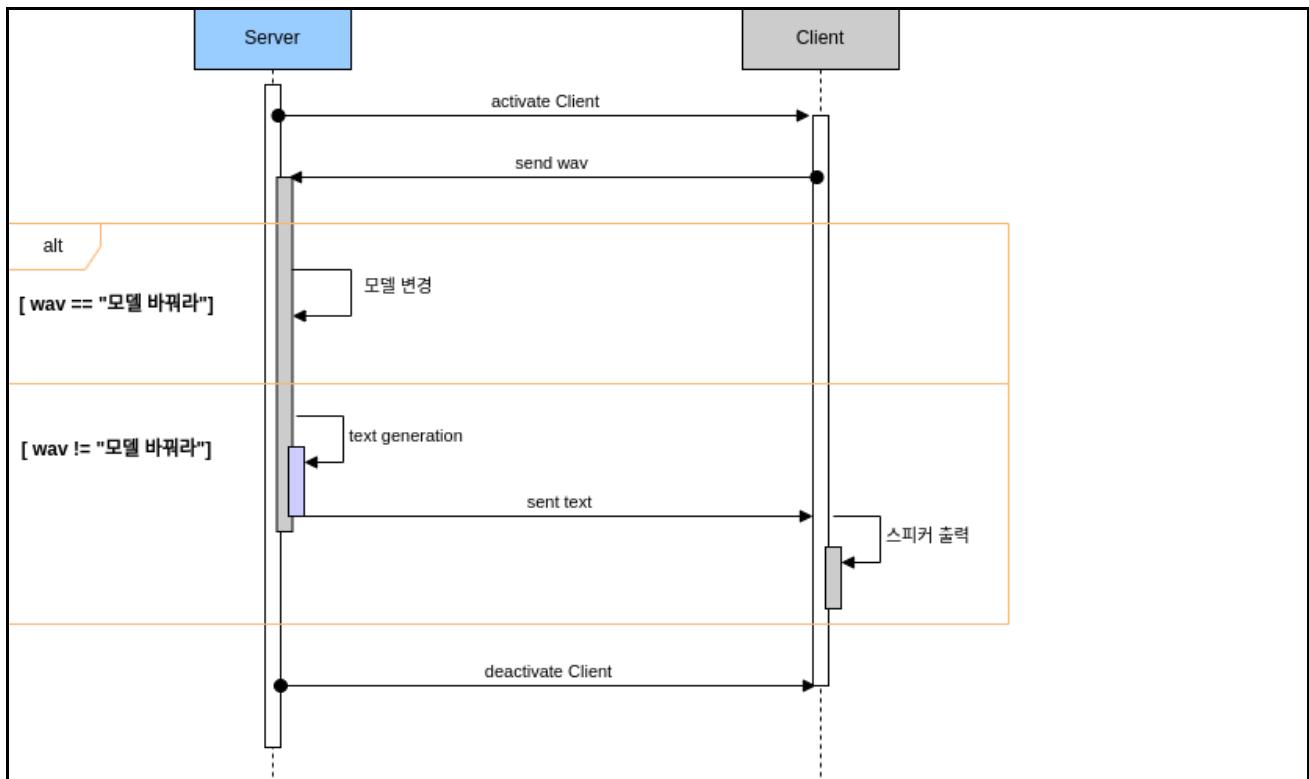


<그림 3> 전체 순서도

#### 2. 시퀀스 다이어그램

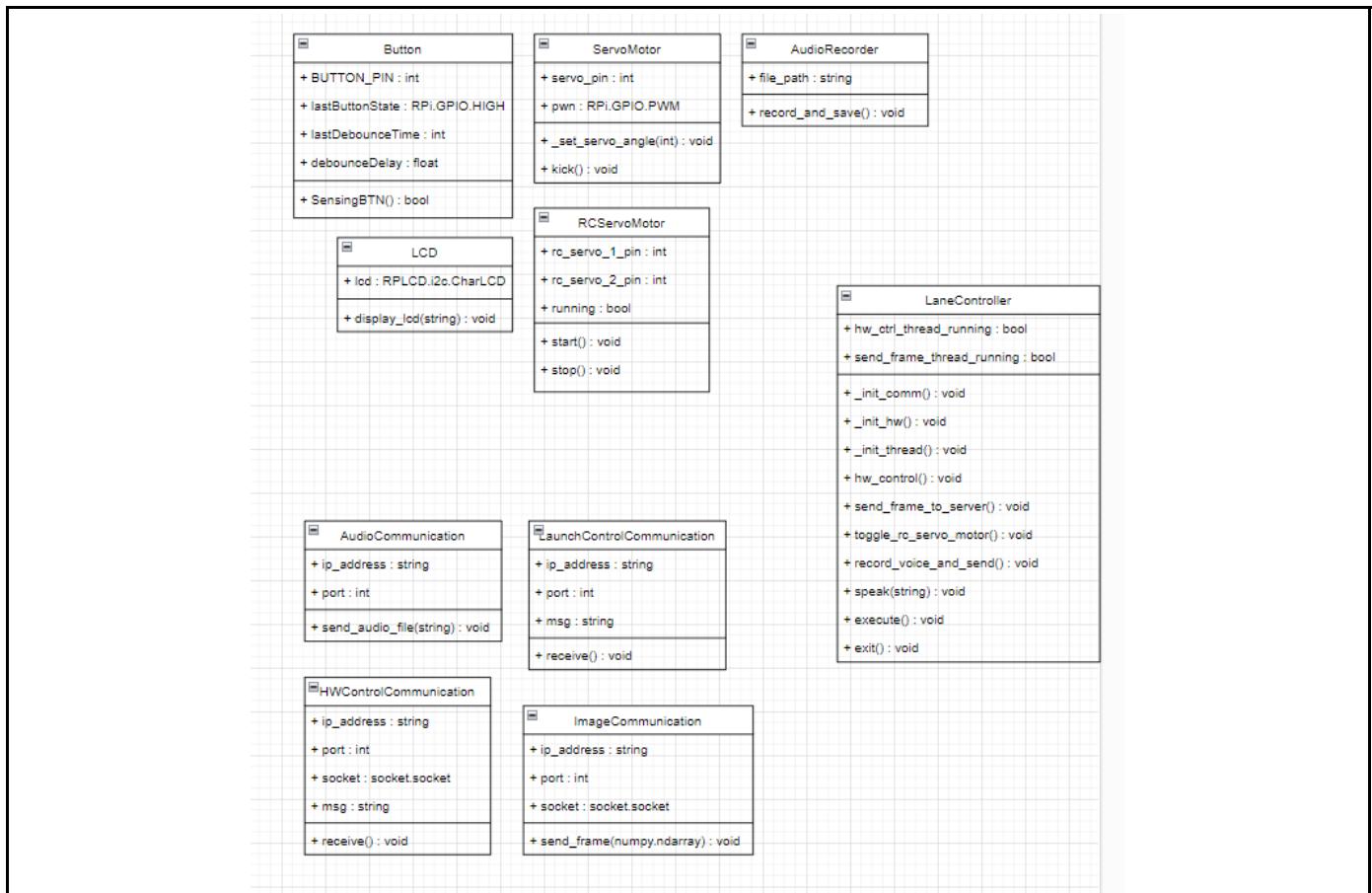


<그림 4> 시퀀스 다이어그램(1)

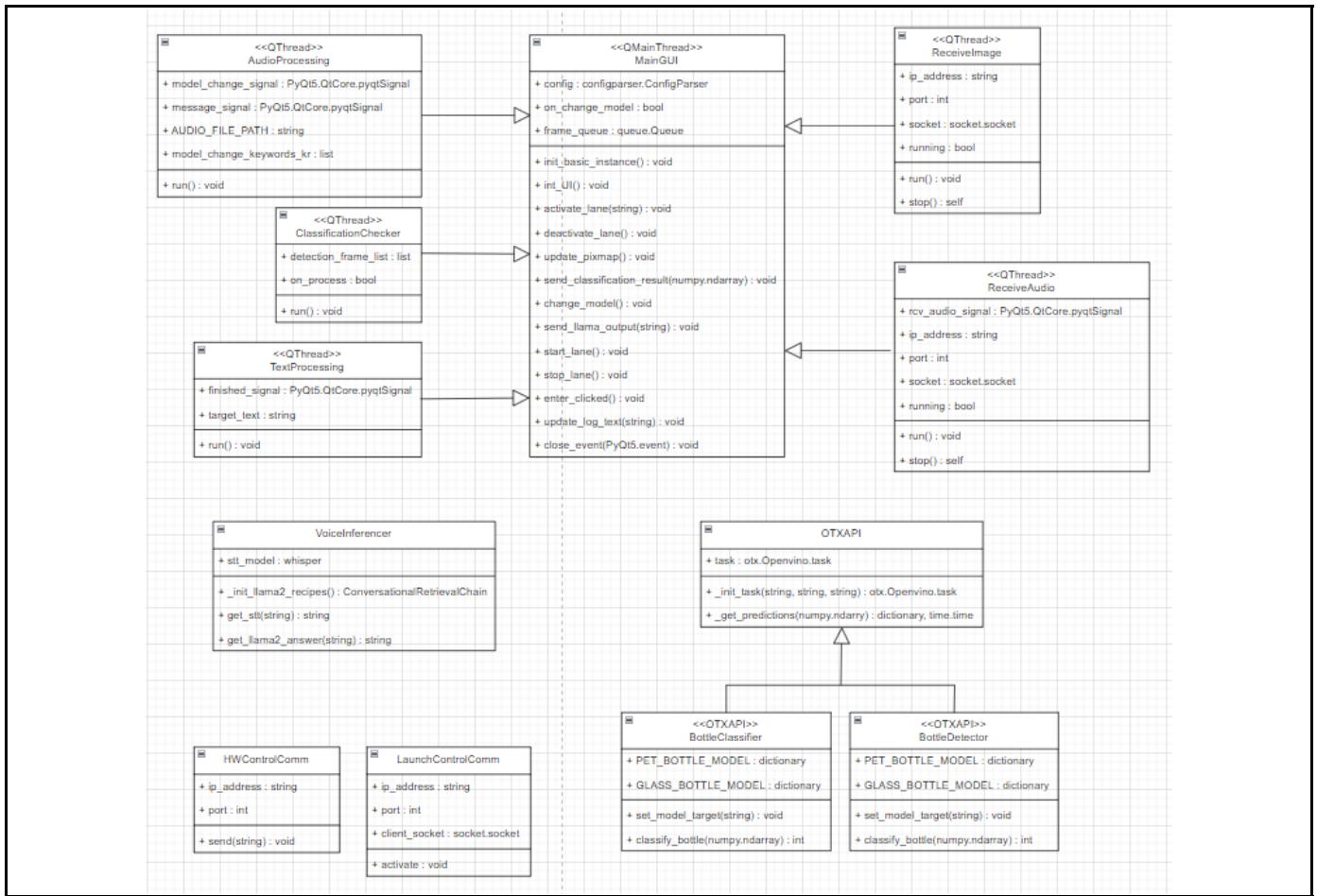


<그림 4> 시퀀스 다이어그램(2)

### 3. 클래스 다이어그램



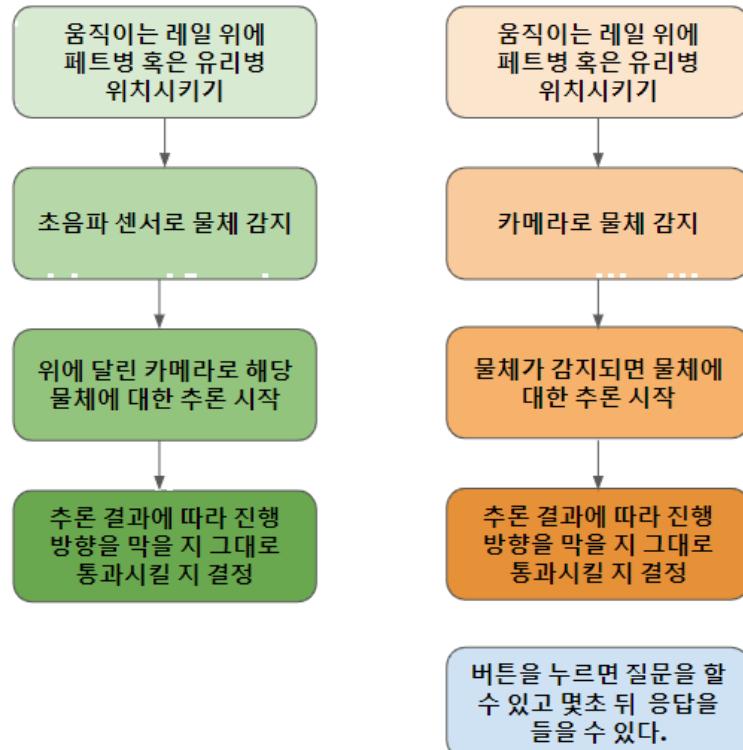
<그림 5> Client 클래스 다이어그램



<그림 6> Server 클래스 다이어그램

#### 4 절. 개발 과정 및 주요 기능

##### 1. 계획 수립



<그림 7> 이론상 구현 계획

<그림 8> 실제 구현 계획

## 2. 단계별 진행 과정

### <그림 9> EOF:EOF 개발 일정

### 3. 주요 기능

### 1) 페트병 감지

OTX의 pre-train된 YOLOX 모델을 전이학습하여 프레임 안의 object가 페트병인지 감지하고 해당 영역을 crop한다.

### 2) 페트병 라벨 분류

crop된 모든 이미지에 대하여 accuracy가 가장 높은 이미지를 classification 한다.

### 3) 유리병 감지

OTX의 pre-train된 YOLOX 모델을 전이학습하여 프레임 안의 object가 유리병인지 감지하고 해당 영역을 crop한다.

#### 4) 유리병 철 뚜껑 분류

crop된 모든 이미지에 대하여 accuracy가 가장 높은 이미지를 classification 한다.

## 5) 음성 인식

- whisper: openAI에서 제공하는 whisper 모델을 이용해 wav 파일에 담긴 사람의 말을 텍스트로 변환 한다.
  - all-mpnet-base-v2: huggingface 에서 제공하는 임베딩 모델로 PDF에 적힌 내용을 임베딩 한다.
  - llama2: llama2와 all-mpnet-base-v2로 임베딩한 벡터를 서로 엮어서 whisper로 변환된 텍스트를 입력으로 넣어 응답에 적합한 텍스트를 생성한다.

#### 6) 서버 프로그램

서버 측의 tshell scrip 파일을 통해 GUI를 띄울 수 있다. 해당 GUI에서는 클라이언트와 통신을 통해 영상을 받아 출력하고, 그 영상에 해당하는 내용이 로그로 출력된다. 메시지 입력으로 Lane start 및 모델 변경 등 다양한 기능을 구현할 수 있다.

### 7) 클라이언트 프로그램

서버와 통신을 한다. 서버 측에서 유리병, 페트병이 label로 분류되면 서보모터를 동작시킨다. 모델을 바꾸면 스피커로 출력을 해주고 LCD에도 표시된다. 버튼을 통해 lane start/stop을 조작할 수 있고, 다른 버튼으로 모델 변경 및 기능에 대한 질문을 할 수 있다.

## 5 절. 구현 결과

### 1. 완성 사진



<표 4> 완성 사진

### 2. 시연 영상

[유튜브 데모 영상](#)

### III. 결 론

#### 1. 향후 목표

(1) 데이터셋 구축에는 많은 시간이 소요된다. 페트병, 유리병 분류, 감지 모델 총 4개의 모델을 각각 2 천장씩 데이터셋을 구축하고 학습시키는데에만 1주일의 시간이 걸렸다. 추가로 데이터를 더 확보하여 데이터셋을 구축하면, 수치적인 accuracy 말고 실제 상황에서의 accuracy를 향상시킬 수 있을 것이다.

(2) 여러 LANE을 원격으로 관리가능한 하나의 서버 프로그램을 구상하고 본 시스템을 구현하였다. 현재는 LANE이 하나밖에 없고 코드도 향후 여려개의 레인에 맞춰 리팩토링할 것이다.

(3) 현재는 분리하는 것까지의 작업만 구현되어 있다. 향후 분리 이후의 추가 작업도 자동화 하는 로직을 추가할 것이다.

#### 2. 기대 효과

일반 사용자들이 그냥 라벨을 뗄 필요 없이 그냥 투명 페트병을 분리해서 포대에 담는 것으로 역할을 종료하고, 그 이후의 라벨 페트병 분류 작업은 별도의 라인에서 시행하여, 불필요한 시간 낭비를 줄일 수 있을 것이다. 또한 페트병에 국한되지 않고 다른 다양한 케이스로 확장되어 분류하는데 활용 될 수 있을 것이라 기대한다.

### IV. 참고 자료

#### • 참고 문헌

- 서영진, 『사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리파이』 (제이펍, 2021), 510-45.
- 로런스 모로니, 『개발자를 위한 머신러닝&딥러닝』 (한빛미디어, 2022), 47-61, 64-92.
- 이코 마코토, 『파이썬으로 배우는 머신러닝의 교과서』 (한빛 미디어, 2018), 207-223
- 오일석, 『파이썬으로 만드는 인공지능』 (한빛 아카데미, 2021), 135-142, 299-302, 309-313, 394-403
- 안효복, 『ARM으로 배우는 임베디드 리눅스』 (한빛 미디어, 2013), 564-631, 710-722

#### • 깃허브 링크

[https://github.com/yg-park/EOF\\_SeparateTrashCollection](https://github.com/yg-park/EOF_SeparateTrashCollection)