**스마트 신발 측정**

**(Safe Step)**

**[목 차]**

[Ⅰ. 서 론 3](#_TOC_250026)

[1절 개발 개요 3](#_TOC_250025)

[1. 개발 배경 3](#_TOC_250024)

[2절 작품 설명 3](#_TOC_250023)

1. [작품 목표 3](#_TOC_250022)
2. 주요 기능 및 특징 4

[Ⅱ. 본 론 5](#_TOC_250021)

1절 개발 환경 및 사용 기술 설명 5

1. [전체 개발 환경 5](#_TOC_250020)
2. [부품 리스트 6](#_TOC_250019)
3. [사용 기술 설명 8](#_TOC_250018)

[2절 작품 구성 11](#_TOC_250017)

1. [회로도 11](#_TOC_250016)
2. [시스템 구성도 13](#_TOC_250015)

[3절 개발 과정 및 기능 17](#_TOC_250014)

1. [Raspberry Pi Server 17](#_TOC_250013)
2. [Ai modeling 17](#_TOC_250012)
3. [Arduino 모듈 제어 21](#_TOC_250011)
4. [전체 시스템 작동 과정 21](#_TOC_250010)

[4절 구현 결과 22](#_TOC_250008)

1. [작품 외관 22](#_TOC_250007)
2. [작품 내부 23](#_TOC_250006)
3. 시연 영상…………………………………………………………………………………………………25

[Ⅲ. 결 론 26](#_TOC_250005)

1. [향후 목표 26](#_TOC_250004)
2. [기대 효과 27](#_TOC_250003)

[Ⅳ. 참고 자료 28](#_TOC_250002)

[참고 문헌 28](#_TOC_250001)

[소스 코드 28](#_TOC_250000)

Ⅰ. 서 론

1절 개발 개요

1. 개발 배경

신발 밑창은 자동차 타이어처럼 시간이 지날수록 마모되는 소모품입니다. 신발은 우리가 일상생활에서 항상 착용하는 필수품이며, 특히 낙상사고를 예방하는 데 중요한 역할을 합니다. 그러나 많은 사람들은 신발 밑창의 상태를 별도로 점검하지 않은 채 착용하고 다니며, 이는 특히 비, 눈, 얼음 등으로 인해 길이 미끄러워졌을 때 낙상사고로 이어질 수 있는 위험 요소가 됩니다.

실제로 신발 밑창에는 미끄럼 방지를 위한 고유한 패턴이 있으며, 이 패턴이 마모되면 그 기능이 크게 저하됩니다. 하지만 사람들은 일반적으로 밑창의 상태를 꾸준히 확인하지 않고, 심지어 마모가 심한 신발을 인지하지 못한 채 계속 착용하는 경우가 많습니다.

이러한 문제를 해결하고자 스마트 신발 마모 측정기가 개발하였습니다. 이 시스템은 사람들이 많이 다니는 공공장소에 설치되어 무료로 신발의 마모도를 측정해주고, 마모 상태에 따라 교체 권고 또는 즉각적인 미끄럼 방지 처치를 제공하여 낙상사고를 예방하는 것을 목적으로 합니다. 추후에는 공사장, 제조업 공장 등 안전이 중요한 환경에도 적용할 수 있도록 확장성을 고려하였습니다.

2절 작품 설명

1. 작품 목표

스마트 신발 마모 측정기는 딥러닝 기반 인공지능 모델을 이용하여 사용자의 신발 밑창을 촬영하고, 그 이미지로부터 마모도를 판단하는 장치입니다. 마모도가 일정 수준 이상일 경우, 자동으로 미끄럼 방지 스프레이를 분사하여 사용자가 일시적으로 미끄러짐을 방지할 수 있도록 설계되어 있습니다.

또한, 사용자에게는 웹 기반 사용자 인터페이스(UI)를 통해 신발의 마모 상태와 교체 권장 시기를 확인할 수 있도록 정보를 제공합니다. 특히 어르신들, 근로자들, 학생들처럼 신발 점검이 어려운 사람들에게 이 장치는 유용하게 사용될 수 있습니다.

1. 주요 동작 및 특징
2. **상처 인식 AI 컴퓨터 비전 영상처리**

사용자가 신발 밑창을 특정 위치에 두면, 카메라 모듈이 이미지를 촬영합니다. 해당 이미지는 NVIDIA Jetson Nano 등의 엣지 디바이스에서 처리되며, 내부에 탑재된 YOLOv8 기반의 딥러닝 모델이 마모 여부를 판단합니다. 마모 수준은 초기 상태 대비 남은 패턴 비율, 윤곽 흐림도를 종합적으로 판단해 4단계(10% /30% / 50% / 70%)로 구분합니다.

1. 음성 인식 및 대처 방법 출력

마모도가 위험 수준일 경우, 시스템은 서보모터와 스프레이를 제어하여 스프레이 노즐을 자동으로 활성화합니다. 분사는 짧은 시간동안 신발 밑창에 균일하게 분무되도록 설계되어 있으며, 스프레이 용량 및 압력은 일정하게 유지됩니다. 스프레이 노즐은 방향 센서 및 거리 센서(초음파 또는 적외선) 를 통해 신발 위치를 인식하고 정확하게 분사합니다.

1. 컴퓨터 비전과 음성 인식 기술을 활용한 응급 처치 및 시간 단축

마모도가 주의 수준일 경우, 사용자가 직접 사용할 수 있도록 스프레이 보관함이 자동 개방됩니다. 스프레이를 꺼낸 후 제자리에 되돌려 놓지 않으면 일정 시간이 지난 후 부저(경고음) 를 울려 사용자에게 알립니다. 보관함 내부에는 IR 센서(적외선 센서)가 있어 스프레이가 제자리에 있는지를 실시간으로 감지하며, 넣었을 경우 자동으로 문이 닫히도록 제어합니다.

1. 특정 명령을 통한 스프레이함 개폐 시스템

사용자는 결과를 모바일 웹 또는 키오스크 화면을 통해 바로 확인할 수 있습니다. 마모 상태에 따라 시각적인 표시(예: 색상별 등급 표시)와 함께, 추천 행동(신발 교체 / 주의 요망 등)을 제공합니다. 추후에는 사용자 이력 저장 기능과 함께, 주기적인 점검 알림 기능도 확장할 예정입니다.

Ⅱ. 본 론

**1절 개발 환경 및 개발 도구 설명**

1. 전체 개발 환경

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Server Main** | **Device1** | **Device2** | **Device3** |
| **Board** |  |  |  |  |
| **Raspberry Pi 4** | **Arduino MEGA** | **Jetson Nano** | **Arduino MEGA** |
| **Language** |  |  |  |  |
| **C, JS** | **C++** | **Python** | **C++** |
| **OS** |  |  |  |  |
| **Raspberry Pi OS (32-bit) Ubuntu 24.04** | **Windows 11** | **JetPack 4.6.0** | **Windows 11** |
| **Library** | **<arpa/inet.h>**  **<dirent.h>**  **<errno.h>**  **<fcntl.h>**  **<netinet/in.h>**  **<pthread.h>**  **<stdio.h>**  **<stdlib.h>**  **<string.h>**  **<sys/socket.h>**  **<sys/stat.h>**  **<sys/types.h>**  **<unistd.h>**  **<mysql/mysql.h>** | **<Wire.h>**  **<WiFiEsp.h>**  **<Adafruit\_NeoPixel.h>**  **<HX711.h>** | **socket**  **threading**  **sys**  **time**  **logging**  **cv2**  **os**  **random**  **numpy as np**  **ultralytics import YOLO** | **<SPI.h>**  **<WiFiEsp.h>**  **<Servo.h>** |
| **System** |  |  |  |  |
| **VSCode** | **Arduino IDE** | **VSCode** | **Arduino IDE** |

1. 부품 리스트

|  |  |  |
| --- | --- | --- |
| **제품명** | **이미지** | **링크** |
| Raspberry Pi 4 Model B |  | https://[www.devicemart.co.kr/goods/view?no=12234534](http://www.devicemart.co.kr/goods/view?no=12234534) |
| NVIDIA Jetson Nano Development Kit-B01 |  | https://[www.devicemart.co.kr/goods/view?no=12513656](http://www.devicemart.co.kr/goods/view?no=12513656) |
| Arduino MEGA 2560 (R3) |  | https://www.devicemart.co.kr/goods/view?no=34405 |
| Raspberry Pi Touch Display 2 |  | https://www.devicemart.co.kr/goods/view?no=15583929 |
| Logitec Webcam C270 |  | https://www.compuzone.co.kr/product/product\_detail.htm?ProductNo=167616 |
| HX711 |  | https://www.devicemart.co.kr/goods/view?no=1327440 |
| Load Cell |  | https://www.devicemart.co.kr/goods/view?no=1360978 |
| SG90-HV Digital Servo Motor |  | https://[www.devicemart.co.kr/goods/view?no=12503476](http://www.devicemart.co.kr/goods/view?no=12503476) |
| LED 스트립 |  | https://eduino.kr/product/detail.html?product\_no=1743&cate\_no=180&display\_group=1 |
| IR SENSOR 적외선센서 |  | https://[www.devicemart.co.kr/goods/view?no=1327441](http://www.devicemart.co.kr/goods/view?no=1327441) |
| ESP8266 시리얼 와이파이 모듈 |  | https://[www.devicemart.co.kr/goods/view?no=1278220](http://www.devicemart.co.kr/goods/view?no=1278220) |

|  |  |  |
| --- | --- | --- |
| 12V 1.5A 아두이노 아답터 [WT-12V1.5A-DC] |  | https://www.devicemart.co.kr/goods/view?no=12504449 |
| 신발 미끄럼방지 스프레이 |  | https://www.coupang.com/vp/products/6960307021?itemId=16941871054&vendorItemId=84119376417&q=%EB%AF%B8%EB%81%84%EB%9F%BC%EB%B0%A9%EC%A7%80%EC%8A%A4%ED%94%84%EB%A0%88%EC%9D%B4+%EC%8B%A0%EB%B0%9C&itemsCount=36&searchId=561e06031859584&rank=9&searchRank=9&isAddedCart |
| 아크릴판 10t |  | https://www.coupang.com/vp/products/8166148012?itemId=23303057313&vendorItemId=90204566097&q=아크릴+판&itemsCount=27&searchId=ea73ccf91440650&rank=0&searchRank=1&isAddedCart |
| 전동 분무기 |  | https://www.devicemart.co.kr/goods/view?no=14082512 |

3. 사용 기술 설명서

**1. PHP 및 데이터베이스(DB) 활용**

본 프로젝트에서는 사용자의 신발 마모 상태를 효과적으로 저장 및 관리하기 위해 PHP와 MariaDB를 활용하였습니다. PHP는 서버 측에서 실행되는 스크립트 언어로, 데이터베이스와 연동하여 웹 애플리케이션의 백엔드를 구축하는 데 필수적인 역할을 합니다. MariaDB는 MySQL 기반의 관계형 데이터베이스 시스템(RDBMS)으로, 신발 마모 분석 결과 및 촬영된 이미지의 경로를 저장하고 관리하는 역할을 담당합니다.

* 1. **PHP의 역할**
* 이미지 및 데이터 처리  
  젯슨 나노에서 촬영한 신발 밑창 이미지를 라즈베리 파이 서버로 전송하면 PHP가 이를 수신하여 지정된 디렉토리에 저장합니다. YOLOv8 분석 결과(마모도 등급)를 JSON 또는 텍스트 형태로 받아 데이터베이스에 저장합니다.
* 클라이언트와 서버 간 데이터 교환  
  PHP는 웹사이트에서 사용자가 요청한 신발 마모 데이터를 MariaDB에서 조회하여 제공하는 역할을 합니다. AJAX 요청을 처리하여 실시간으로 마모도 결과를 웹사이트에 출력할 수 있도록 구성하였습니다.
* 웹사이트 개발  
  HTML, CSS, JavaScript와 연계하여 마모 상태를 시각적으로 표현하는 웹 대시보드를 구축하였습니다.

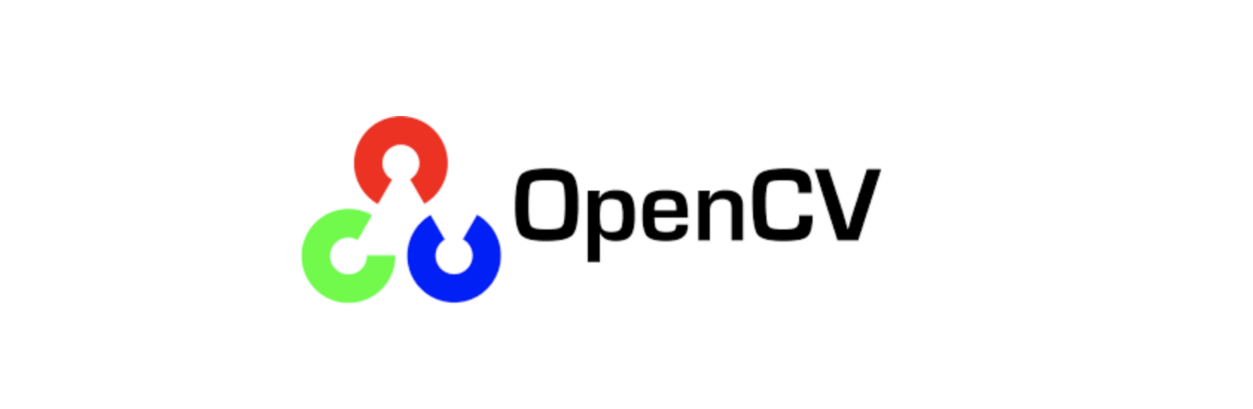
**1.2 데이터베이스 구조**

본 프로젝트에서 사용한 데이터베이스 테이블은 다음과 같습니다.

테이블 구조

* shoe\_wear\_data (마모 상태 데이터)
* id (INT, Primary Key, Auto Increment)
* timestamp (DATETIME) - 촬영된 시간
* wear\_percentage (FLOAT – ‘100', '70', '50', ’30’)
* image\_path (VARCHAR) - 촬영된 이미지 파일 경로

데이터베이스는 인덱싱을 최적화하여 검색 성능을 향상시켰으며, 사용자가 필요한 데이터를 빠르게 조회할 수 있도록 설계되었습니다.

**2. OpenCV 활용**

본 프로젝트에서는 신발 마모 분석을 보조하기 위해 OpenCV (Open Source Computer Vision Library)를 활용하였습니다. OpenCV는 이미지 처리에 최적화된 라이브러리로, YOLOv8이 보다 효과적으로 분석할 수 있도록 데이터를 전처리 하는데 사용되었습니다.

**2.1 OpenCV 적용 기능**

YOLOv8 분석을 위한 사전 처리 과정에서 OpenCV를 활용하여 그레이스케일 변환(흑백 처리)을 수행하였습니다.

cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) 함수를 이용하여 컬러 이미지를 흑백 이미지로 변환하였습니다.

**3. YOLOv8을 활용한 신발 마모 분석**

본 프로젝트에서는 YOLOv8 (You Only Look Once, Version 8)을 활용하여 신발 밑창의 마모 상태를 감지하였습니다. YOLOv8은 실시간 객체 탐지에 특화된 딥러닝 기반 모델로, 본 프로젝트에서는 신발 마모도를 '정상', '70', '50', ‘30’ 4가지 클래스로 분류하는 데 활용되었습니다.

**3.1 데이터셋 구축 및 학습 과정**

데이터셋 구축

본 프로젝트에서 사용할 신발 밑창 이미지를 직접 촬영하여 데이터셋을 제작하였습니다.

다양한 각도와 조명 환경에서 촬영한 신발 밑창 이미지를 포함하여 모델이 다양한 조건에서도 정확도를 유지할 수 있도록 하였습니다.

* 1. **YOLOv8 모델 학습**

YOLOv8 모델을 학습시키기 위해 직접 수집한 신발 마모 이미지를 라벨링하여 데이터셋을 구성하였습니다.

약 2,400장 이상의 신발 밑창 이미지를 활용하여 학습을 진행하였습니다.

PyTorch 기반의 Ultralytics YOLO 라이브러리를 활용하여 로컬에서 Nvidia 4060의 GPU 가속을 통해 학습하였습니다.

학습률, 배치 크기 등의 세부 설정을 조정하여 모델 성능을 최적화해주었습니다.

* 1. **실시간 마모도 감지 시스템 구현**

학습된 YOLOv8 모델을 젯슨 나노에 배포하여 실시간 분석이 가능하도록 구성하였습니다.

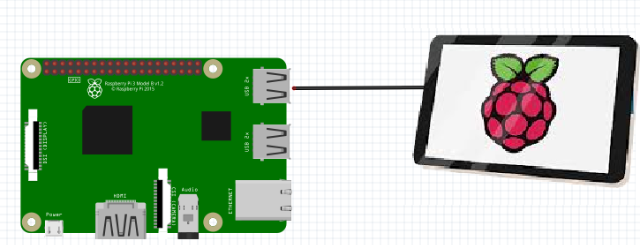
분석된 마모도 결과를 JSON 형식으로 변환하여 라즈베리 파이 서버와 MariaDB에 저장하였습니다.

웹사이트를 통해 사용자가 분석 결과를 직관적으로 확인할 수 있도록 구현하였습니다.

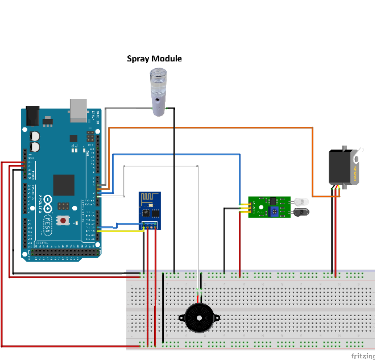
또한, 라즈베리 파이 디스플레이를 통해 실시간으로 마모 상태를 시각적으로 출력하였습니다.

2절 작품 구성

1. 회로도

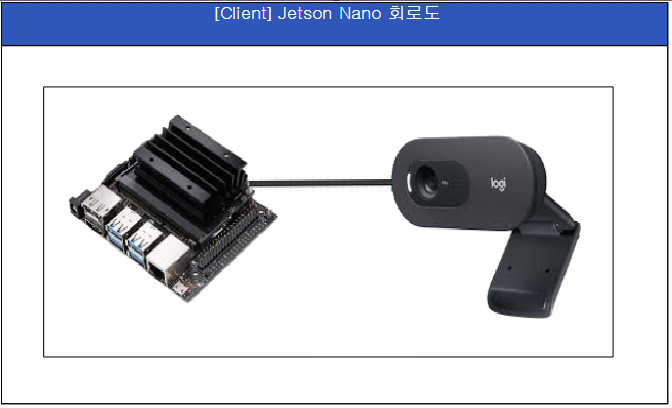


[Server] Raspberry Pi 회로도



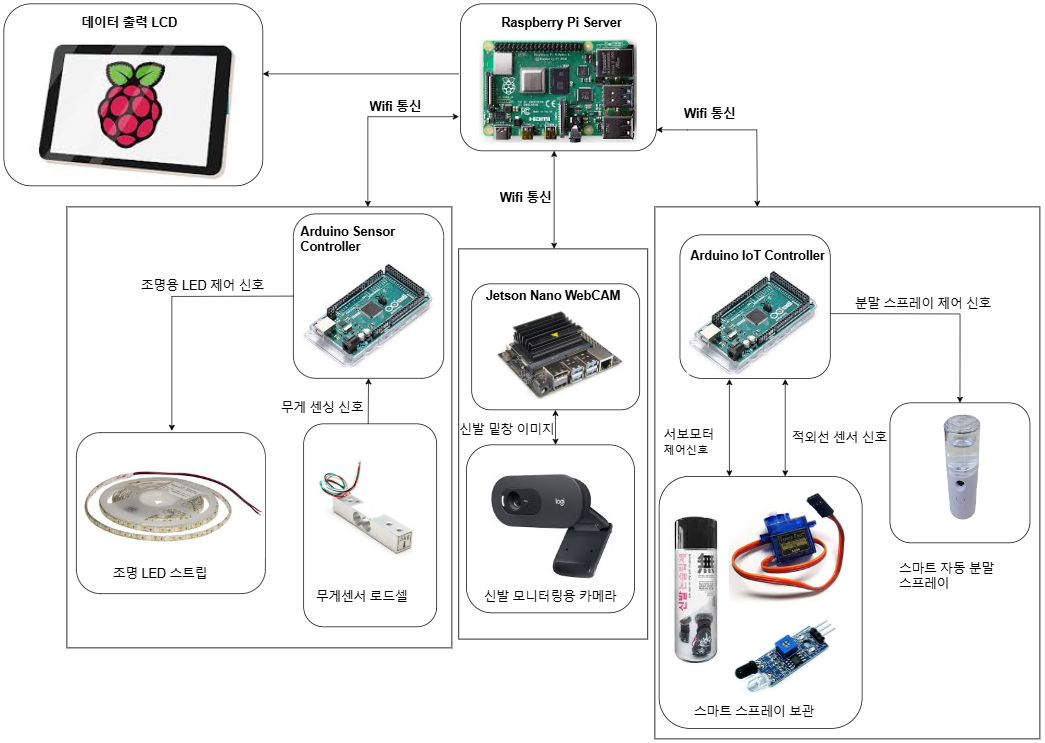
[Client] Arduino Mega(box) 회로도

[Client] Arduino Mega(무게 측정) 회로도



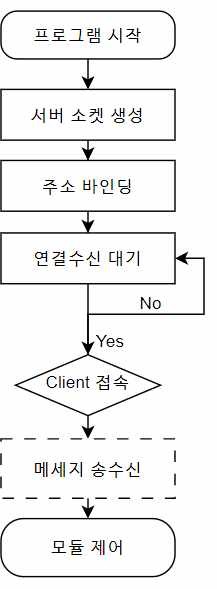
1. 시스템 구성도

* **전체 시스템 구성**



<전체 시스템 구성>

* 세부 시스템 구성

1. **Raspberry Pi (Server)**

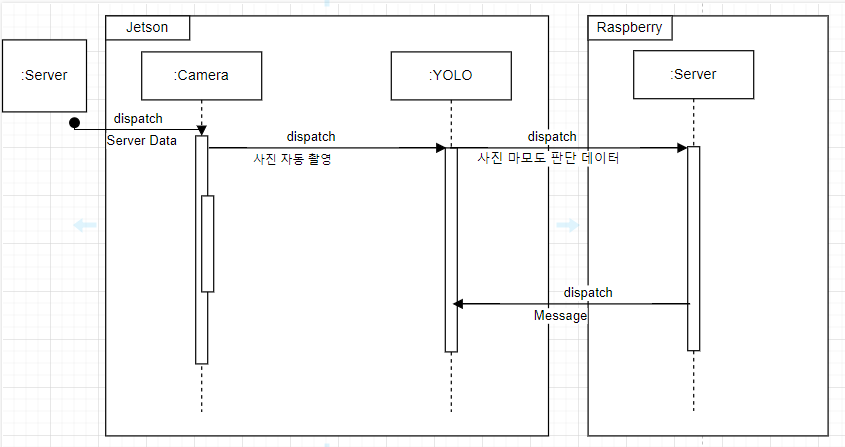
클라이언트 간의 데이터 송수신을 위해 서버 소켓을 생성합니다. 클라이언트가 접속하기 전까지 대기 상태를 유지하다가 클라이언트로부터 연결 요청이 들어오면 서버는 연결을 수락합니다.

이 부분에서 보안을 위해 서버에 클라이언트 해당 ID를 만들어 인증된 해당 클라이언트만 접속할 수 있도록 하였습니다.

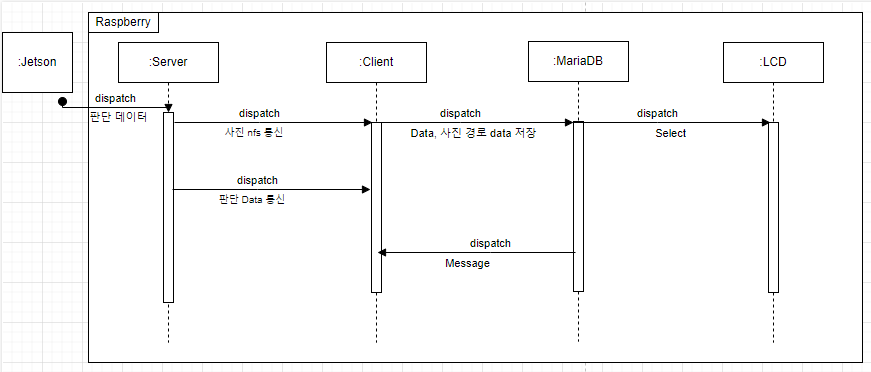
인증된 클라이언트는 메 시지를 서버에 전송할 수 있으므로 클라이언트 간의 송수신을 통해 다양한 접속기기로 아두이노와 젯슨을 제어하도록 했습니다.

1. **Jetson Nano (Client)**

Jetson Nano는 딥러닝을 이용해 객체를 감지하는 역할을 담당합니다. 웹캠으로 신발을 촬영하고 판단한 결과를 Raspberry Pi Server와의 소켓 통신을 통해 Raspberry Pi Client에 송신합니다 그리고 송신된 결과 값을 서버가 받으면 DB에 저장하게 해주었습니다.

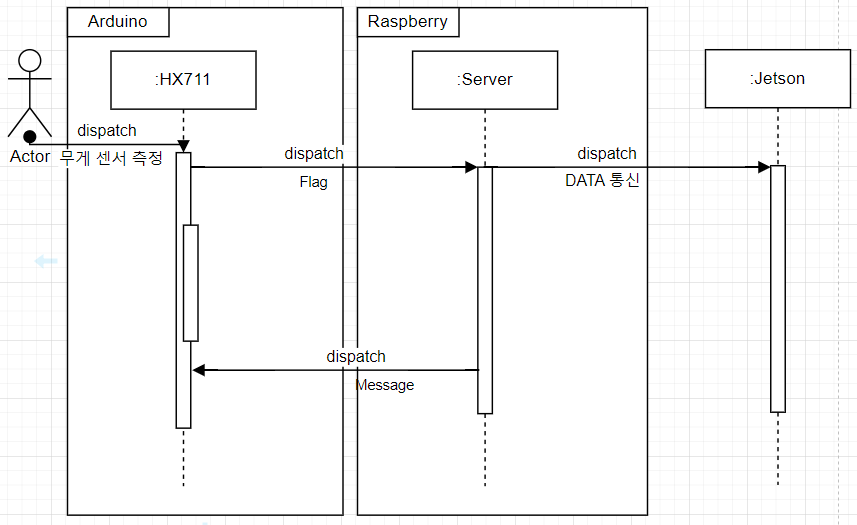
****

1. Raspberry Pi (Client)



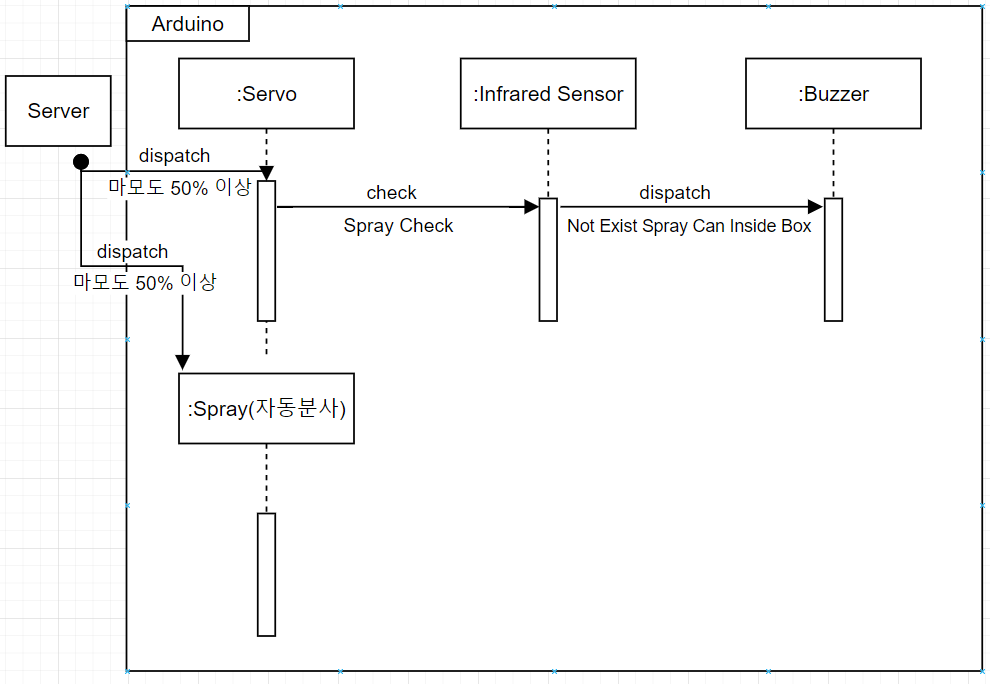
Jetson에서 받은 판단 완료된 데이터를 Raspberry Pi Server와의 와이파이 통신을 통해 Client로 전송하고, Client는 해당 데이터를 MariaDB에 저장한 후 LCD에 출력합니다.

1. **Arduino (Client)**

****

사용자가 신발걸이에 신발을 걸게 되면 Arduino에 연결된 무게 센서가 값을 읽고 설정한 무게보다 더 큰 값의 무게가 들어오게 되면 Flag 값을 Raspberry Pi Server와의 와이파이 통신을 통해 보내주게 됩니다. Flag 값을 받은 Raspberry Pi Server에서는 Flag값을 Jetson에 전달해 주게 됩니다. Jetson은 Flag를 받아서 정해진 Flag값을 받게 되면 연결된 카메라를 통해 신발 밑창을 찍습니다. Yolov8으로 미리 학습한 모델을 통해 찍은 신발 밑창 사진에서 신발 밑창 마모도를 판단합니다.

1. **Arduino (Client)**

****

Raspberry Pi 서버는 신발 마모도 데이터를 분석하여 Wi-Fi를 통해 Arduino로 전송합니다. 마모도가 50% 이상인 경우, Arduino는 서보모터를 작동시켜 미끄럼 방지 스프레이 보관함을 자동으로 열어줍니다. 사용자가 스프레이 캔을 꺼내면 박스 하단의 적외선 센서가 이를 감지합니다. 스프레이 사용 후 캔을 다시 보관함에 넣으면 센서가 이를 감지하여 서보모터가 작동해 박스를 닫습니다. 만약 2분 이내에 스프레이가 보관함에 반납되지 않으면 경고음이 울립니다. 신발 마모도가 70% 이상으로 심각한 경우에는 시스템이 자동으로 보관함 내부의 스프레이 노즐을 작동시켜 걸려있는 신발 밑창에 직접 미끄럼 방지 스프레이를 분사합니다.

3절 개발 과정 및 기능

1. Raspberry Pi Server

Raspberry Pi Server는 무게를 측정하는 Arduino Mega와 사진을 촬영하고 찍은 사진을 판단하는 Jetson Nano 모듈로부터 클라이언트 연결을 받아들여 데이터와 명령어를 수신하는 중추적 역할을 합니다. Raspberry Pi Server 안에 있는 MariaDB에 받은 데이터를 저장하고 연결된 LCD에서 웹페이지를 통해 결과 값과 사진을 보여주는 역할을 합니다.

1. Ai modeling

구간 별로 마모된 신발의 데이터 수집

저희의 프로젝트(Safe Step)에서는 신발 밑창의 마모도를 자동으로 인식하여 안전사고를 예방하는 시스템을 개발하고자 했습니다. 시중에 마모된 신발 데이터셋이 부재하고, 특히 단계별 마모도를 구분한 데이터가 전무하여 직접 데이터셋을 구축하기로 결정했습니다. 한정된 개발 기간 내에 효과적인 학습을 위해 단일 종류의 신발(슬리퍼)을 선택하여 집중적으로 데이터를 수집했습니다. 슬리퍼는 밑창 패턴이 비교적 단순하고 마모 진행 과정을 명확히 시각화 할 수 있다는 장점이 있었습니다.

실험을 위해 동일한 디자인과 크기의 슬리퍼 4쌍을 준비했습니다. 실제 사용 환경에서의 마모 패턴을 참고하여 그라인더를 사용해 인공적으로 마모도를 조정했습니다. 마모도는 정상(0%), 경도 마모(30%), 중도 마모(50%), 심도 마모(70%)의 4단계로 구분했으며, 각 단계별로 밑창의 홈 깊이와 패턴 소실 정도를 정밀하게 조정했습니다.

데이터 촬영은 일정한 조명 환경 일반에서 진행했으며, 핸드폰 카메라를 사용하여 촬영을 하였습니다. 배경은 흰색 무광 종이를 사용하여 신발 밑창의 경계를 명확히 구분할 수 있게 했습니다. 왼쪽 신발과 오른쪽 신발을 각각 300장씩, 총 4단계의 마모도를 포함하여 최종적으로 2,400장(300장 × 2(좌/우) × 4(마모도))의 고품질 이미지를 확보했습니다.

**데이터 사진 예시:**



Right\_30%

Right\_Normal



Right\_70%

Right\_50%

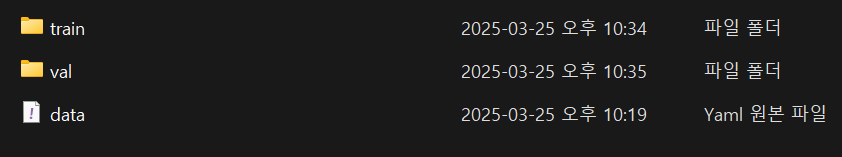
이미지 전처리를 위해 OpenCV 라이브러리를 활용했습니다. cv2.cvtColor() 함수를 사용하여 RGB 이미지를 그레이스케일로 변환했습니다. 이후 오픈소스 도구인 'labelImg' 1.8.1 버전을 사용하여 다음과 같은 8가지 클래스로 객체 감지용 라벨링을 진행했습니다:

- Left\_normal, Right\_normal (정상 마모도)

- Left\_30, Left\_50, Left\_70 (왼쪽 신발의 30%, 50%, 70% 마모도)

- Right\_30, Right\_50, Right\_70 (오른쪽 신발의 30%, 50%, 70% 마모도)

데이터셋 구성을 위해 전체 2,400장 중 10%인 240장을 층화 추출법(stratified sampling)을 통해 검증용(validation) 데이터로 분리했습니다. 이는 각 클래스별로 균등한 비율(각 30장씩)로 추출하여 검증 데이터의 균형을 유지했습니다. 나머지 2,160장은 학습용(training) 데이터로 활용했습니다.



<데이터셋 폴더 구성>

모델 선정을 위해 여러 객체 감지 알고리즘을 검색해본 결과, Ultralytics의 YOLOv8 small 모델을 채택했습니다. 이 모델은 실시간 객체 감지에 최적화되어 있고, 경량화된 구조로 모바일 기기 등에 배포하기 용이하며, 다른 유사 모델들보다 추론 속도가 빠르다는 장점이 있습니다. 또한 적은 양의, 그리고 우리처럼 커스텀 데이터셋에서도 우수한 성능을 보인다는 선행 연구 결과를 고려했습니다.

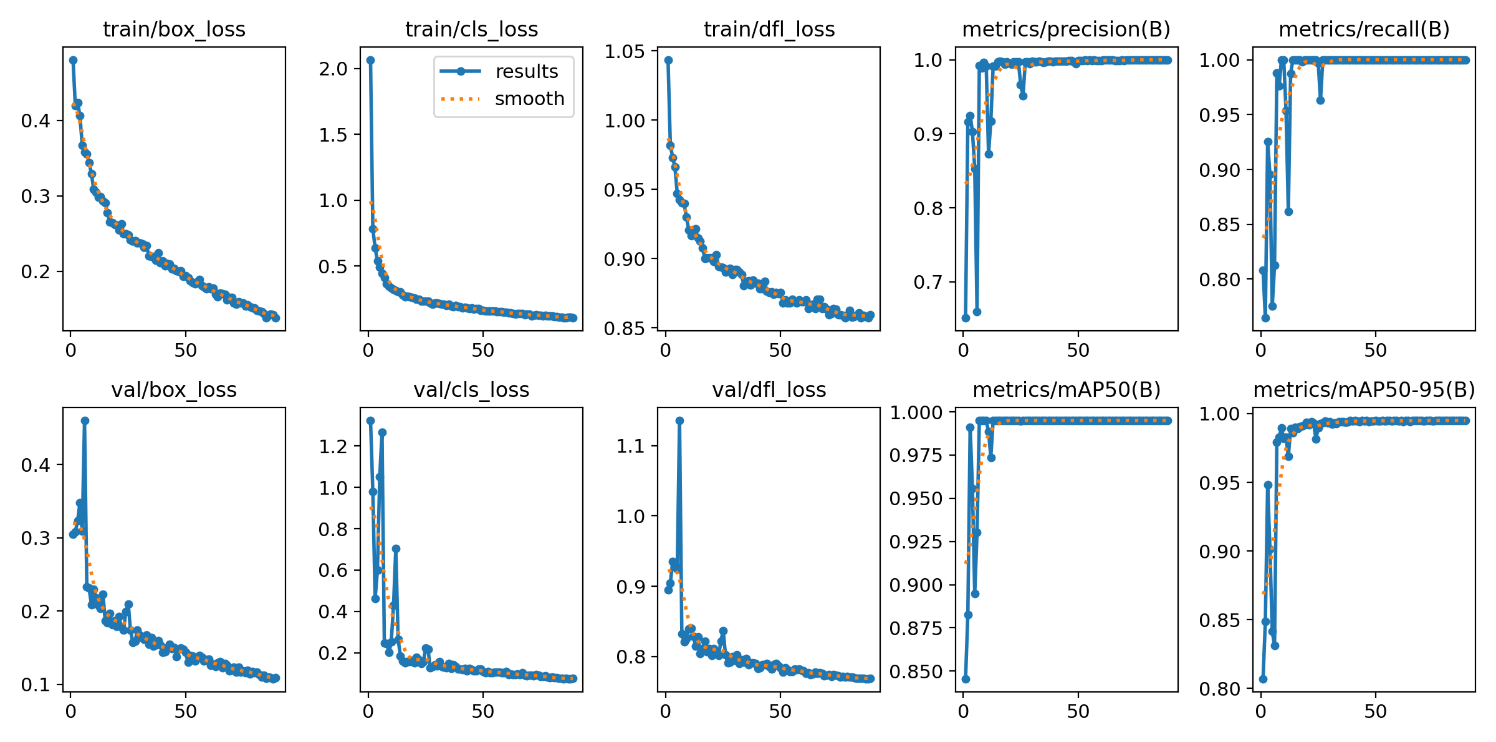
학습 환경은 로컬 워크스테이션에 Nvidia RTX 4060 GPU, CUDA Toolkit 12.8, PyTorch 2.6.0을 구성하여 진행했습니다. 세부설정은 다음과 같이 설정했습니다:

- Epoch: 100 (충분한 학습 반복을 통한 정확도 향상)

- Batch size: 16 (GPU 메모리 효율성과 학습 안정성 고려)

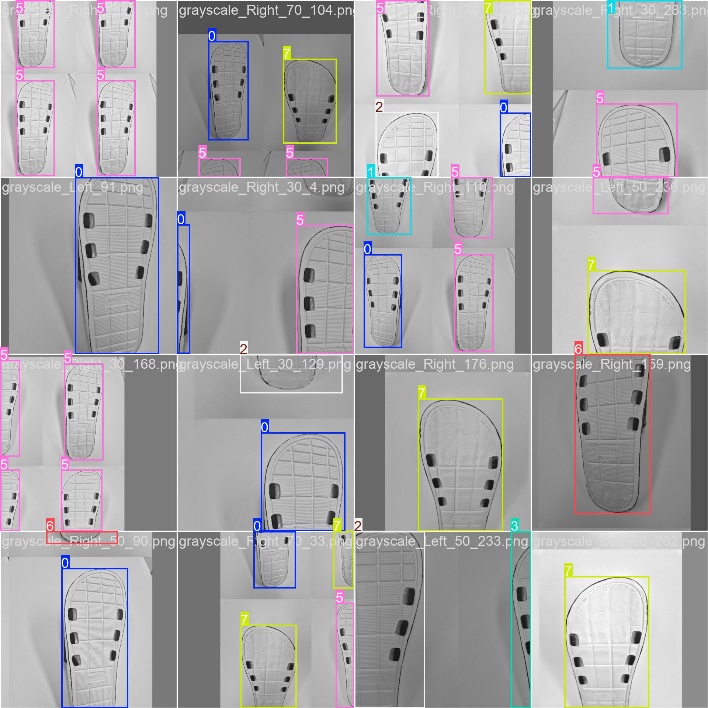
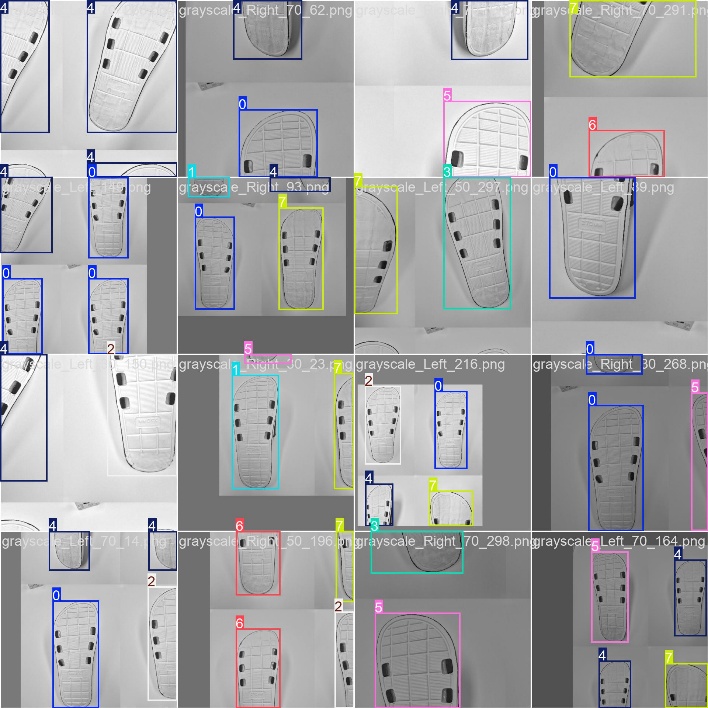
- Learning rate: 0.001 (초기값: 0.005)

- Data augmentation: 랜덤 밝기 조정(±30%), 랜덤 스케일(±50%), 모자이크를 적용하여  
다양한 환경에서의 인식율을 증가



<ai 학습 결과 그래프>

학습 진행 결과, mAP(mean Average Precision)@0.5 지표를 기준으로 약 40 epoch부터 검증 데이터에서 0.92 이상의 안정적인 정확도를 보이기 시작했습니다. 최종적으로 100 epoch 학습 완료 후, 검증 데이터에서 0.95이상의 정확도를 보였습니다.



<yolov8 판단 데이터>

위와 같이 여러 환경에서 데이터들을 학습시켜주었습니다.

1. Arduino 모듈 제어

사용자가 신발 걸이에 신발을 걸었을 때 설정한 무게보다 높은 경우 Raspberry Pi Server에 Flag 값을 보내고 다시 Flag값을 Jetson에 보내서 맞는 Flag값이 들어오게 되면 연결된 카메라를 통해 걸려있는 신발의 밑창을 찍습니다.

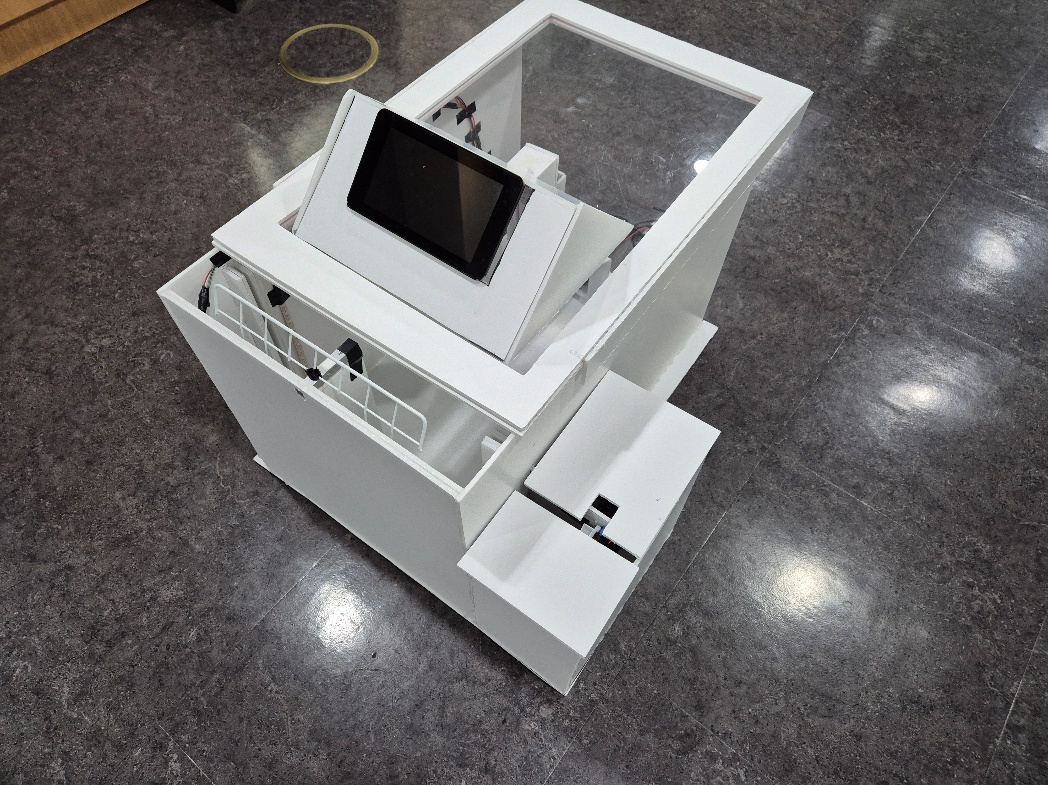
찍은 사진을 Yolov8로 미리 학습한 모델로 판단한 마모도를 Raspberry Pi Server에서 받고, 찍은 사진을 nfs 통신으로 주고받는 공유폴더 안에 저장합니다. 저장하는 Raspberry Pi Client에서 받은 마모도가 50%일 경우 50% 메세지를 보내고 70%일 경우 70% 메세지를 Arduino Box Client에 보냅니다. 만약 Arduino가 50%의 메세지를 받을 경우 미끄럼방지 스프레이가 있는 박스를 열어주고 부저음을 울립니다.

그 후 1분 후에도 스프레이가 박스 안으로 들어오지 않으면 적외선 센서로 감지 후 부저음을 울려줍니다. 만약 70% 메세지를 받으면 50% 기능과 같은 기능을 하고 추가 기능으로 전체 박스 천장에 있는 스프레이가 신발 밑창을 향해 자동 분사합니다.

1. 전체 시스템 작동 과정 정리
2. **무게 감지 단계**
   * 사용자가 신발 걸이에 신발을 걸면 무게 센서가 이를 감지
   * 설정된 기준치보다 높은 무게가 감지되면 Raspberry Pi Server에 Flag 값 전송
   * Raspberry Pi Server는 해당 Flag 값을 Jetson으로 전달
3. **이미지 촬영 단계**
   * Jetson이 올바른 Flag 값을 수신하면 연결된 카메라가 활성화
   * 카메라가 걸려있는 신발의 밑창을 촬영
4. **마모도 분석 단계**
   * 촬영된 신발 밑창 이미지를 YOLOv8로 사전 학습된 모델을 통해 분석
   * 분석된 마모도 정보는 Raspberry Pi Server로 전송
   * 촬영된 이미지는 NFS 통신을 통해 공유 폴더에 저장
5. **조치 단계**
   * Raspberry Pi Client가 마모도 정보를 수신
   * 마모도에 따른 대응:
     + **50% 마모 시**: Arduino Box Client에 50% 메시지 전송
       - 미끄럼방지 스프레이 박스 개방
       - 알림 부저 작동
       - 1분 후 스프레이가 감지되지 않으면 적외선 센서로 확인 후 추가 부저 알림
     + **70% 마모 시**: Arduino Box Client에 70% 메시지 전송
       - 미끄럼방지 스프레이 박스 개방
       - 알림 부저 작동
       - 1분 후 스프레이가 감지되지 않으면 적외선 센서로 확인 후 추가 부저 알림
       - 추가 기능: 박스 천장의 스프레이가 신발 밑창을 향해 자동 분사

4절 구현 결과

1. 작품 외관

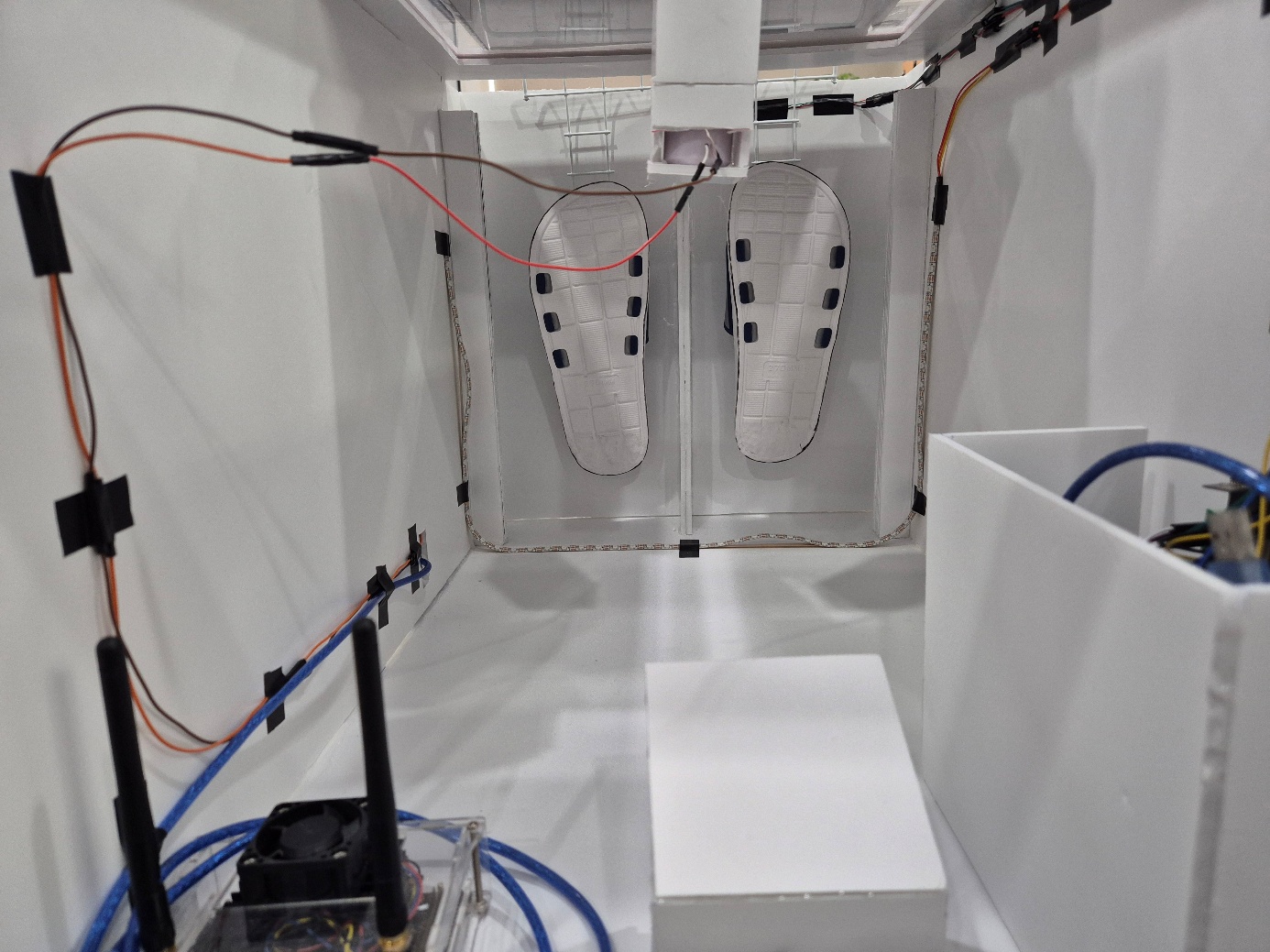
**** 

<제품 외부1 사진> <제품 외부2 사진>



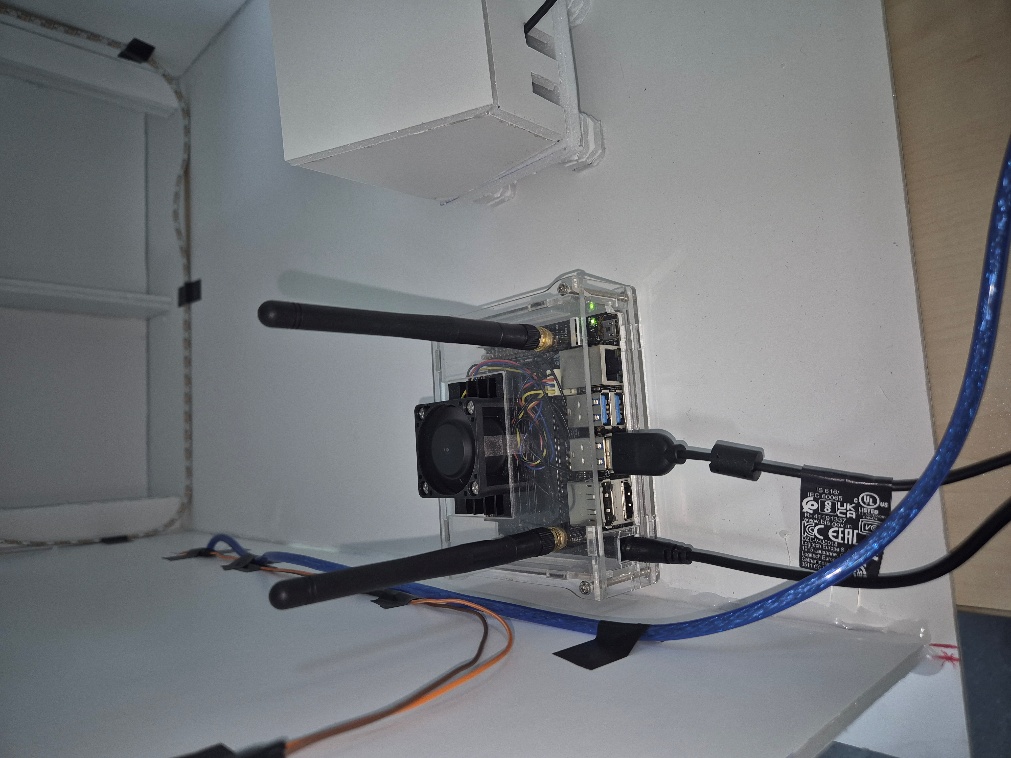
<제품 외부3 사진>

1. 작품 내부

****

<제품 내부 사진>

 <디바이스 1 회로 구성 사진> <디바이스 1 LED, 로드셀 연결 사진>

 <디바이스 2 회로 구성 사진>  <디바이스 3 회로 구성 사진>

1. 시연 영상

첨부영상 참고

Ⅲ. 결 론

1. 향후목표

본 스마트 신발 마모 측정기에는 신발의 마모된 정보를 학습시키기 위한 다양한 마모도의 데이터셋과 학습한 인공지능의 마모도 측정 결과를 통해 사용자에게 현재 신발의 밑창 상태 위험도를 시각화 정보로 제공하고, 다양한 스마트 IoT 제어기를 이용해 사고 예방을 위한 후처리를 제공합니다, 다양한 산업 현장에서의 적용 가능성을 보완하여 아래와 같은 향후 목표를 제안합니다.

* 1. 빅데이터 분석 기술 적용

현재 학습된 데이터는 특정한 신발에 대한 4단계 구간의 마모도 케이스로 한정되었지만 다양한 신발에 대한 방대한 데이터베이스를 추가할 경우, 임의의 신발에 대해서도 분류가 가능할 것으로 예상되며, 다양한 산업 현장에서 신발의 밑창 상태에 의해 발생할 수 있는 사고를 미연에 방지할 수 있습니다.

* 1. 사용자의 편의성 및 개인화를 위한 UI 개선

사용자가 시각화된 정보를 사용할 수 없는 시각장애인, 혹은 한글화 되어있는 UI정보를 인식하기 어려운 외국인일 경우에도, 본 제품을 사용할 수 있도록, 음성 인식 방식을 도입할 수 있을 것 같습니다. 또한 데이터베이스에 특정 사용자의 신발 정보를 꾸준히 축적할 경우, 맞춤형 관리를 제안하는 기능을 도입할 것으로 보입니다.

* 1. 상황 인식 및 신고 연계 기능 개발

특수한 목적을 지닌 공간에서는 특히나 산업재해를 방지하는 것이 중대한 상황일 수 있으므로, 주기적으로 신발의 관리상태가 좋지 않은 사용자는 데이터센터를 통해 입장을 제한하는 방식을 도입하는 방식이 사용될 수 있으며, 매우 위험한 신발의 경우, 미끄럼방지용 스프레이 뿐이 아닌, 추가적인 더 방지효과가 강한 후처리를 제안할 수 있습니다.

* 1. 다양한 아날로그 센서 추가를 통한 기능 개발

현재 신발의 밑창 정보를 판단하는 방식은, 인공지는 데이터 학습모델 기반의 시각 정보를 이용한 분류방식입니다. 이러한 시각화 정보 뿐이 아닌, 다양한 아날로그 센서를 IoT 방식으로 연동하여 좀 더 정밀한 신발의 마모도 측정이 가능할 것 같습니다. 초음파 센서, 압력 저항 센서 등을 이용할 수 있을 것 같습니다.

2. 기대 효과

**1) 사고 미리 예방**

마모된 신발이 산업재해 및 인명피해로 이어질 수 있는 상황을 현장에서 즉시 방지할 수 있도록, 간편한 측정과 후처리 기능이 구현되어 있습니다.

2) 겨울철 빙판길 낙상사고 다발지역 관리

보행시 미끄럼 사고가 취약한 겨울철 얼어붙은 도로변과, 미끄러짐에 의한 낙상사고 다발지역 등, 특수한 장소에서는 특히 본 제품이 효율적으로 사용될 수 있습니다. 특별히 미끄러짐을 방지해야 하는 공간에서 효율적으로 사용될 전망으로 보입니다.

3) 안전 취약계층 맞춤형 신발 관리 시스템

현재 서버에 연동된 데이터베이스는 노인, 산업현장 근로자, 어린이 등 안전 취약계층별 신발 마모도 임계치와 위험 패턴을 분석하여, 각 사용자 그룹에 최적화된 안전 경고와 대응책을 제공합니다. 특히 요양시설이나 산업현장에서 관리가 가능해 안전사고 예방에 효과적압니다.

4) 사용자의 신발 관리 서비스 접근성 향상

본 제품의 가장 뛰어난 요소는 측정이 매우 간편하고, 즉각적이라는 점에 있습니다. 따라서 평소 신발의 상태 측정을 매우 간편화 할 수 있고, 부주의에 의한 인명 피해를 크게 감소시킬 수 있습니다.

Ⅳ. 참고 자료

참고 문헌

* <https://m.blog.naver.com/eduino/222065695612> 로보다인시스템 2020
* <https://like-grapejuice.tistory.com/401> 가상현실을 꿈꾸던 방랑자의 이모저모 2024
* [**https://github.com/ultralytics/ultralytics**](https://github.com/ultralytics/ultralytics)
* [**https://github.com/opencv/opencv**](https://github.com/opencv/opencv)

소스 코드

* 첨부 파일 참고