

AIoT 프로젝트

미니 소방 로봇

2조 불나방

권오준 박민혁 유광균 이윤혁 임지원 장혜원

00. 목차

[Intel] 엣지 AI S/W 아카데미

- 01 프로젝트 팀 구성 및 역할
 - 02 프로젝트 개요
 - 03 프로젝트 수행 절차 및 방법
 - 04 프로젝트 수행 경과
 - 05 자체 평가 의견
-

01.

팀 구성/역할

[Intel] 엣지 AI S/W 아카데미

| 이름 | 역할 | 담당 업무 |
|-----|----|----------------------|
| 권오준 | 팀장 | 화재 위치 측정 및 좌표 송신 |
| 유광균 | 팀원 | 화재 감지 AI 모델 설계 및 훈련 |
| 임지원 | 팀원 | 서버 프로그램 및 안드로이드 앱 개발 |
| 박민혁 | 팀원 | 서버 프로그램 및 알림 기능 개발 |
| 장혜원 | 팀원 | 소화 장치 부분 제어 및 설계 |
| 이윤혁 | 팀원 | 터틀봇 자율 주행 및 제어 |

02. 배경

[Intel] 엣지 AI S/W 아카데미



- 스프링클러를 설치할 수 있는 장소에 제한이 있으며, 수조 설치 문제, 비용 문제 등이 있음
 - 스프링클러설비의 화재안전성능기준(NFPC 103)
 - 수원의 양은 스프링클러 설치 헤드 수의 1.6m^3 (1.6톤) 배 이상이어야 함
 - 소화 수조를 사용하지 않는 간이 스프링클러의 경우도 설치에 적지 않은 비용이 들어감
- 연기 / 열로 감지하는 스프링클러는 반응 까지 시간이 걸림

02. 배경

[Intel] 엣지 AI S/W 아카데미



스프링클러는 초기 화재(작은 불) 감지에 시간이 걸릴 수 있음

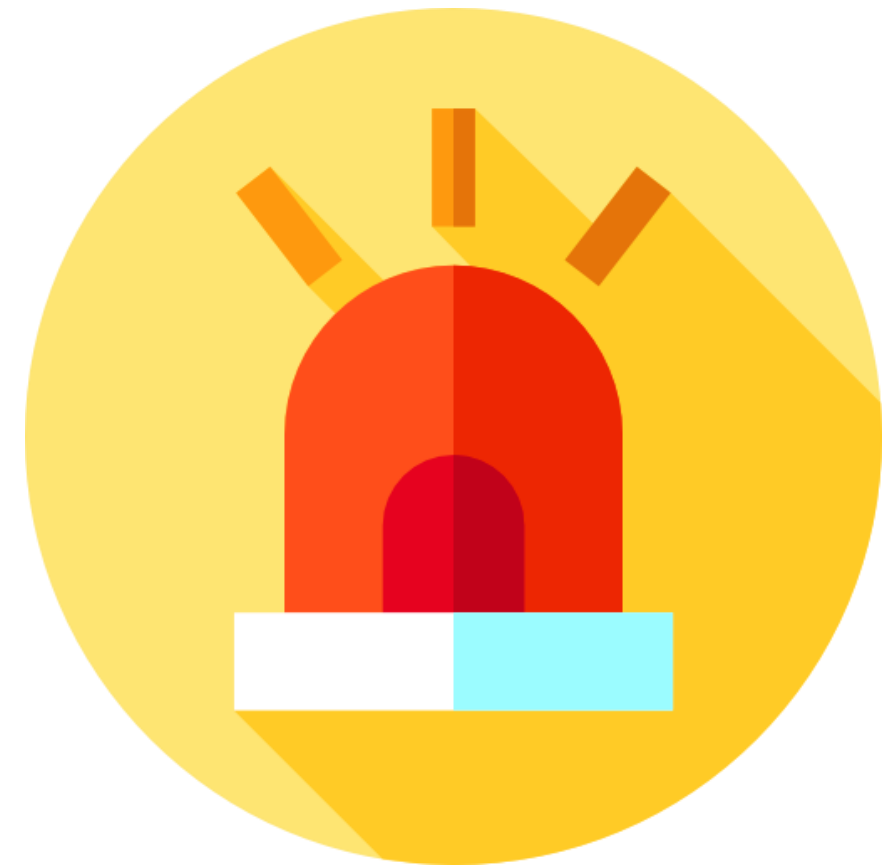
또한, 스프링클러는 **넓은 범위**에 걸쳐 소화를 진행하기에 오작동시 재산 피해가 발생할
가능성이 존재

02. 활용 방안

[Intel] 엣지 AI S/W 아카데미



화재 발생 부분에만 소화 작업을
펼치기 때문에 재산 피해 최소화

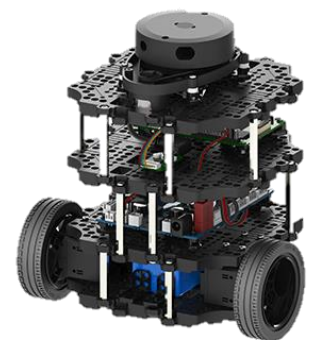


자동 작동 시스템으로
사람이 없는 상태에서도
골든 타임을 놓치지 않을 수 있음

환경/장비

하드웨어

소방 로봇



Turtlebot 3



Jetson Nano



팬틸트



가습 장치



웹캠

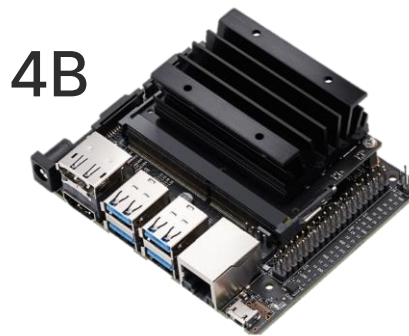


열화상 카메라

CCTV



Raspberry Pi 4B



Jetson Nano



웹캠

서버



데스크탑 PC

02.

환경/장비

[Intel] 엣지 AI S/W 아카데미

소프트웨어

소방 로봇



Ubuntu 20.04



ROS1 Noetic
(20.04)



C++



Python 3.8

CCTV



Python 3.11



OpenCV 4

서버



Ubuntu 20.04



YOLO v8



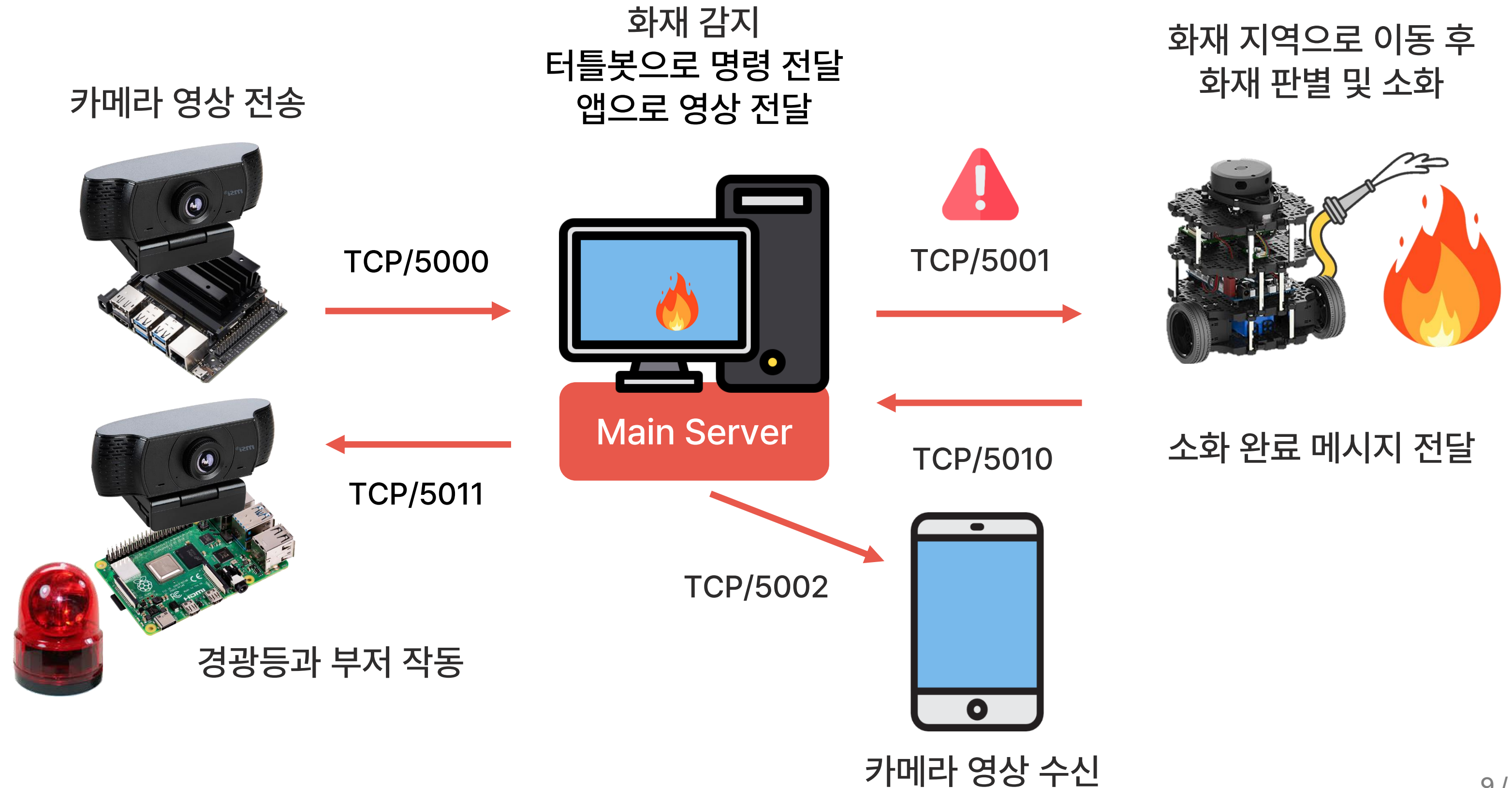
Qt5



OpenCV 4

02. 구조도

[Intel] 엣지 AI S/W 아카데미



03. 작업과정

[Intel] 엣지 AI S/W 아카데미

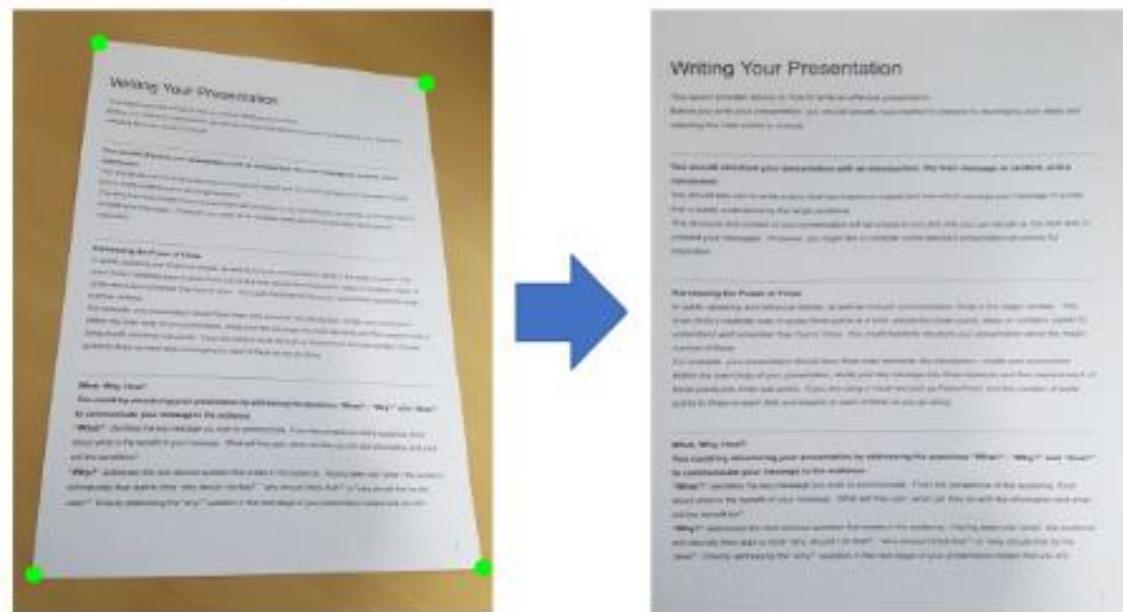
화재 좌표 전송

OpenCV의 **원근 변환** 기능을 사용하여 카메라 이미지 상의 좌표를 목표 좌표로 변환

- `cv::getPerspectiveTransform` : 두 개의 사각형 사이의 변환 행렬을 계산
- `cv::perspectiveTransform` : 변환 행렬을 이용하여 좌표를 변환

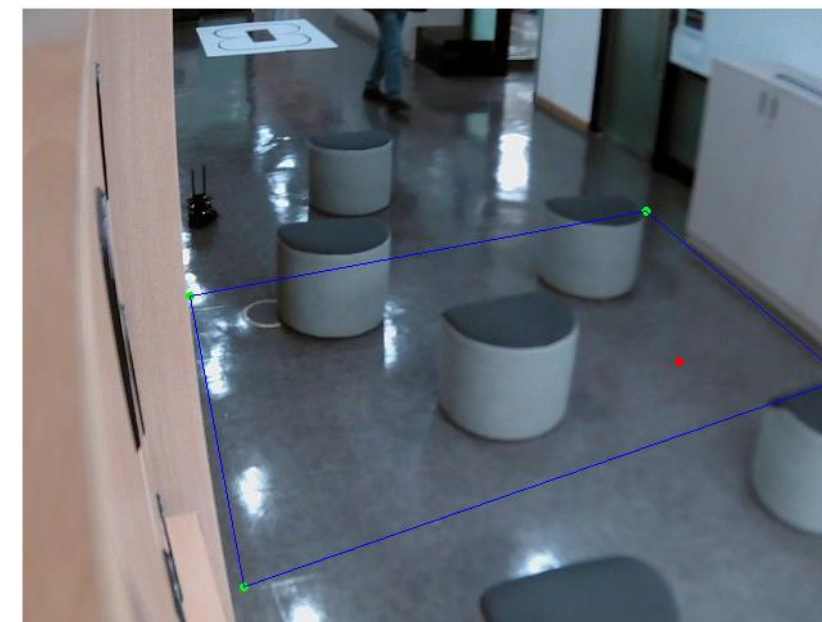
원근 변환(Perspective Transform):

임의의 직사각형을 다른 형태의 사각형으로, 또는 그 반대로 변형할 수 있는 변환



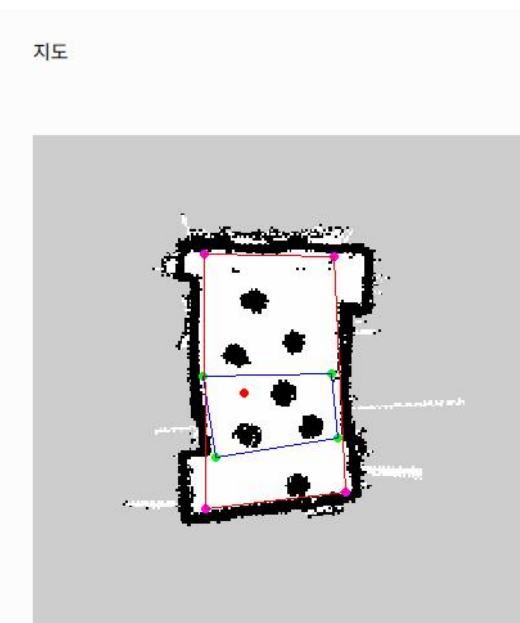
Perspective Transform 예시

(출처: <https://mvje.tistory.com/129>)



카메라 이미지의 좌표를 (왼쪽)

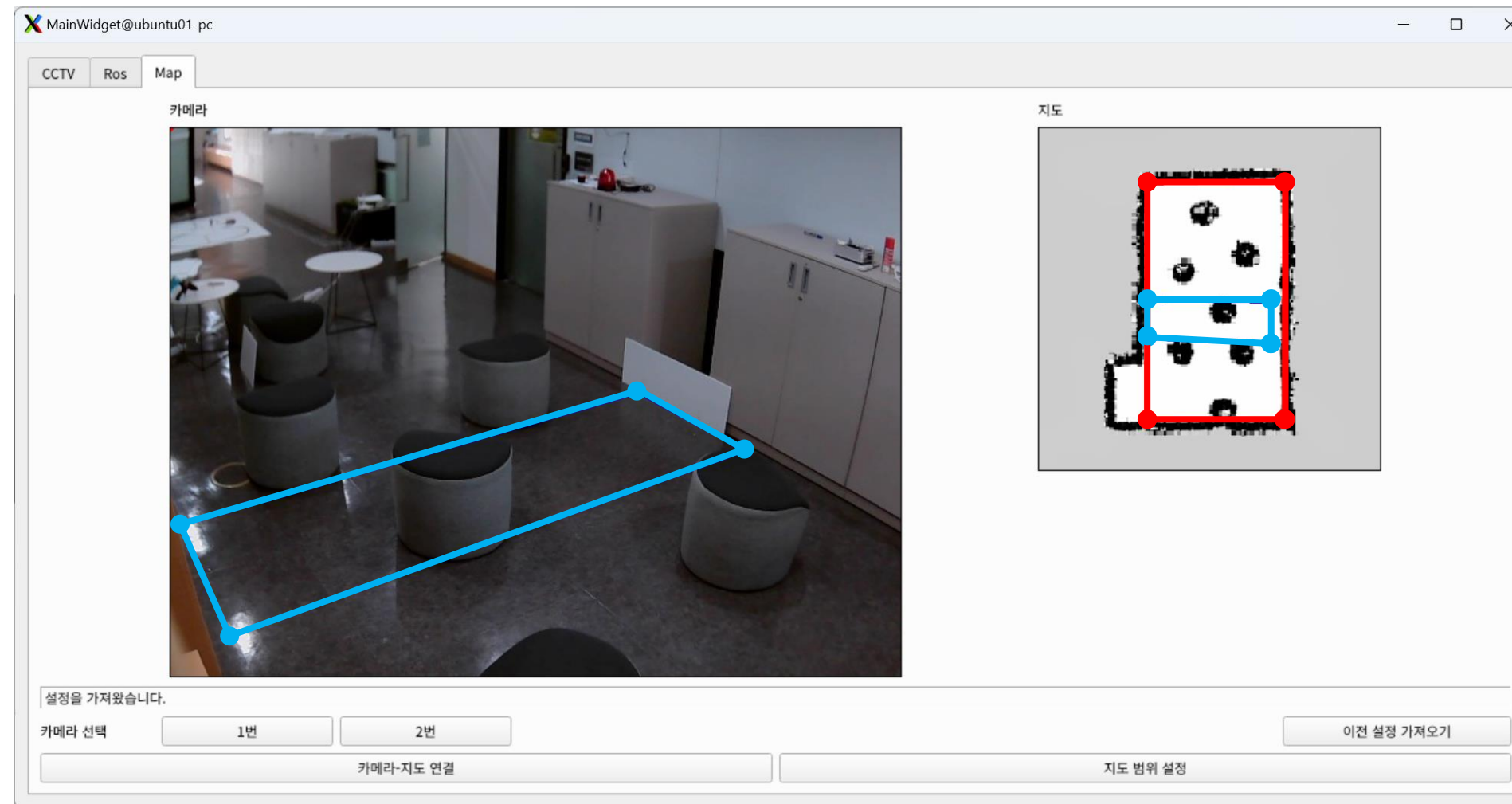
지도 위의 좌표로 변환한 모습 (오른쪽)



03. 작업과정

[Intel] 엣지 AI S/W 아카데미

화재 좌표 전송



```
std::vector<cv::Point2f> src_vector;  
std::vector<cv::Point2f> dst_vector;  
  
// set 4 points  
src_vector.push_back(cv::Point2f(mouse.x, mouse.y))  
;  
dst_vector.push_back(cv::Point2f(mouse.x, mouse.y))  
;  
  
transform_array = cv::getPerspectiveTransform(src_v  
ector, dst_vector);
```

Qt 프로그램의 맵핑(mapping) 기능 화면

카메라-지도 맵핑(파란색)과 지도-로봇 맵핑(빨간색)을 설정하는 화면

영역을 설정하면 cv::getPerspectiveTransform() 함수를 이용하여 변환 행렬을 생성한다.

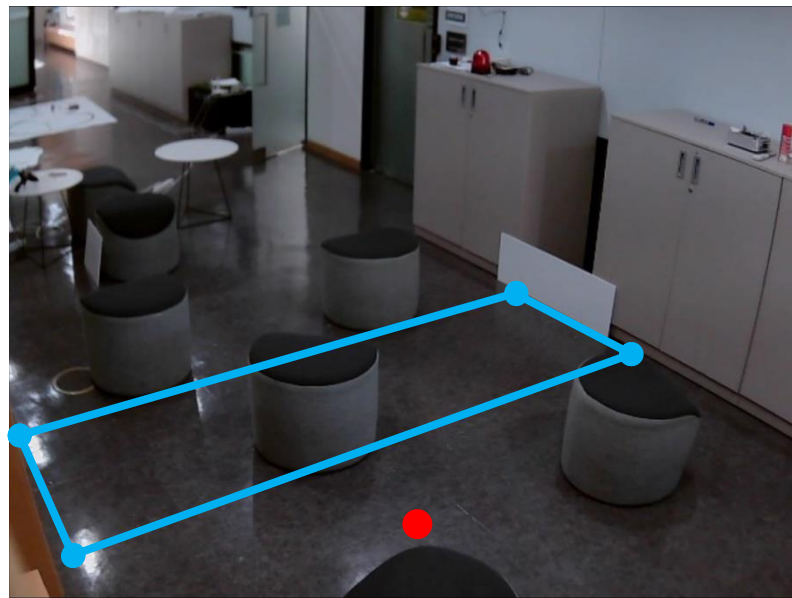
03. 작업과정

[Intel] 엣지 AI S/W 아카데미

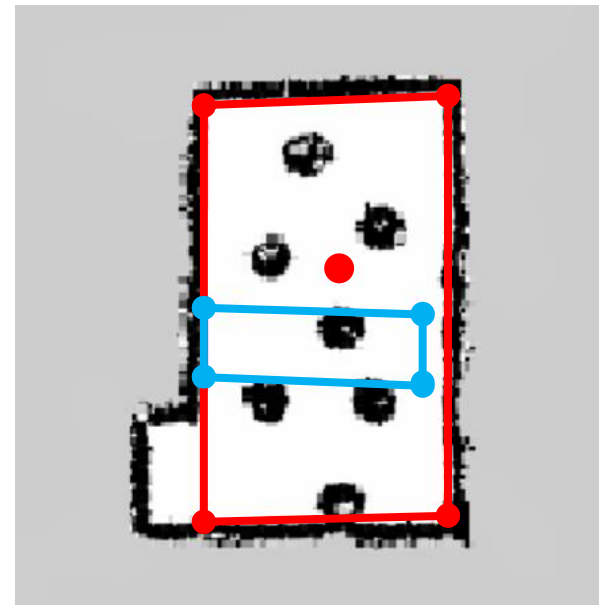
화재 좌표 전송

소방 로봇에게 좌표가 전달되는 과정

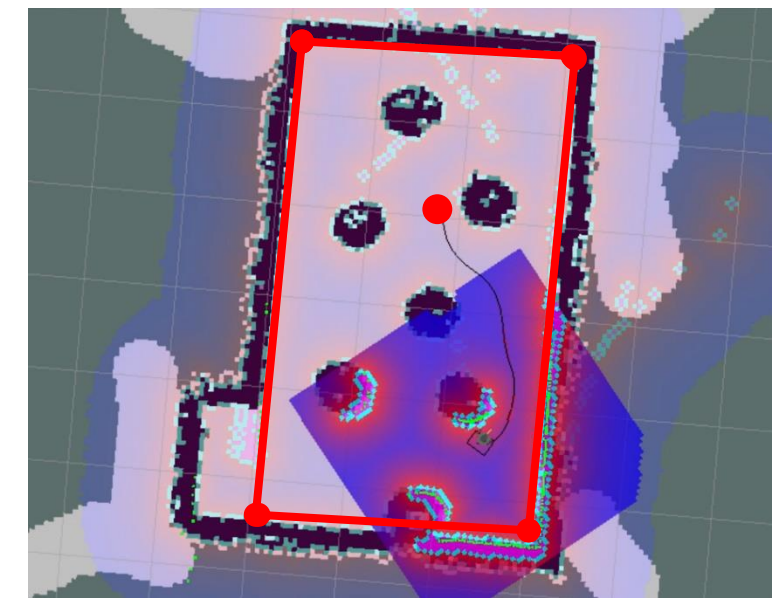
1. 카메라 이미지에 불이 감지될 경우, 카메라 이미지 상의 불의 중심 좌표를 구함
2. 불의 좌표를 지도 이미지 상의 좌표로 변환
3. 지도 이미지 상의 좌표를 소방 로봇의 좌표계 값으로 변환



`cv::perspectiveTransform(cam-map)`



`cv::perspectiveTransform(map-bot)`



변환된 좌표는 TCP 소켓을 통해 문자열 형식으로 로봇에 전달

"GOAL@x@y"

03.

[Intel] 엣지 AI S/W 아카데미

작업과정

CCTV 화재 데이터 셋

다양한 상황의 화재 데이터로 학습 진행



CCTV에 잡히는 불꽃을 인식하지 못함

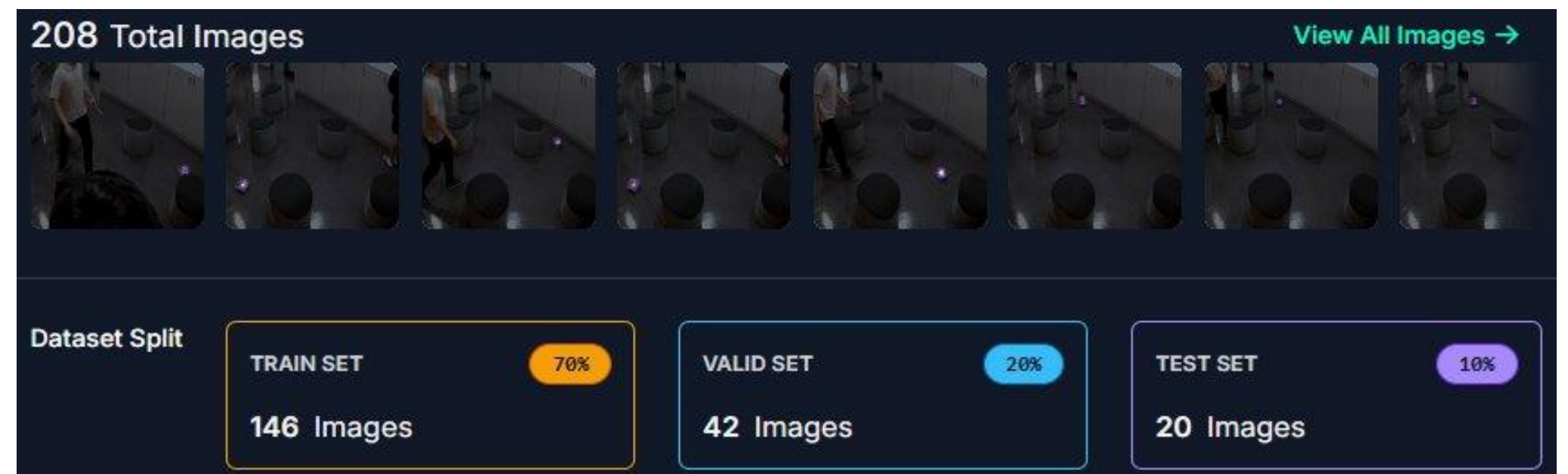
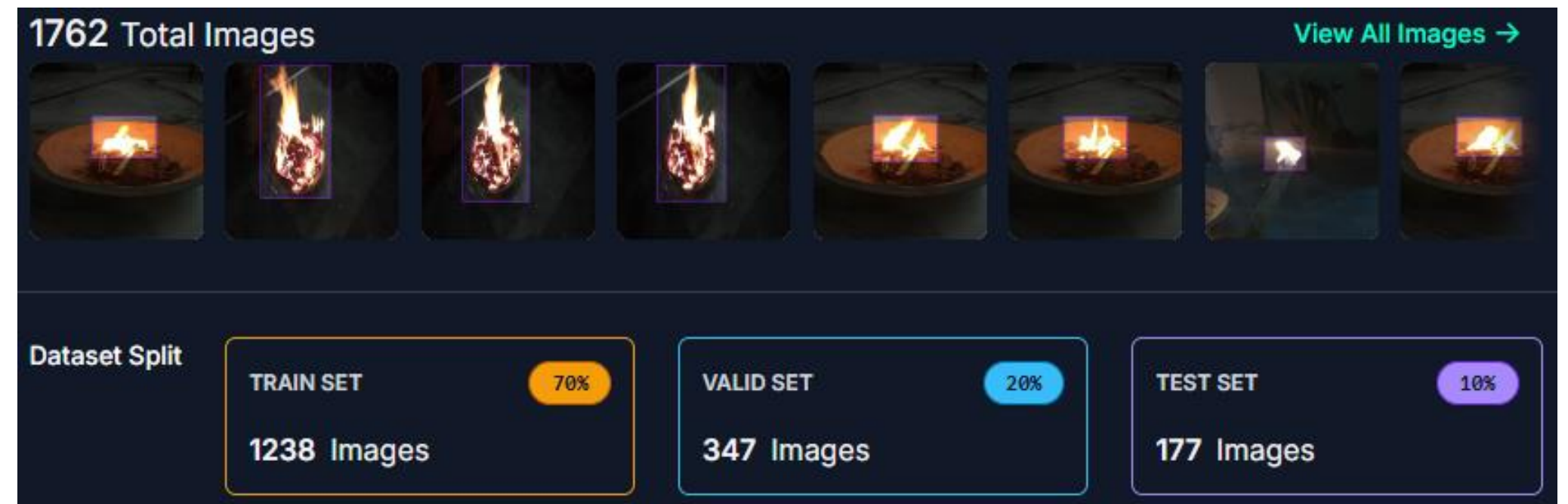


CCTV에 촬영한 불꽃을 데이터로 활용

Total : 208 image

- Train : 146 image
- Valid : 42 image
- Test : 20 image

Epochs : 300



258 epochs completed in 0.261 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.3MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.3MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.2.98 Python-3.10.12 torch-2.4.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3,005,843 parameters, 0 gradients, 8.1 GFLOPs

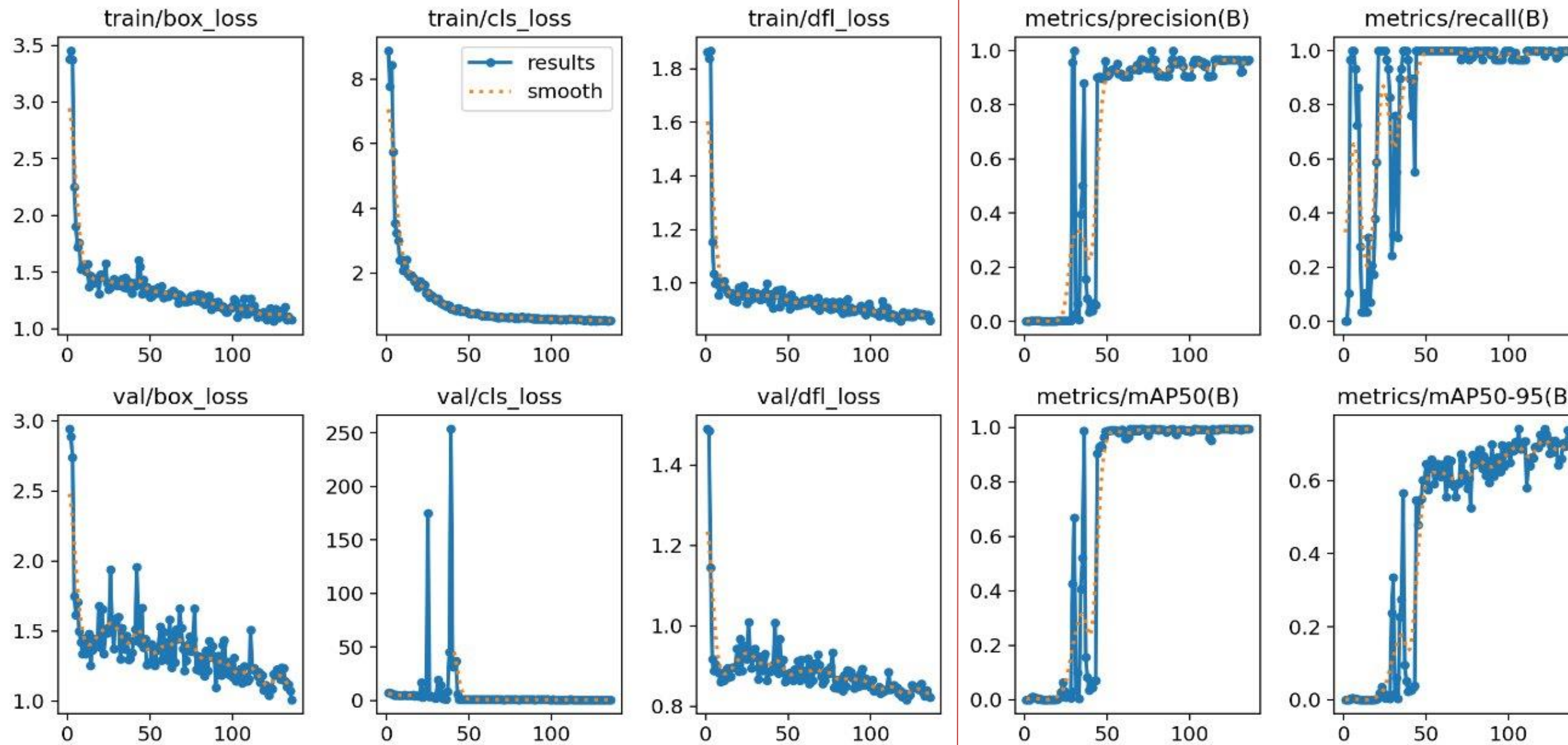
| Class | Images | Instances | Box(P) | R | mAP50 | mAP50-95 | 100% | 1/1 | [00:00<00:00, 2.94s] |
|-------|--------|-----------|--------|---|-------|----------|------|-----|----------------------|
| all | 42 | 42 | 0.998 | 1 | 0.995 | 0.66 | | | |

13/27

03. 작업과정

[Intel] 엣지 AI S/W 아카데미

CCTV 화재 학습



손실율 그래프

정확도 그래프

그래프를 살펴보면 전체적으로 모델의 성능이 학습과정에서 **손실은 감소**하고 **정확도는 꾸준히 상승**하는 지표를 보여주고 있어 학습이 효과적으로 진행되고 있는 걸 보여줌

03. 작업과정

[Intel] 엣지 AI S/W 아카데미

CCTV 화재 검출

화재 검출 전



화재 검출 후

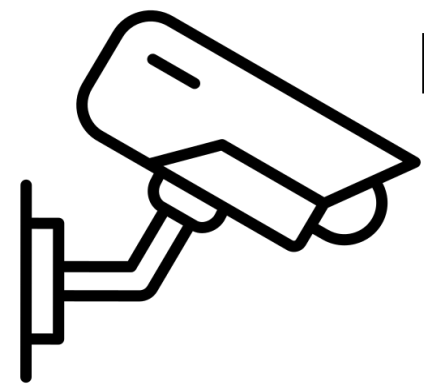


03.

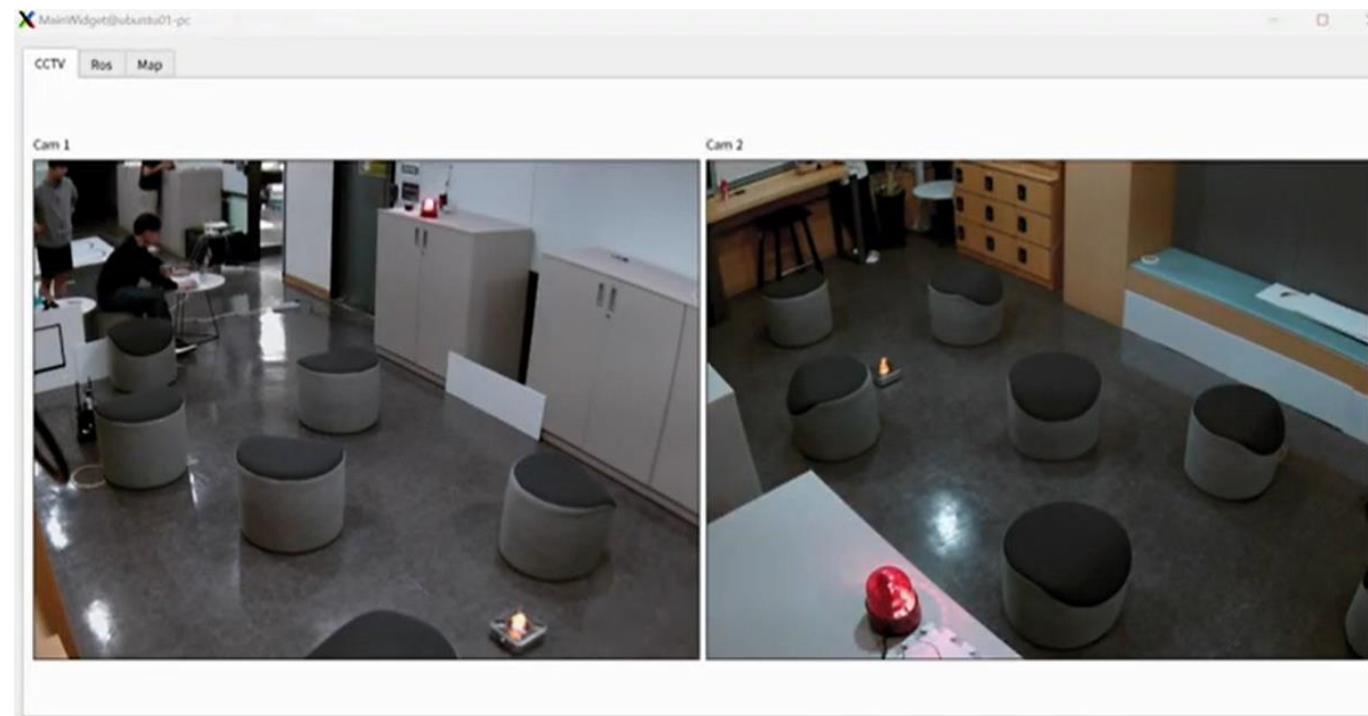
작업과정

[Intel] 엣지 AI S/W 아카데미

서버와 CCTV 통신 및 앱 연동

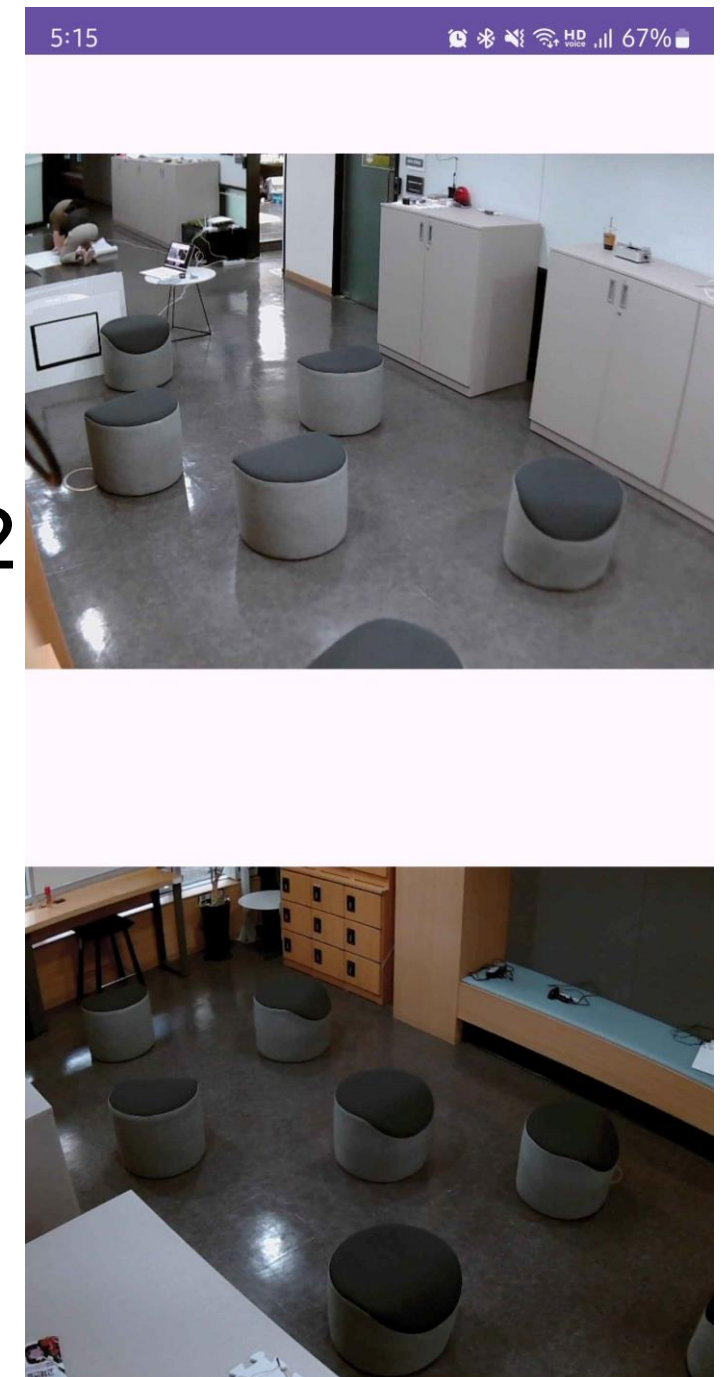


Port : 5000



Server

Port : 5002



TCP 소켓 통신을 통해 CCTV(Raspberry Pi)로부터 서버가 영상을 수신,
서버에서 해당 영상을 헤더와 함께 앱으로 재송신 및 OpenCV로 처리해 동시 송출 지원

03.

[Intel] 엣지 AI S/W 아카데미

작업과정

화재 알림 장치

- 모듈 적재 in Kernel

물리 주소 접근

```
#define BCM2711_PERI_BASE    0xFE000000
#define GPIO_BASE            (BCM2711_PERI_BASE + 0x2000000)

#define GPIO_PIN_BCM_BUZZER    17
#define GPIO_PIN_BCM_BUZZER_LED 27

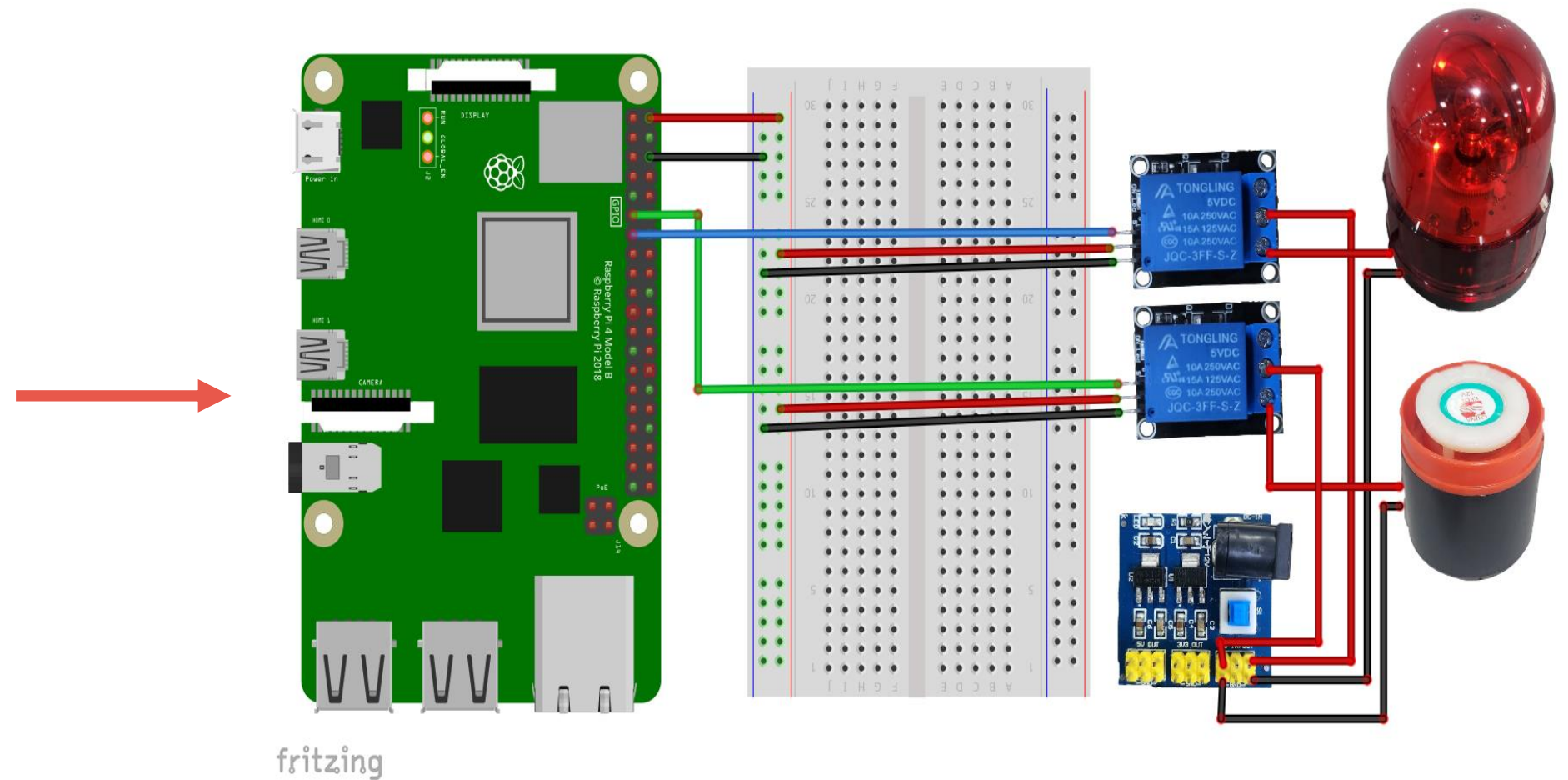
static volatile unsigned int *gpio;
```

메모리 매핑

```
printk("buzzer_dev: Init\n");
gpio = (unsigned int *)ioremap(GPIO_BASE, 0xB4);
```

GPIO 제어

```
// 인덱스 하나 -> 10개, 핀마다 3비트 -> 7
void set_gpio_input(int g) {
    unsigned int index = g / 10;
    unsigned int shift = (g % 10) * 3;
    gpio[index] &= ~(7 << shift);
}
```

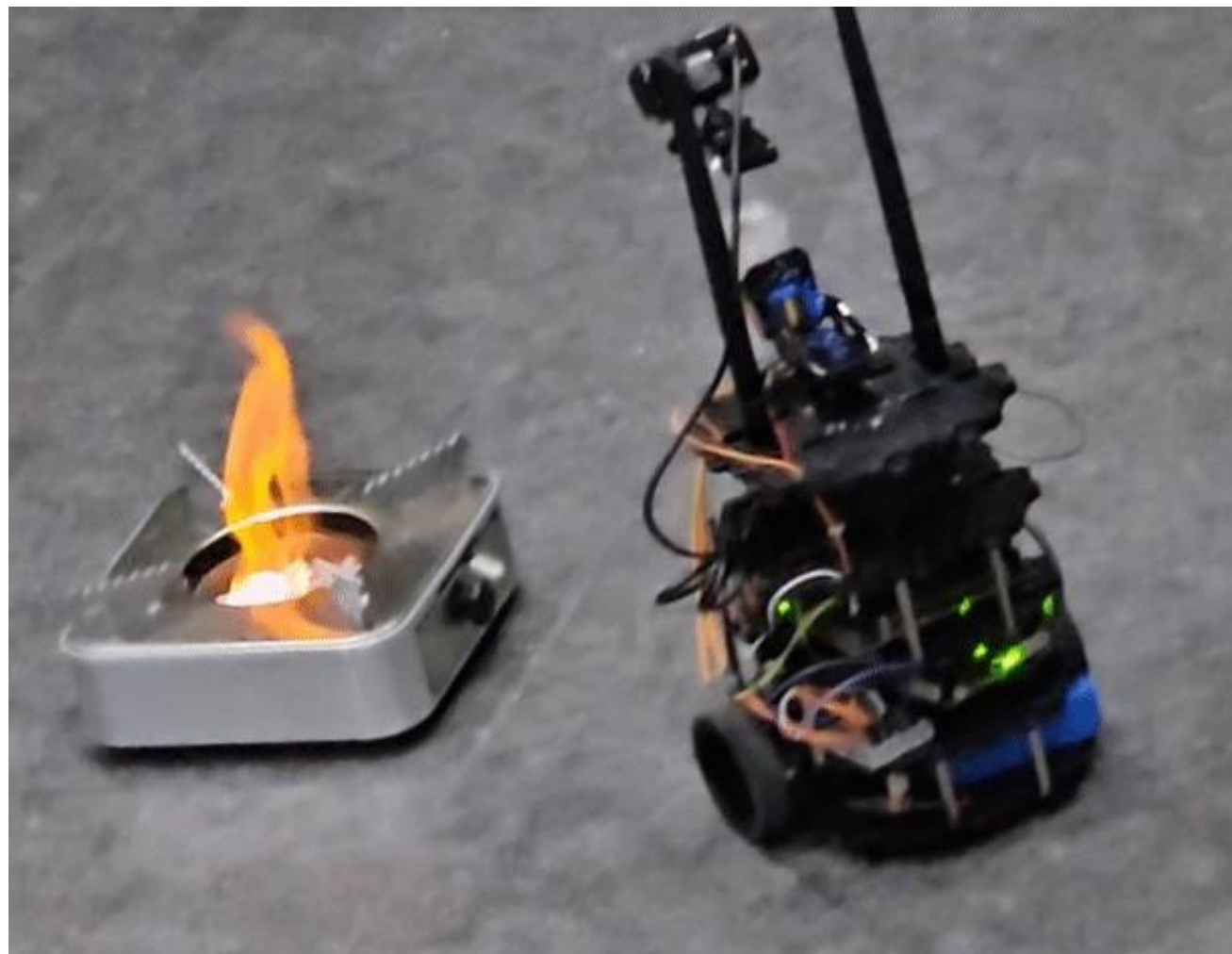


03.

작업과정

[Intel] 엣지 AI S/W 아카데미

소화장치 설계



```
int main(int argc, char* argv[]) {
    int pid_ai = 0;

    FILE *fp;
    char buf[6];

    bool fire_flag;
    int angle_x, angle_y;

    fp = fopen("ai_proce
fgets(buf, BUFSIZ, f
pid_ai = atoi(buf);
fclose(fp);

    fp = fopen("main_prc
itoa(getpid(), buf,
fputs(buf, fp);
fclose(fp);

    controlMotorAngle(0,
controlMotorAngle(2,
sendSignal(0);
controlMotorAngle(2,
sendSignal(1);
controlMotorAngle(2,
sendSignal(2);
waitCtl(0.3);

    workAI(&pid_ai);
    fire_flag = findPos(&angle_x, &angle_y);

    if (fire_flag) {
        controlMotorAngle(0, angle_y);
        controlMotorAngle(2, angle_x);
        toggleSpray(78);

        while (fire_flag) {
            waitCtl(10);
            break;
        }
        toggleSpray(78);
    }

    controlMotorAngle(0, 90);
    controlMotorAngle(2, 90);

    puts("main ended");

    return 0;
}
```


03. 작업과정

[Intel] 엣지 AI S/W 아카데미

소화장치 설계



03.

작업과정

[Intel] 엣지 AI S/W 아카데미

소화장치 설계- 문제 / 해결



화재 방향 확인

TCP / PWM



left.jpg



front.jpg



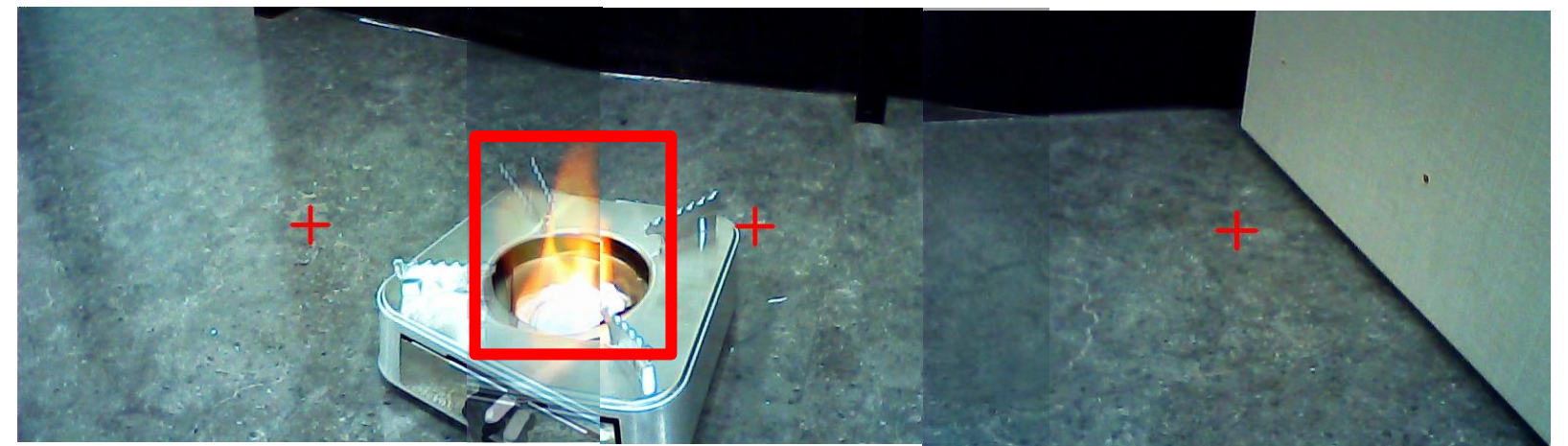
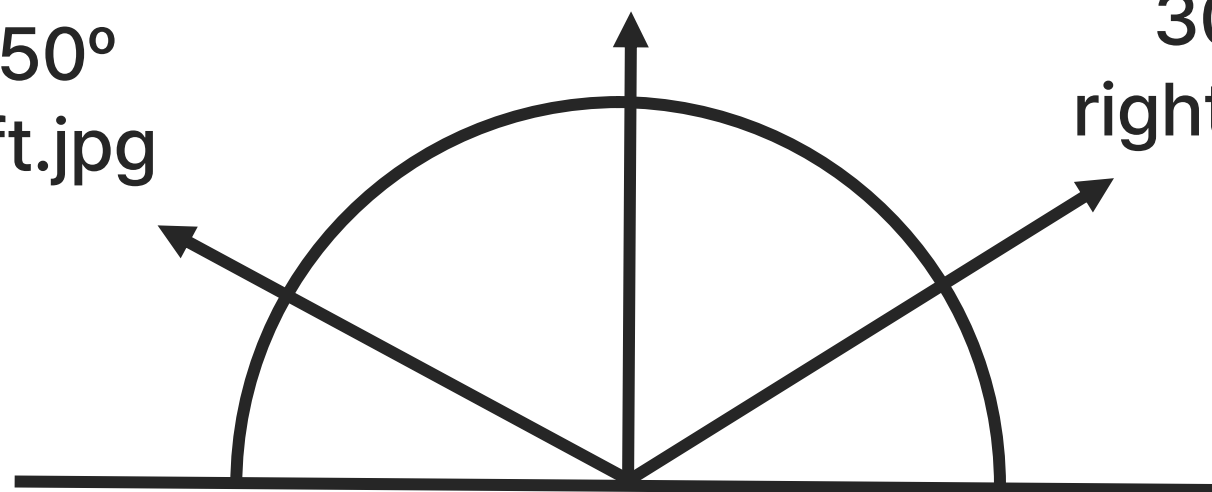
right.jpg



90°
front.jpg

150°
left.jpg

30°
right.jpg



03.

작업과정

[Intel] 엣지 AI S/W 아카데미

소화장치 설계- 문제 / 해결



AI 화재 감지
signal

AI를 별도의 프로그램화

```
jetson@nano21:~/fin_proj$ time python3 detectPic.py
0: 640x640 1 fire 82.2ms
Speed: 26.8ms preprocess, 82.2ms inference, 10.8ms
Saved detection result to img/left.txt

0: 480x640 (no detections) 66.0ms
Speed: 4.5ms preprocess, 66.0ms inference, 2.2ms
Saved detection result to img/left.txt

0: 480x640 (no detections), 62.9ms
Speed: 4.4ms preprocess, 62.9ms inference, 2.2ms
Saved detection result to img/left.txt

real    0m19.666s
user    0m14.436s
sys     0m4.456s
```

AI 프로그램

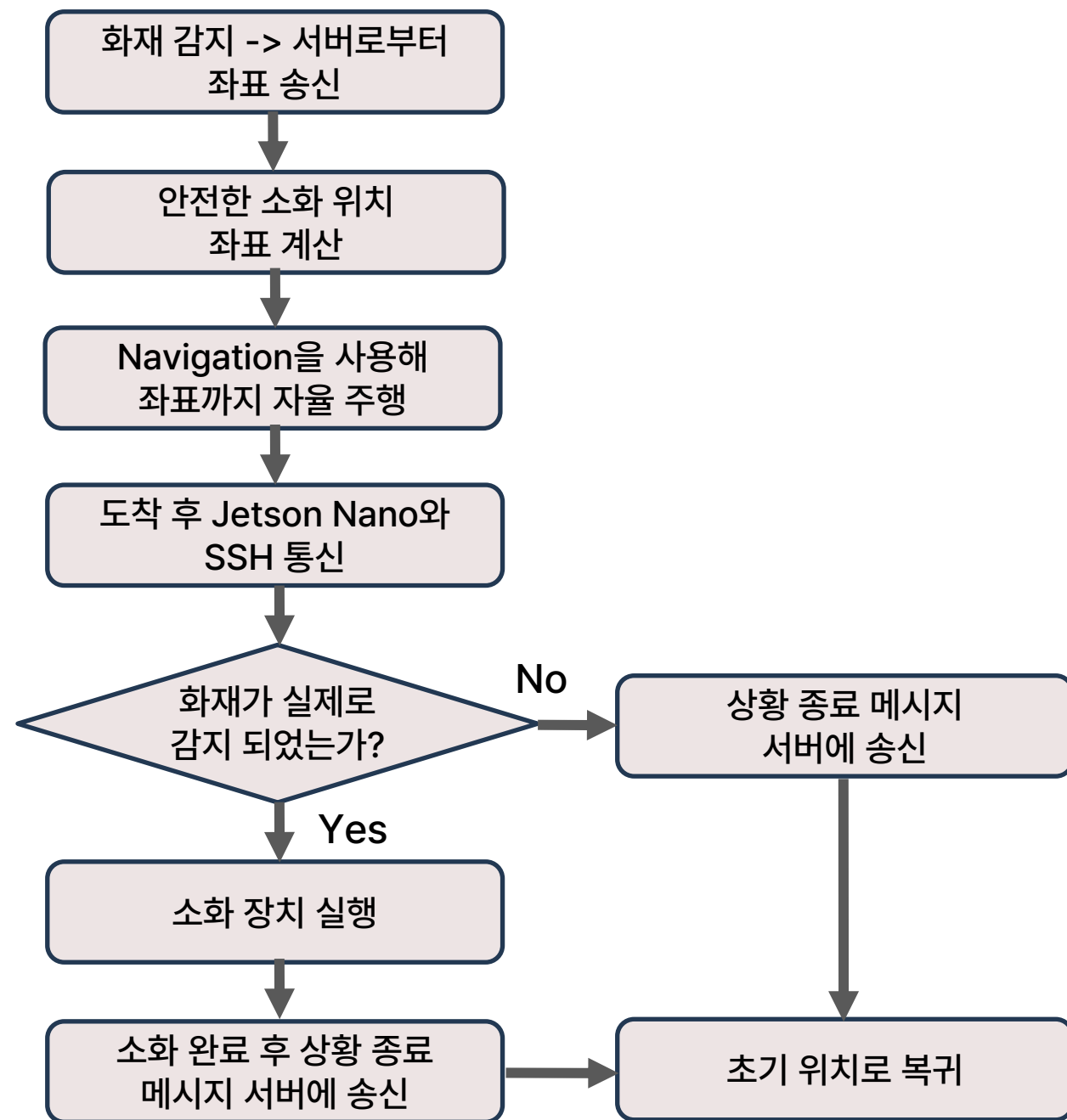
사진 탐지 요청

결과 저장 완료

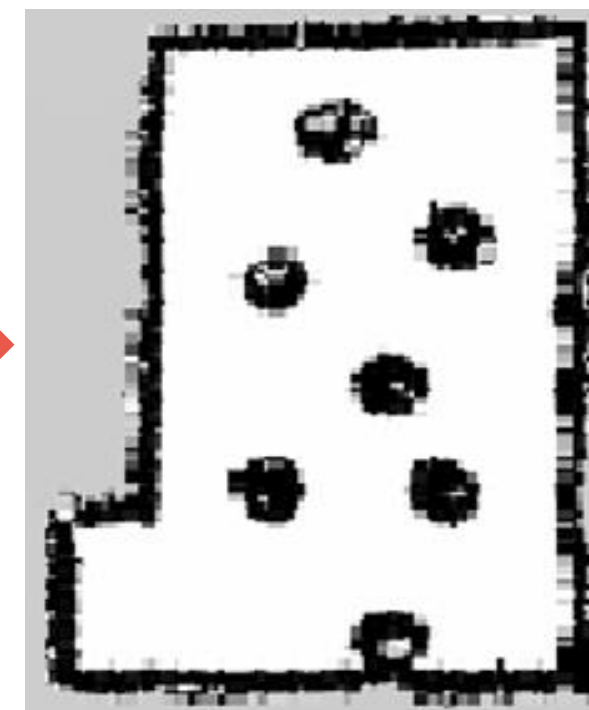
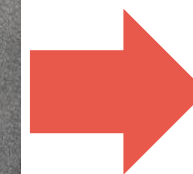
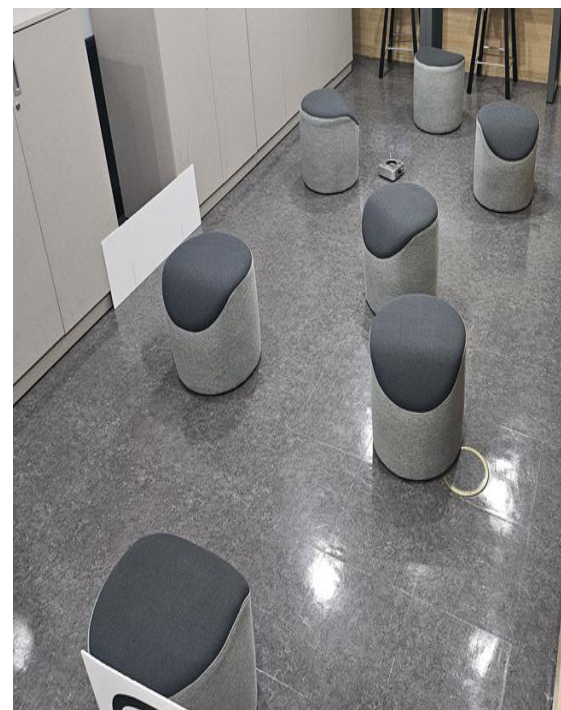
소화 기능 프로세스

작업과정

터틀봇 자율주행 및 소화장치 실행



서버를 거치지 않고 자체적으로 화재 인식을 하기 위해 Raspberry Pi 보다 성능이 좋은 **Jetson Nano**를 연결한 터틀봇에 소화장치를 결합



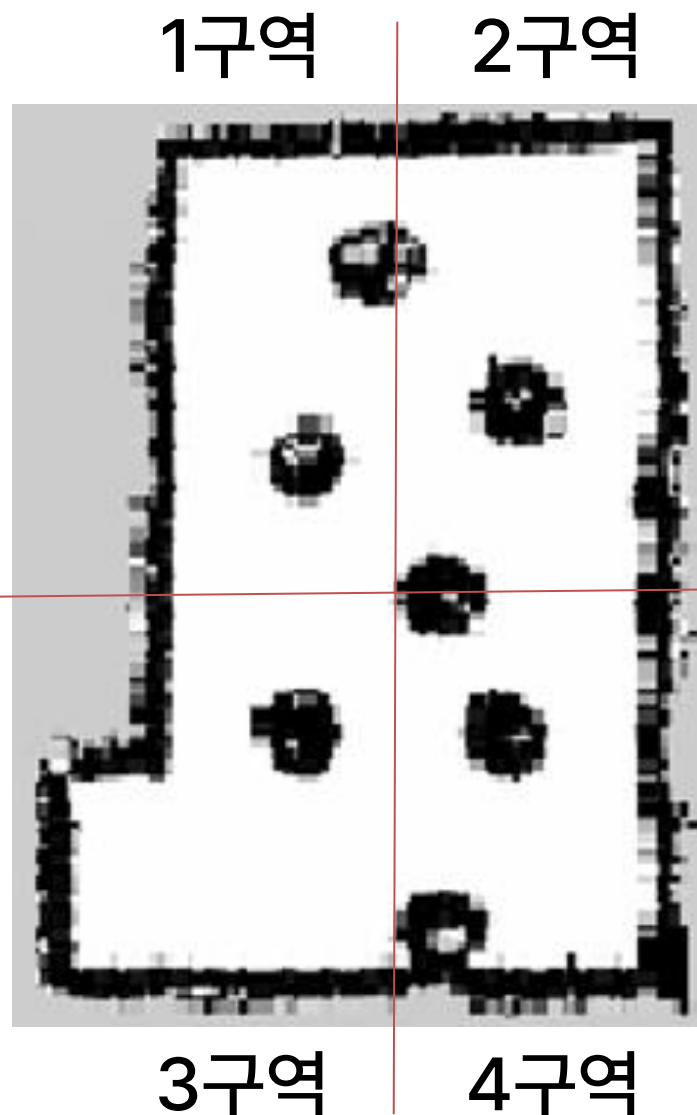
터틀봇의 **LiDAR**와 **ROS**의 **SLAM**을 이용하여 **Map** 제작

03.

[Intel] 엣지 AI S/W 아카데미

작업과정

터틀봇 자율주행 및 소화장치 실행



서버로부터 수신 받은 좌표는 화재의 한가운데 좌표이기 때문에 **안전한 소화를 위한 좌표 보정** 필요

GOAL@x@y 형태로 수신 받은 좌표 값을 파싱하여 추출한 후 구역별로 다른 조건으로 보정

<조건>

■ 1구역

$z = 0.94$ $w = 0.33$ / $x \rightarrow 0.3$ $y \rightarrow -0.22$

■ 2구역

$z = 0.34$ $w = 0.94$ / $x \rightarrow -0.3$ $y \rightarrow -0.22$

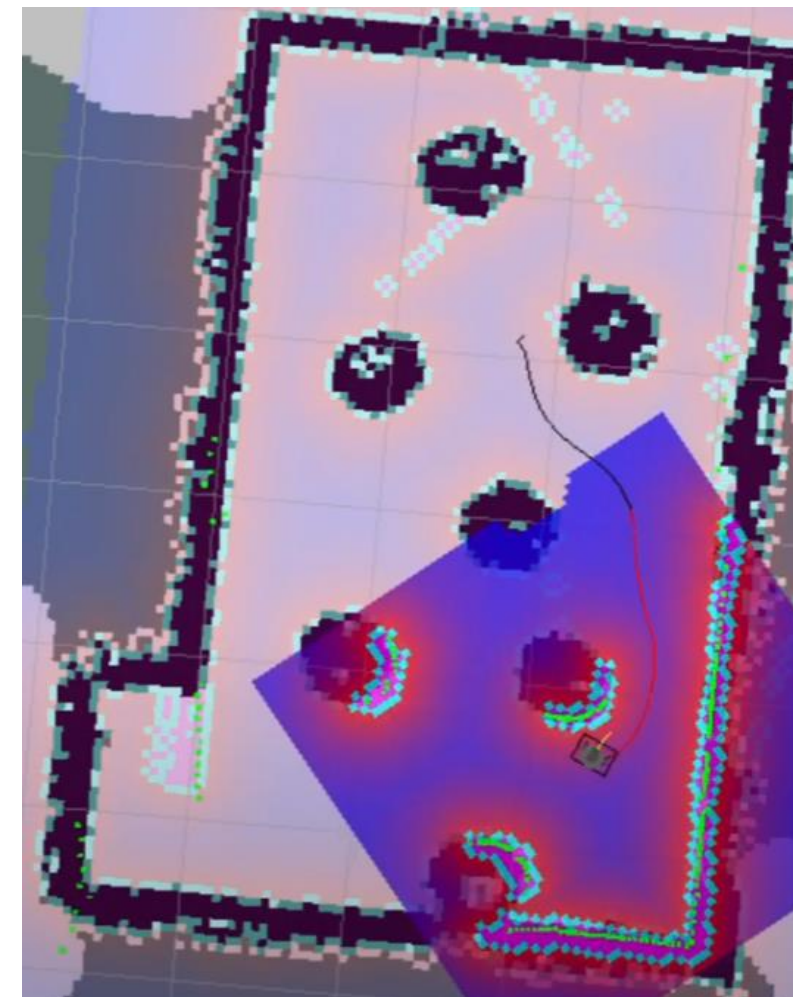
■ 3구역

$z = 0.94$ $w = 0.33$ / $x \rightarrow 0.3$ $y \rightarrow -0.22$

■ 4구역

$z = 0.94$ $w = 0.33$ / $x \rightarrow 0.3$ $y \rightarrow -0.22$

```
move_base_msgs::MoveBaseGoal goal;
goal.target_pose.header.frame_id = "map";
goal.target_pose.header.stamp = ros::Time::now();
goal.target_pose.pose.position.x = x;
goal.target_pose.pose.position.y = y;
goal.target_pose.pose.orientation.z = z;
goal.target_pose.pose.orientation.w = w;
```



ROS의 **Move_base** 기능과 **Navigation**을 사용하여 보정된 좌표 값으로 장애물을 피하며 이동

도착 후 **SSH**로 Jetson Nano에 접속하여 소화 장치 코드 실행하여 실제 화재 여부 판별 후 소화 진행

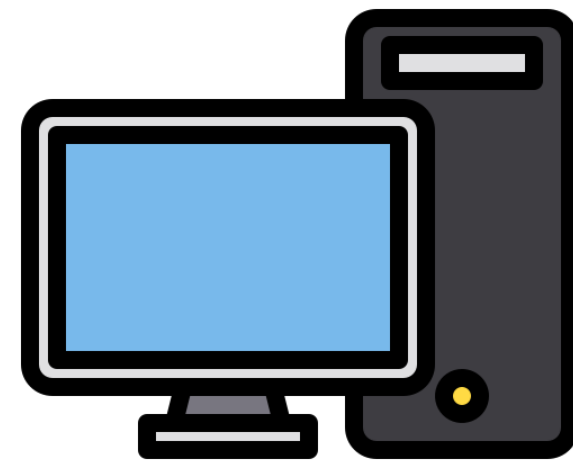
소화 완료 시 완료 메시지를 서버에 송신 후 초기 위치로 복귀

03.

작업과정

[Intel] 엣지 AI S/W 아카데미

터틀봇 자율주행 및 소화장치 실행 - 문제 / 해결

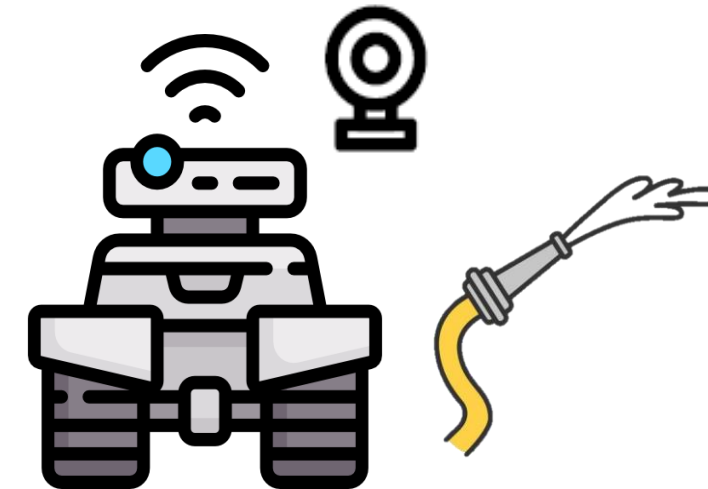


터틀봇 주행 코드 실행

SSH
프로그램 실행 명령어



소화 장치 코드 실행 값 미러링



소화 장치 코드 실행

서버(PC)에서 실행하는 주행 코드와
Jetson Nano(터틀봇)에서
실행되는 소화장치 코드의 통합 문제



서버에서 실행되는 코드에 서버와 Jetson Nano 간의 **SSH 통신** 기능을
삽입하여, 소화 관련 기능을 실행하는 명령어를 전달하도록 구성.
이를 통해 터틀봇 실행 시 AI 모델 로딩 스레드, 화재 감지 스레드,
소화 기능 스레드를 순차적으로 실행할 수 있게 함.

```
// SSH 명령어 실행 (소화 감지 시스템 시작)
system("ssh jetson@192.168.100.108 'cd ~/final_code && ./init.sh
&& nohup python3 ai/detectPic.py & sleep 2 && exit'");
return NULL;
```

```
0: 480x640 (no detections), 63.4ms
Speed: 11.9ms preprocess, 63.4ms inference, 2.7ms postprocess per image at shape (1, 3, 480, 640)

0: 480x640 (no detections), 63.2ms
Speed: 5.3ms preprocess, 63.2ms inference, 2.3ms postprocess per image at shape (1, 3, 480, 640)

0: 480x640 2 fires, 63.1ms
Speed: 5.1ms preprocess, 63.1ms inference, 152.4ms postprocess per image at shape (1, 3, 480, 640)
[ INFO] [1727085941.741262530]: Fire extinguishing program completed. Returning to the start location
```

04. 시연영상

[Intel] 엣지 AI S/W 아카데미



[Intel] 엣지 AI S/W 아카데미

|

AIoT 프로젝트

미니 소방 로봇

2조 불나방

권오준 박민혁 유광균 이윤혁 임지원 장혜원

자체 평가 및 소감

| 이름 | 평가 및 소감 |
|-----|---|
| 권오준 | 이번 교육 과정 동안 학부에서 배우지 않은 펌웨어 프로그래밍, AI 학습 등에 대한 지식을 습득해 유익했고, 다른 교육생들과 여러 프로젝트를 진행하며 협업하는 방법을 배울 수 있어 좋았습니다. 이번 프로젝트에서 팀장을 맡으면서 의사소통, 업무 배분, 상황 판단에 대한 중요성을 느꼈고, 미숙했던 부분을 되돌아보며 보완하고자 합니다. |
| 유광균 | 퇴사 후 지식의 부족함을 느껴 센터에 입학해 다양한 프로젝트를 하면서 임베디드 개발과 AI 모델 학습을 실무적으로 배울 수 있었습니다. 프로젝트 진행 중 처음 학습한 모델의 정확도가 낮은 문제가 있었고, 팀원들과 의견을 주고받으며 데이터셋을 분석하고 필요한 데이터를 추가해 정확도를 높일 수 있었습니다. 이 과정에서 협업을 통한 문제 해결 능력을 키울 수 있었습니다. |
| 임지원 | 이번 교육을 통해 SW 지식을 리마인드할 뿐만 아니라 잘 모르던 HW 지식에 대한 폭을 넓힐 수 있었습니다. 여러 프로젝트를 진행 하면서 단순히 기술을 적용하는 데 그치지 않고, 최적화된 결과물을 얻기 위해 성능을 지속적으로 개선하고자 노력하는 습관을 가지게 되었습니다. |
| 박민혁 | 교육 과정을 이수하면서, 다양한 프로젝트를 진행하였습니다. 여러 분야를 배웠지만, 그중에서도 특히 HW와 SW를 이어주는 연결 다리와 같은 디바이스 드라이버라는 아름답고, 흥미있는 분야를 찾게 되었습니다. 많이 배우며 의미있는 시간이었습니다. |
| 장혜원 | 교육과정 동안에 학부에서 시도해보지 않았던 하드웨어와 펌웨어에 대해서 배워, 기존의 프로젝트와 다르게 최종프로젝트에서는 관련 파트를 담당할 수 있을 만큼 성장할 수 있는 과정이었습니다. |
| 이윤혁 | 교육을 통해 학부 시절 접하지 못했던 다양한 임베디드에 관련된 기술을 실무적으로 경험할 수 있어 매우 유익했습니다. 또한, 프로젝트를 진행하면서 협업과 의사소통의 중요성을 실감하며 매우 의미 있는 시간을 보낼 수 있었습니다. |

감사합니다
