

2024 ICT 융합 프로젝트 공모전

AI를 활용한 산불 감지 수막 설비 시스템

김민환, 이병찬, 한민규

[목 차]

I. 서론

1. 개요	3
(1) 개발 배경	3
(2) 개발 목표	4
(3) 작품 개요	4
(4) 개발 일정	5

II. 본론

1. 개발 환경 및 제품 구성	6
(1) 개발 환경	6
(2) YOLO v5	6
(3) 부품 리스트	11
2. 작품 구성	13
(1) 회로도 구성	13
(2) 시스템 구성도	14
(3) 설비 시스템 외관	15
3. 주요 동작 및 특성	15
(1) 전체 시스템 흐름도	15
(2) Arduino Sensor	16
(3) Jetson Nano	18
(4) Arduino Actuator	20
(5) Raspberry PI	21
4. 구현 결과	22

III. 결론

1. 향후 목표	23
2. 기대 효과	23

IV. 참고 자료

1. 참고 문헌	25
2. 소스 코드	25

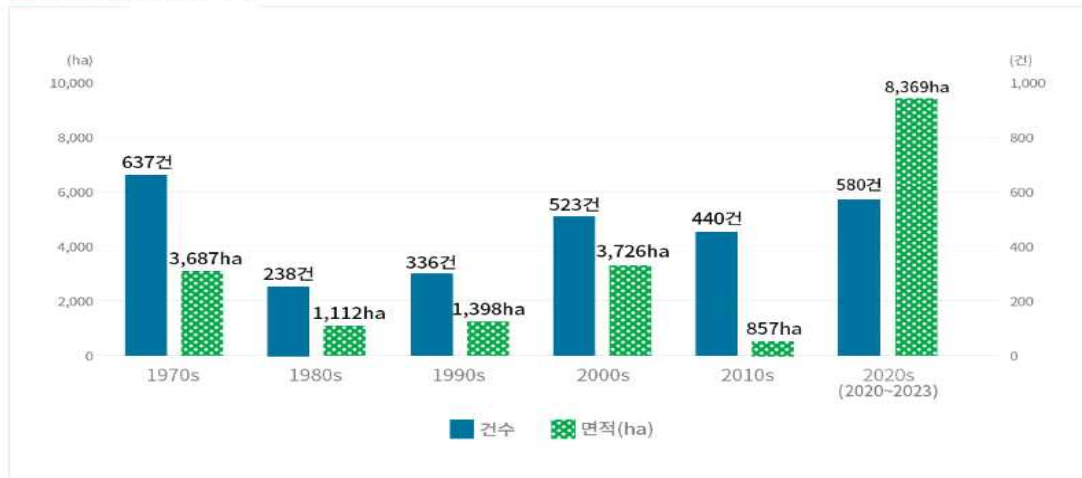
I. 서론

1. 개요

(1) 개발 배경

전 세계적인 기후 변화로 인해 산불의 발생 빈도와 규모가 커지는 추세이다. 우리나라의 경우 2023년 한해에만 크고 작은 산불이 596건 발생하여 4,992ha의 산림이 소실 되었다. 여기서 주목해야 할 부분은 산불이 주로 빠른 대피가 힘든 고령층의 인명피해를 가증시키고 있다는 점이다.

10년간 산불발생 현황



지난해 4월, 대전과 충남 홍성 산불 당시 노인 거주 주택과 시설까지 번진 불을 목격한 소방관들은 이러한 사례를 반복하지 않기 위해 주민들의 대피 시간을 벌 수 있는 수막 설비를 직접 개발하여 산자락에 자리 잡은 주택 바로 옆에 설치했고 산불 발생 시 주민들이 수동으로 밸브를 열어 산을 향해 물을 분사해 불이 근접해 와도 주변의 땅과 주택에 쉽게 불이 붙지 못하도록 하였다.

이 설비는 산림 인접 주택과 노인 요양 시설을 시작으로 점차 확대 보급될 예정이지만 맨눈으로 감지하는 방법이어서 산불 발견이 힘들고 주민 부재 시에 대응하기 힘들다는 문제점이 있다. 이에 따라 일선 소방관들과 산간지역 주민들을 중심으로 이러한 고전적인 방법에서 벗어나 IoT 센서, AI 등 다양한 첨단기술을 이용한 화재감지 기술을 활용한 선진적인 재난 관리 시스템이 필요하다는 의견이 대두되고 있다.



(2) 개발 목표

<문제점>

- 기후 변화와 함께 산불이 발생하는 빈도와 규모가 증가하고 있다.
- 산불은 시야 확보가 어려운 산림지역에 주로 발생하여 초기 발견이 어렵다.
- 산불 확산 시 거동이 불편한 고령자와 취약 계층은 빠른 대피가 어렵다.

<개선점>

- 산불 발생 시 초기에 감지 및 진화하여 산불 확산을 사전에 방지한다.
- 맨눈으로는 산불 발견이 어려운 만큼 IoT 센서와 딥러닝(Deep Learning)으로 학습된 카메라를 통해 이중으로 감지한다.
- 자동화된 시스템을 통해 산불이 마을로 근접하는 경우 주민들의 대피 시간을 확보할 수 있고 물적 피해 또한 줄일 수 있다.

(3) 작품 개요

적외선, 연기 감지가 가능한 센서 장치를 마을 근처 산자락에 복수 배치하여 산불을 감지하고 와이파이 통신으로 감지된 센서들의 값과 위치 정보를 서버에 전송한다. 소방당국이 웹을 통해 실시간 정보와 GPS 위치를 확인할 수 있도록 하고 센서에 임계값 이상의 데이터가 들어오는 경우 카메라를 활성화한다. 이후, 산불을 인식한다면 수막 설비를 통해 산불의 근접을 막고 경보장치를 가동해 주민들에게 위험을 알린다.

(4) 개발 일정

AI를 활용한 산불 감지 수막 설비 개발 일정

분류	작업	3/1~3/5	3/6~3/10	3/11~3/14	3/15~3/17	3/18~3/21	3/22~3/24	3/25~3/28	3/29~3/30
프로그램 구성 및 착수	주제 선정 및 계획 수립								
	센서 및 보드 조사								
개발 환경 셋팅	Jetson Nano 및 OpenCV 개발 환경 셋팅								
	YOLOv5를 이용한 사물/사람 Detect Test								
	raspberry Pi 및 MariaDB 개발 환경 셋팅								
	arduino 보드 및 센서 셋팅 및 동작 확인								
AI 모델 및 영상 처리 구현	영상/사진 데이터 수집								
	Roboflow를 이용하여 데이터 셋 만들기(전처리)								
	모델 학습								
	학습된 모델 Jetson Nano에 적용 및 최종 확인								
제어 시스템 개발	동작 및 센서 부 개발								
	회로 배치 및 외관 제작								
제어 시스템 통합	통신 서버 구축								
	데이터베이스 생성								
	제어 시스템 통합								
	웹페이지 제작								
최종 종합 검증	제품 구동 및 최종 확인								
	프로젝트 보고서 작성								

II. 본론

1. 개발 환경 및 제품 구성

(1) 개발 환경

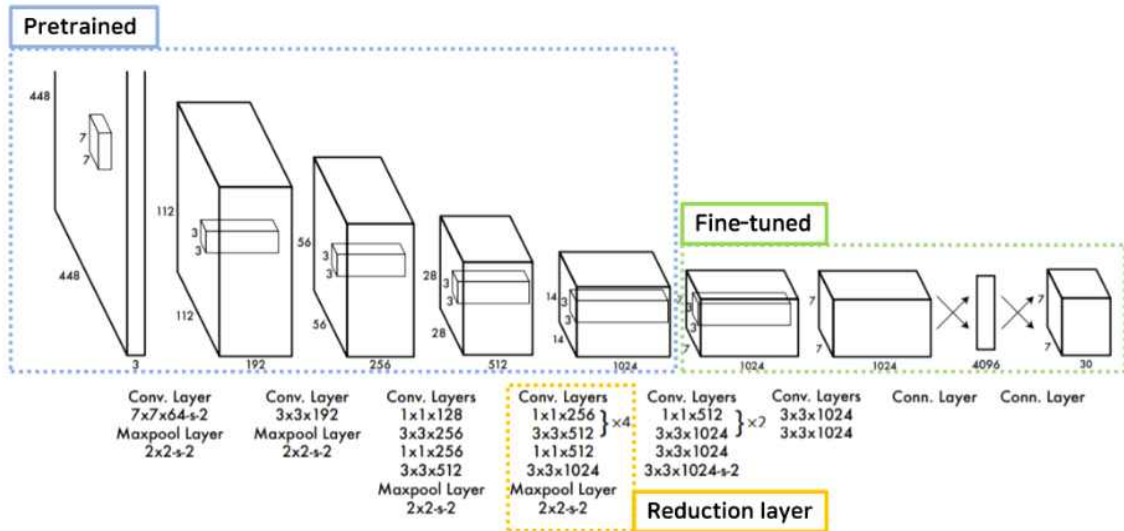
Board	Arduino	Jetson Nano	Raspberry Pi
Language			 
OS			 
Library	DHT, SerialSoftware , WiFiEsp , TinyGPS , I2C		sys/socket , sys/types , stdio , pthread , netinet/in , sys/stat , arpa/inet
System			  

(2) YOLO v5

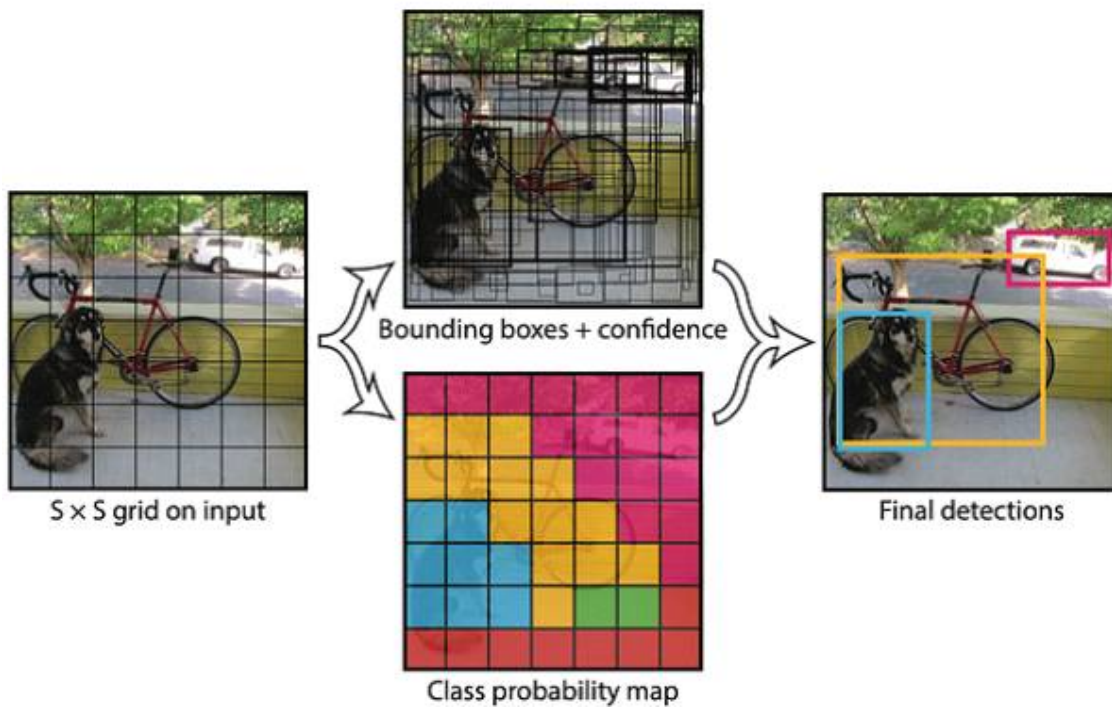
-1. 특징

YOLO는 실시간 객체 감지를 위한 딥러닝 모델로, 다음과 같은 특징이 있다.

- ① 이미지 전체를 한 번만 본다. 이전 R-CNN 계열의 방식처럼 이미지를 여러 장 나눠서 여러 번 분석하는 일을 하지 않는다는 의미로 YOLO는 원본 이미지 그대로 CNN에 통과한다.
- ② 통합된 모델을 사용하기 때문에, 다른 모델들에 비해서 속도가 빠르다. 이러한 특징으로 실시간 객체 탐지를 할 수 있다.
- ③ 주변 정보까지 학습하여 이미지 전체를 처리하기 때문에 배경에 의한 오류가 적다.
- ④ 훈련 단계에서 보지 못한 새로운 이미지에 대해서도 검출 정확도가 높은 특징이 있다.



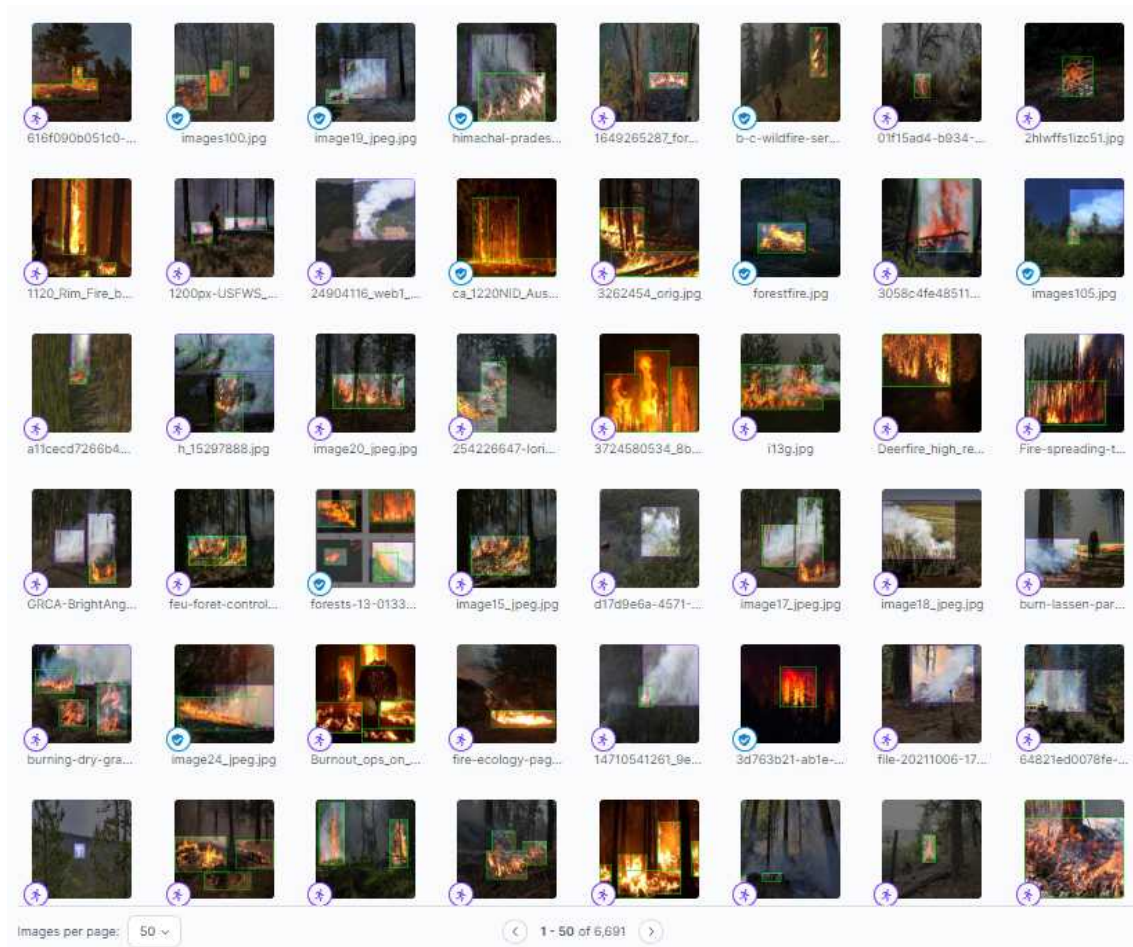
-2. 동작 과정



YOLO는 사진이 입력되면 가로-세로 동일한 그리드 영역을 나눈다. 그 후 각 그리드 영역에 대해서 어디에 사물이 존재하는지 Bounding Box와 Box에 대한 신뢰도 점수를 예측한다. 신뢰도가 높을수록 굵게 박스를 그려주고 어떤 사물인지에 대한 classification 작업이 동시에 진행되어 굵은 박스들만 남기고 얇은 것은 지워 준다.

-3. 데이터 세트(Roboflow)

roboflow는 객체 감지, 분류 및 세분화를 위해 이미지에 라벨을 지정하는 데 사용할 수 있는 온라인 주석 도구이다. roboflow에서 산불 이미지를 수집하여 6691장으로 클래스와 라벨링을 수정해 학습시켰다.



-4. 학습

1) 학습 과정

학습은 Colab을 활용해서 아래와 같은 코드를 작성하여 학습을 완료하였다.

```
!python train.py --img 416 --batch 16 --epochs 80 --data /content/forest-fire-response-system-1/data.yaml  
--cfg ./models/yolov5s.yaml --weights yolov5s.pt --name results
```

```
nc: 2  
names: ['fire', 'smoke']  
  
test: D:/device/train_y5/forest-fire-response-system-1/test/images  
train: D:/device/train_y5/forest-fire-response-system-1/train/images  
val: D:/device/train_y5/forest-fire-response-system-1/valid/images
```

학습 코드에 관한 내용은 아래와 같다.

img 416(416 * 416)

- img size를 줄이면 OOM Error가 줄어들지만, 성능이 저하 될 수 있음
- oom error를 줄이고 싶었고 img가 너무 작아도 안 됐기에 416으로 설정
- img size는 임의로 변경할 수 있고 특정 숫자 배수로 된 입력 Size여야 하기 때문이다.

batch 16(batch 16이 표준)

- 학습을 1번 반복할 때 사용되는 샘플 세트이다. 예를 들어 이미지가 80개가 있고 배치 크기가 16이라고 가정한다면 데이터가 배치 5번($80/16=5$)으로 나뉘진다는 의미이다.

epochs 80(전체 데이터 세트를 80번 학습)

- 학습 횟수를 의미하며 1 epochs는 학습 데이터 세트가 학습 모델을 통해 한 번 학습한 것이다.

2) 학습 결과

모델을 학습한 결과를 검증해본 결과 아래와 같이 나오게 되었다.

* 아래는 데이터 세트 중 test 세트를 활용해서 검증한 결과이다.



* 아래는 데이터 세트에 포함되지 않는 영상을 찍은 결과이다.



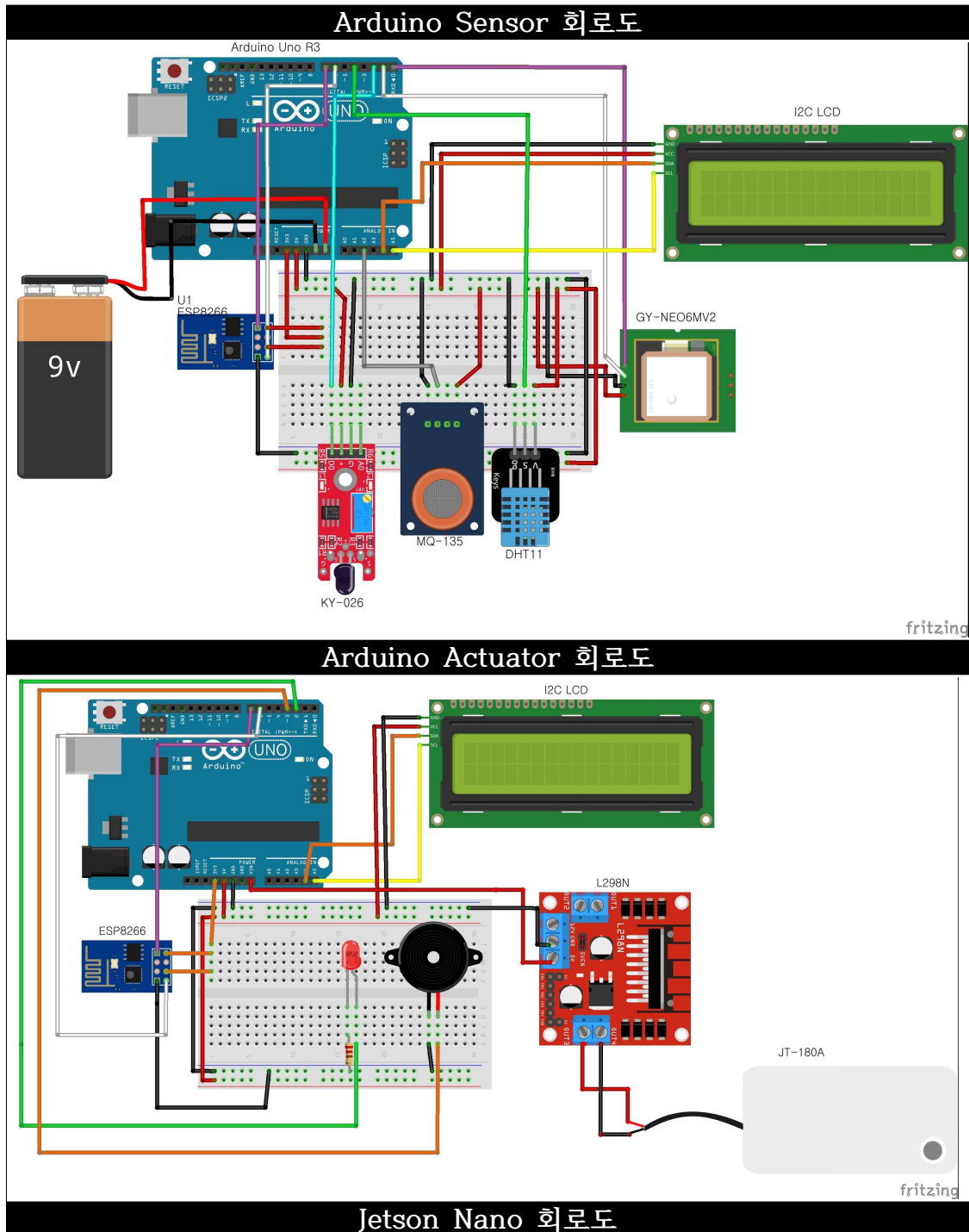
(3) 부품 리스트

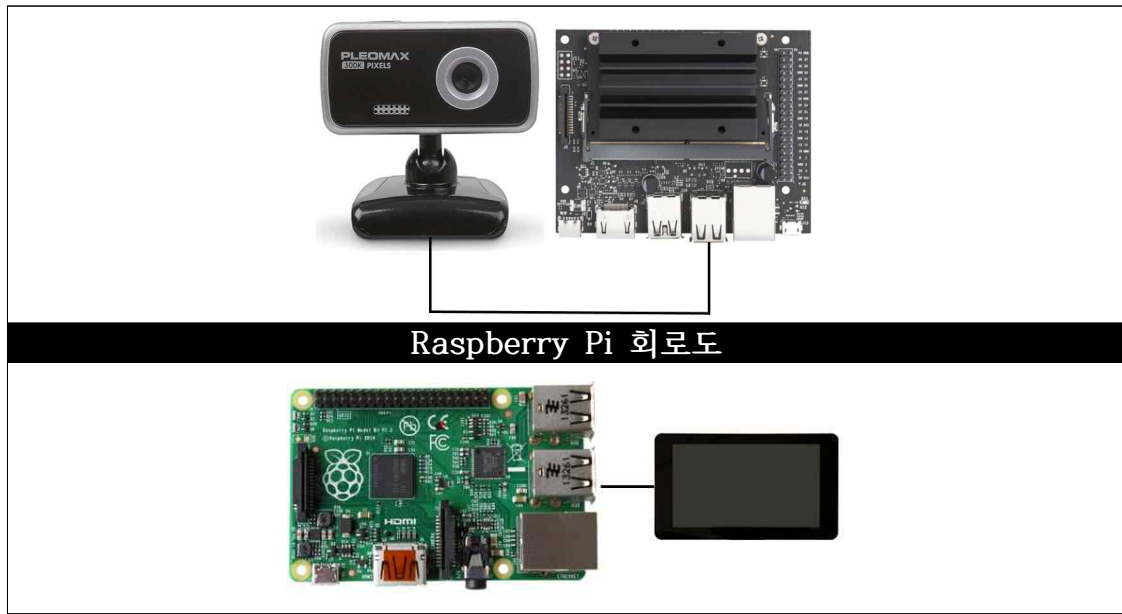
기능	부품명	이미지	링크	상품 번호
보드	아두이노 우노 R3 호환보드		https://www.devicemart.co.kr/goods/view?no=1245596	1245596
	Raspberry Pi 4 Model B		https://www.devicemart.co.kr/goods/view?no=12234534	12234534
	Jetson Nano		https://www.devicemart.co.kr/goods/view?no=12513656	12513656
센서	불꽃 감지 적외선 센서 모듈		https://www.devicemart.co.kr/goods/view?no=1385147	1385147
	MQ-135		https://www.devicemart.co.kr/goods/view?no=1327411	1327411
	DHT11		https://www.devicemart.co.kr/goods/view?no=1358495	1358495
	GPS모듈		https://www.devicemart.co.kr/goods/view?no=1321968	1321968
동작	방수 펌프(JT-180A)		http://itempage3.auction.co.kr/DetailView.aspx?ItemNo=E289968410&frm3=V	개별 구매

			2	
	모터 드라이버 모듈 (2A L298)		https://www.devicemart.co.kr/goods/view?no=1278835	1278835
	피에조 부저 (IMT12D2001AP)		https://www.devicemart.co.kr/goods/view?no=2733	2733
통신	10파이 라운드 LED		https://www.devicemart.co.kr/goods/view?no=12234077	12234077
	ESP8266 시리얼 와이파이 모듈		https://www.devicemart.co.kr/goods/view?no=1279338	1279338
	영상			
	플레오맥스 웹캠 (W-210)		https://item.gmarket.co.kr/Item?goodscode=1928217231	개별 구매
부속	아두이노 I2C LCD모듈		https://www.devicemart.co.kr/goods/view?no=1327456	1327456
	Raspberry Pi 터치스크린		https://www.devicemart.co.kr/goods/view?no=1273487	1273487
	강하형 DC-DC 다중 출력 컨버터		https://www.devicemart.co.kr/goods/view?no=1321159	1321159

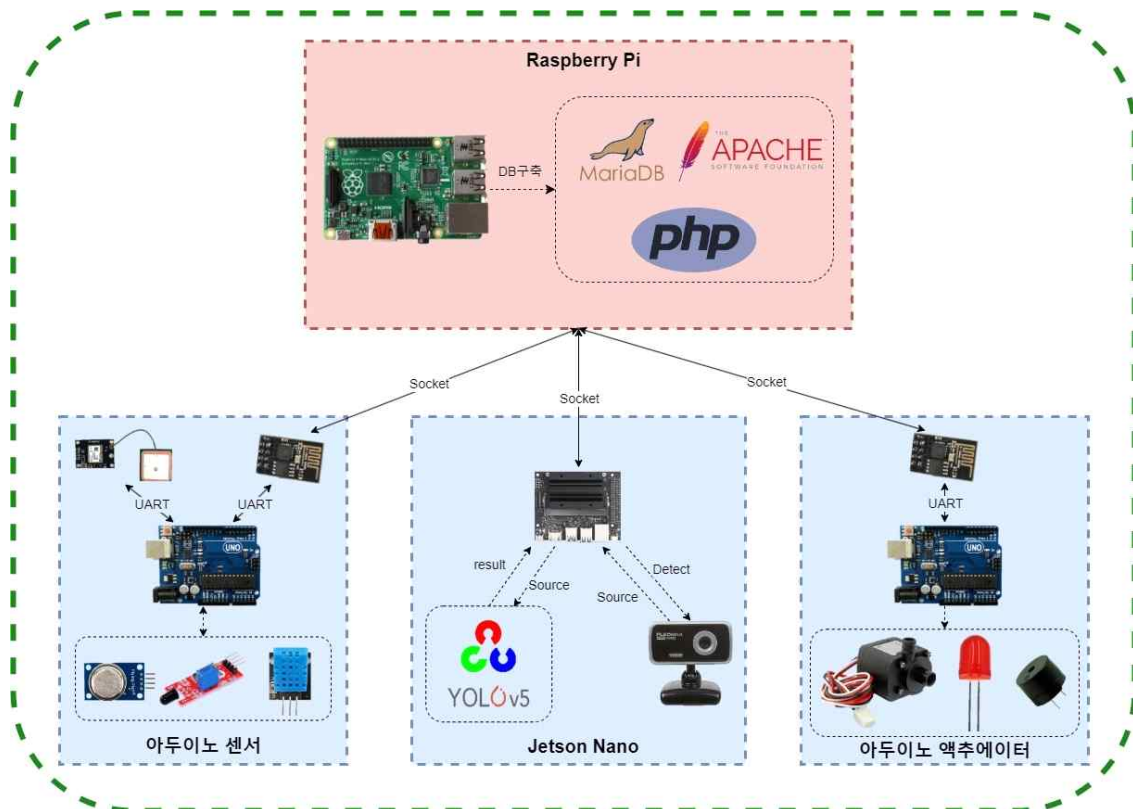
2. 작품 구성

(1) 회로도 구성

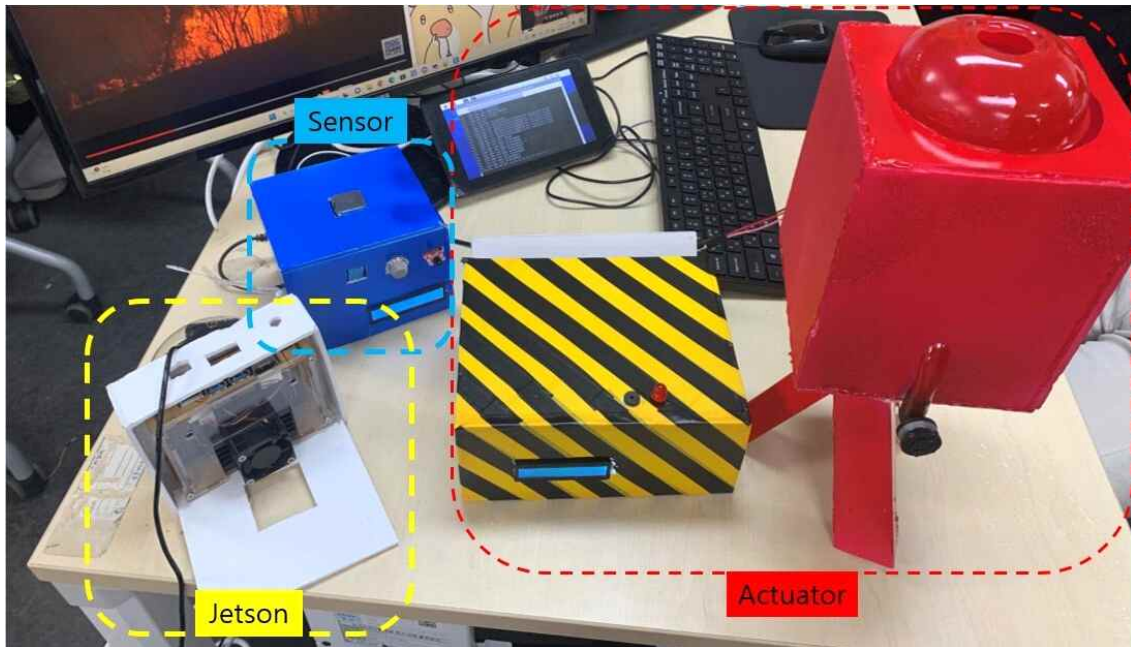




(2) 시스템 구성도

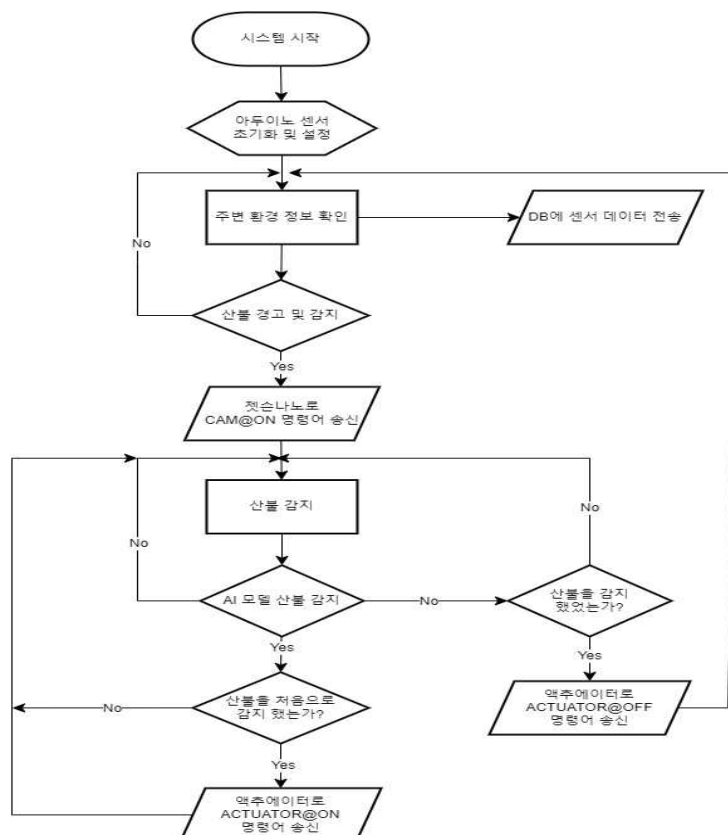


(3) 설비 시스템 외관



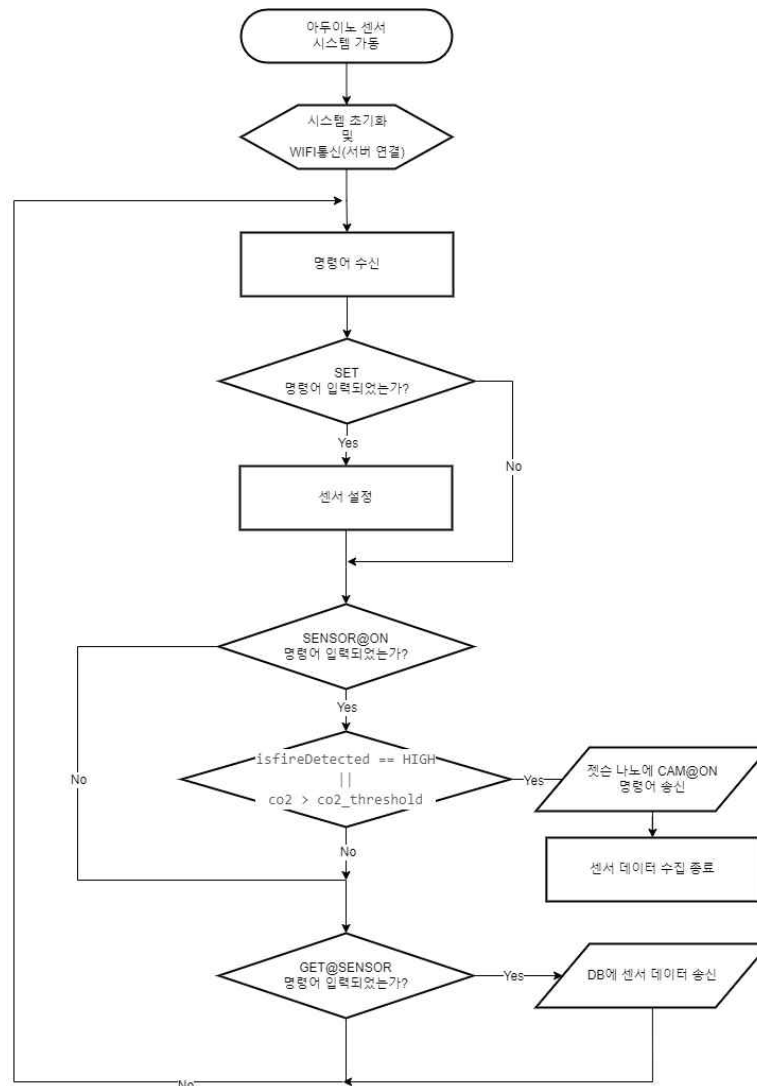
3. 주요 동작 및 특성

(1) 전체 시스템 flowchart



(2) Arduino Sensor

-1. flowchart



-2. 주요 동작

시스템에서 산불을 감지하기 위해서 MQ-135 대기감지 센서와 KY-026 불꽃 감지 센서를 주요 센서로 활용하고, 추가로 주변의 환경 정보를 받기 위해 온도 습도 센서를 사용하였다.

1) 대기감지 센서의 정확성 향상

대기감지 센서의 정확성을 높이기 위해서 SET 명령어를 사용해서 센서 주변 값을 일정 기간 수집 후, 평균을 계산하고 150을 더해 임계값으로 설정하였다. 이를 통해 약간의 환경 변화로 오류가 발생할 수 있는 상황을 줄여서 오류를 최소화하고, 센서 데이터의 신뢰성을 높였다.

```

Output Serial Monitor pi@pi09: ~/iot_socket_idpasswd
Message (Enter to send)
SETTING ON
co2_threshold : 343
co2_threshold : 685
co2_threshold : 1028
co2_threshold : 1372
co2_threshold : 1715
co2_threshold : 2059
co2_threshold : 2401
co2_threshold : 2743
co2_threshold : 3084
co2_threshold : 3426
SET SENSOR COMPLETE
co2_threshold : 342

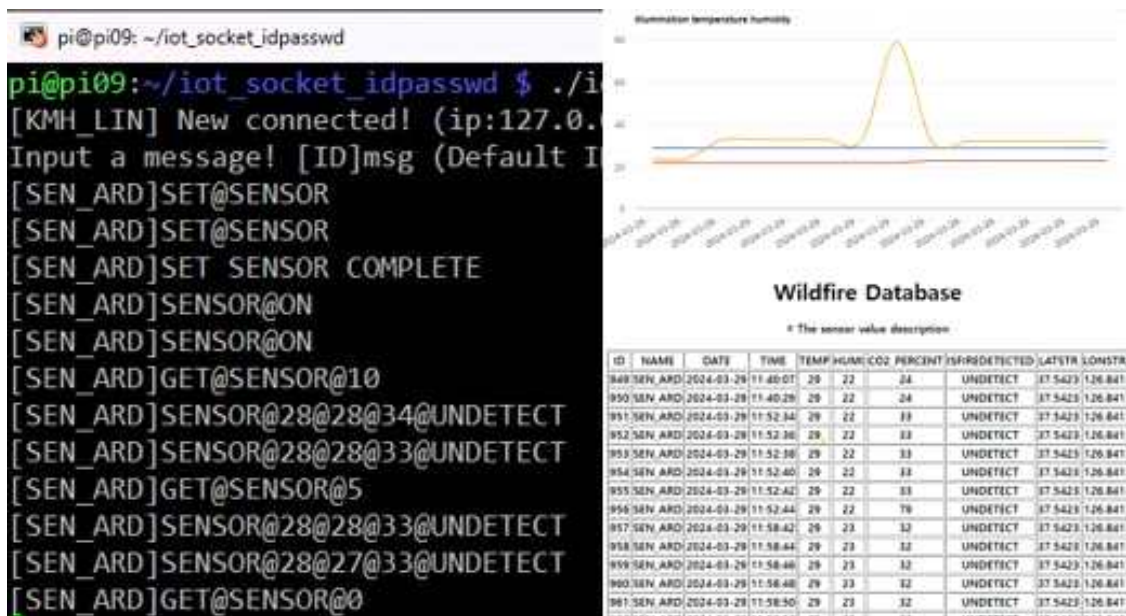
IoT Server Start!!
[KMH_LIN] New connected! (ip:127.0.0.1,fd:4,sockcnt:1)
[SEN_ARD] New connected! (ip:10.10.14.193,fd:5,sockcnt:2)
msg : [KMH_LIN->SEN_ARD] SET@SENSOR
msg : [SEN_ARD->KMH_LIN] SET@SENSOR
msg : [SEN_ARD->ALLMSG] SET SENSOR COMPLETE

pi@pi09: ~/iot_socket_idpasswd
pi@pi09:~/iot_socket_idpasswd $ ./iot_client 127.0.0.1 5000 KMH_LIN
[KMH_LIN] New connected! (ip:127.0.0.1,fd:4,sockcnt:1)
Input a message! [ID]msg (Default ID:ALLMSG)
[SEN_ARD]SET@SENSOR
[SEN_ARD]SET@SENSOR
[SEN_ARD]SET SENSOR COMPLETE

```

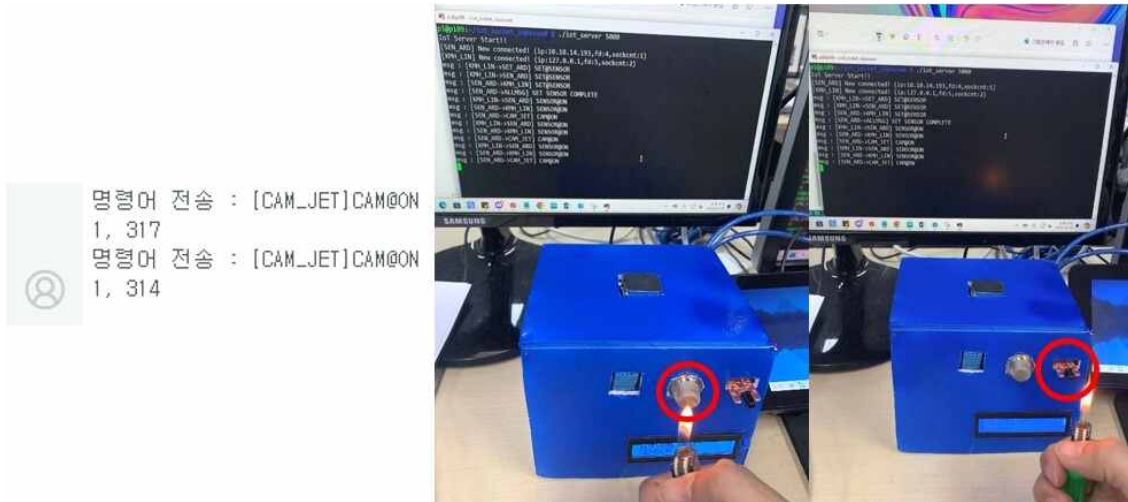
2) 센서값 수집 주기 및 전송

실시간으로 주변 상황을 관찰하기 위해 SENSOR@ON 명령어를 사용해서 1초 주기로 주변 환경 정보를 수집한다. 또한 GET@SENSOR@(second) 명령어를 사용해서 설정한 초마다 DB로 데이터를 전송한다. DB로 보낸 데이터를 활용해서 산불이 발생할 수 있는 환경인지 확인하기 위해서 그래프로 시각화하였다.



3) AI 모델의 감지 시작 조건

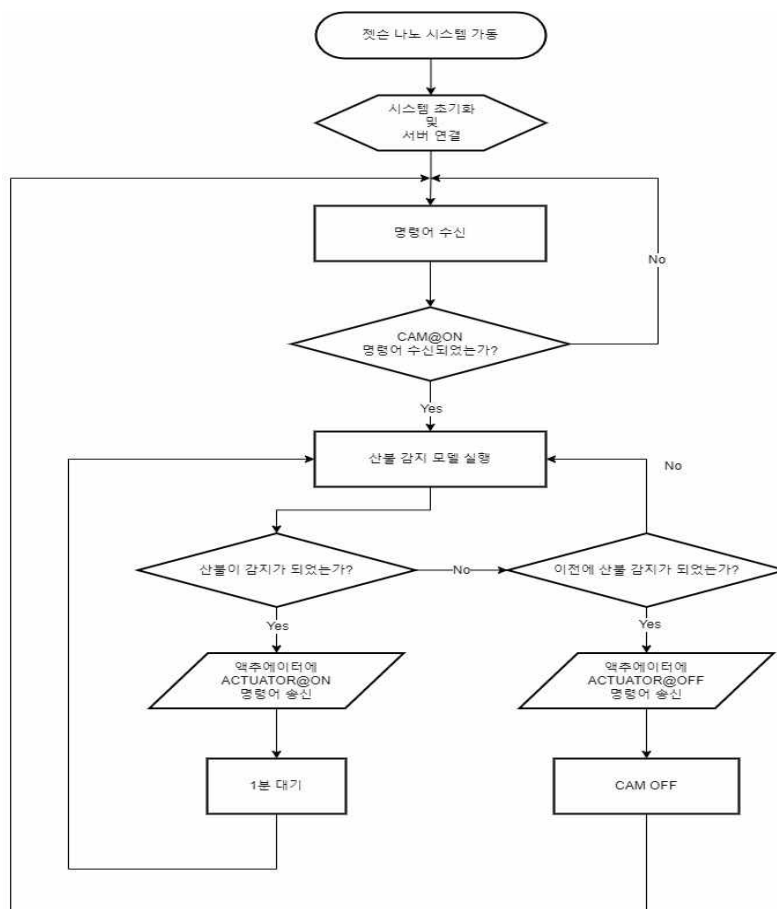
불꽃 감지 데이터와 대기감지 센서의 두 센서값을 종합하여, 두 값 중 하나라도 만족하는 경우 CAM@ON 명령어를 Jetson Nano로 송신해서 AI 모델 감지를 시작하도록 하였다.



명령어 전송 : [CAM_JET]CAM@ON
1, 317
명령어 전송 : [CAM_JET]CAM@ON
1, 314

(3) Jetson Nano

-1. flowchart



-2. 주요 동작

1) 카메라와 AI 모델을 활용한 산불 감지

센서에서 CAM@ON 명령어를 송신하면, 카메라와 사전에 산불과 연기데이터를 학습시킨 YOLO v5 모델을 활용해서 산불 감지를 시작한다. 처음으로 산불을 감지할 때까지 지속해서 감지한다.

2) 액추에이터 동작 조건 및 산불 소멸 판단

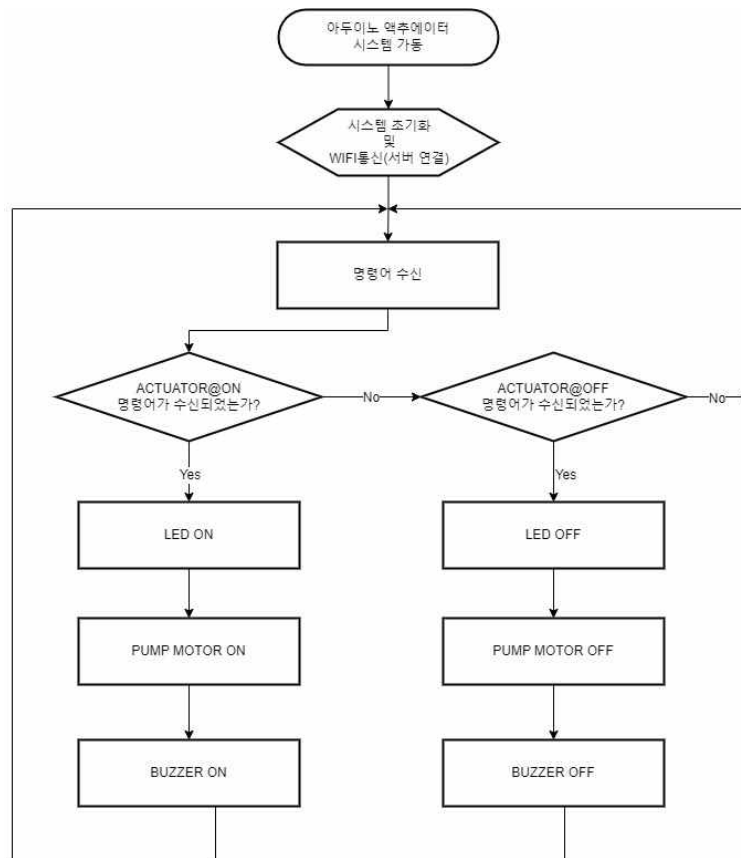
산불이 감지되면, 액추에이터로 ACTUATOR@ON 명령어를 보내서, 펌프를 활성화하고 경보를 울리도록 한다. 그 후 시스템은 1분 동안 대기 하며, 객체 인식을 중지한다. 이후 다시 객체 인식을 시작하여 1분간 산불을 확인하는 작업을 반복한다. 산불이 인식되지 않는다면, 시스템은 산불이 소멸하였다는 것으로 판단해 액추에이터로 ACTUATOR@OFF 명령어를 전송하여 작동을 중지시킨다.



```
[18->CAM_JET] CAM@ON  
[CAM_JET->ACT_ARD] ACTUATOR@ON
```

(4) Arduino Actuator

-1. flowchart



-2. 주요 동작

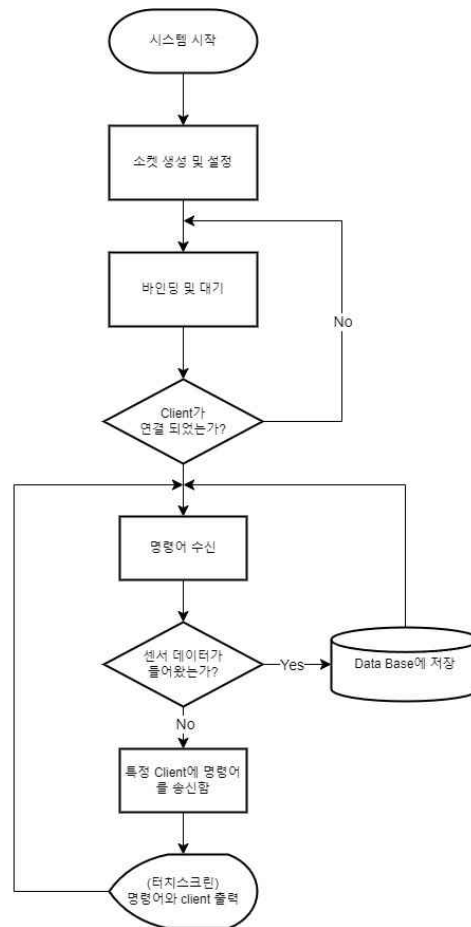
1) LED, PUMP, BUZZER 동작

Jetson Nano에서 산불을 감지했을 때 보낸 ACTUATOR@ON 명령어를 받아서 LED, BUZZER, PUMP를 동작시킨다. LED와 BUZZER를 활용해 주민들에게 산불 경보를 전달하고, PUMP를 사용해서 산불이 마을까지 확산하는 것을 방지해, 주민들의 대피 시간을 확보하는 역할을 한다. 이후 ACTUATOR@OFF 명령어가 수신될 때까지 동작을 반복한다.



(5) Raspberry Pi

-1. flowchart



-2. 주요 동작

1) TCP/IP 통신 서버

라즈베리 파이는 Socket 통신을 사용해 서버 역할을 담당한다. 서버에는 사전에 설정된 ID와 Password를 만족하는 클라이언트만 접속할 수 있도록 하고, 클라이언트들끼리 통신할 수 있도록 한다.

```
pi@pi05: ~/iot_socket/iot_socket_idpasswd
pi@pi05:~/iot_socket/iot_socket_idpasswd $ ./iot_server 5000
IoT Server Start!!
[10] New connected! (ip:127.0.0.1,fd:4,sockcnt:1)
[SEN_ARD] New connected! (ip:10.10.14.193,fd:5,sockcnt:2)
msg : [10->SEN_ARD] SET@SENSOR
msg : [SEN_ARD->10] SET@SENSOR
msg : [SEN_ARD->ALLMSG] SET SENSOR COMPLETE
msg : [10->SEN_ARD] SENSOR@ON
msg : [SEN_ARD->10] SENSOR@ON
msg : [10->SEN_ARD] GET@GPS
msg : [10->SEN_ARD] GET@SENSOR@5
```

2) DB 구성

라즈베리 파이에서 MariaDB를 사용해서 데이터베이스를 구축하고 데이터를 저장한다. 아두이노 센서에 GET 명령어를 송신해 센서가 위치한 곳의 주변 환경에 대한 정보를 수집하고 서버로 송신해, 데이터베이스에 저장한다. 데이터를 사용자가 활용할 수 있도록 Apache를 활용해 웹서버를 만들고, PHP를 사용해 데이터를 시각화하여 데이터의 변화를 알아보기 쉽게 하는 역할을 한다.

```
MariaDB [iotdb]> explain sensor;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(20)	YES		NULL	
date	date	YES		NULL	
time	time	YES		NULL	
temp	float	YES		NULL	
humi	float	YES		NULL	
co2_percent	float	YES		NULL	
isfireDetected	varchar(20)	YES		NULL	
latStr	float	YES		NULL	
lonStr	float	YES		NULL	

```
10 rows in set (0.003 sec)

MariaDB [iotdb]>
```

4. 구현 결과

- 1. 최종 시연



- 2. 시연 영상

첨부파일 참고

Ⅲ. 결론

1. 향후 목표

AI를 활용한 산불 감지 수막 설비 시스템에서는 수막 설비를 대신하여 모터 펌프를 사용하고 일반 웹캠으로 산불을 감지하였으며, 외부 환경 변화가 적은 실내에서 시연을 진행하였다. 이점을 보완하여 상용화하기 위해서는 아래와 같은 향후 목표를 제안한다.

1) 센서 변경 및 추가

탐지 거리가 짧은 불꽃 감지 센서를 긴 거리를 인식할 수 있는 센서로 교체하고, 산불에서 발생하는 일산화탄소와 같은 유독가스를 감지할 수 있는 센서를 추가해 아두이노 센서 부에서 보내는 데이터의 신뢰도를 높일 수 있을 것이다.

2) 빅데이터 활용 기술 적용

센서 부를 통해 얻은 온도 및 습도 및 환경데이터를 활용해서 산불의 발생 가능성이 얼마나 되는지를 분석하고, 이를 바탕으로 경고 및 예방 시스템을 구축하면 산불의 피해를 줄일 수 있을 것이다.

3) 에너지 효율성 개선

전력 소비를 최소화하기 위해서, 하드웨어적으로 저전력 센서 사용, 태양광 및 신재생에너지를 활용해서 시스템에 필요한 전력을 공급하고, 젓슨 나노와 액추에이터와 같이 비활성 상태일 때의 전력 소비를 줄이기 위해서 Sleep Mode를 활용해 시스템을 장기간에 걸쳐 운영할 수 있을 것이다.

4) 설비 비용 개선

기존 주택에 설치되어있는 물탱크와 스프링클러 시스템, 분기 배관 등을 활용해 설치비를 절감시키며 정기적인 유지 보수 또한 쉬워 시설 유지비가 줄어들어 경제적 효용성을 확보할 수 있을 것이다.

2. 기대 효과

- 1) 산불 발생 시 대피 시간 확보와 산불의 번짐을 저지 및 둔화시킬 수 있음
산불로 인해 발생한 대부분의 인명피해는 빠른 확산 속도로 인해 미처 대피

하지 못한 주민들이었다. 본 시스템의 경우 실시간으로 산불을 인식하여 주민들에게 경보 및 수막 설비를 통해 화재 방어선을 구축하여 건조한 날씨에도 주위 환경의 습도를 높여 산불의 빠른 번짐을 저지 및 둔화시킬 수 있다.

2) 적은 인원으로 효율적인 시스템 관리가 가능

기존의 산불관리 시스템의 경우 많은 인력과 장비를 필요로 하였는데 본 시스템의 경우 이를 IoT 센서 장치와 외부 카메라로 대체하고 서버를 통해 중앙통제가 가능하여 실시간 데이터를 기반으로 산불 발생 시 필요한 만큼의 인력과 장비를 적재적소에 투입할 수 있다.

3) 소방산업 활성화

산불의 빈도와 규모 증가 문제는 비단 우리나라만의 문제가 아니라 기후 변화로 인한 전 세계적인 이슈이며 각국에서 첨단 장비와 효율적인 시스템에 대한 수요는 나날이 늘어가고 있어 이러한 추세에 따라, 본 시스템의 적용과 상용화가 소방산업의 대외 수출을 견인할 것으로 기대가 된다.

IV. 참고 자료

1. 참고 문헌

- 산림청, 산불 발생 현황

https://www.forest.go.kr/kfswweb/kfi/kfs/frfr/selectFrfrStats.do?mn=NKFS_02_02_01_05

- KBS, “산불 취약한 산지 주택…‘수막으로 보호’”

<https://news.kbs.co.kr/news/pc/view/view.do?ncd=7909242>

- 인텔리즈, [딥러닝 기술 동향] Yolo V5 - Deep learning

<https://blog.naver.com/PostView.naver?blogId=intelliz&logNo=222824372526>

- YOLO v5 논문 리뷰, <https://cake.tistory.com/34>

- 한국건설기술연구원, (2020.12), “경제성 및 화재 안전성 향상을 위한 스마트 스프링클러 시스템”, p.16

- 한국농촌경제신문, “산불 진화효율 높이자... 'K-산불시스템' 기대”

<http://kenews.co.kr/mobile/article.html?no=79718>

- 이준원, (2020), 메이커를 위한 아두이노의 모든 것, 프리렉

- 윤성우, (2009), 윤성우의 열혈 TCP/IP 소켓 프로그래밍, 오렌지미디어

- 서영진, (2020), 사물인터넷을 위한 리눅스 프로그래밍 with 라즈베리파이, 제이펍

2. 소스 코드

- 첨부파일 참고