*대한기계학회 주최*

## 제13회 전국학생설계경진대회(2023년)

# 설계 최종 보고서

| 참가부 | 고등부 ( O ) | | | | |
|---|---|---|---|---|---|
| 참가분야 | 공모주제 ( O ) / 자유주제 ( ) | | | | |
| 참가팀명 | Snoo.py | | | | |
| 설계제목 | 라디오존데를 활용한 화산 예측 기구 | | | | |
| 지도교수/교사 | (소속) 경북소프트웨어고등학교 (성명) 김종석 (연락처) (이메일) | | | | |
| 대표자<br>(신청인) | 성명 | 소속 | 연락처<br>(휴대폰) | E-mail | 주소 |
| | 김동영 | 경북소프트웨어고등학교<br>소프트웨어개발과 | | | |

## 참가팀원 인적사항

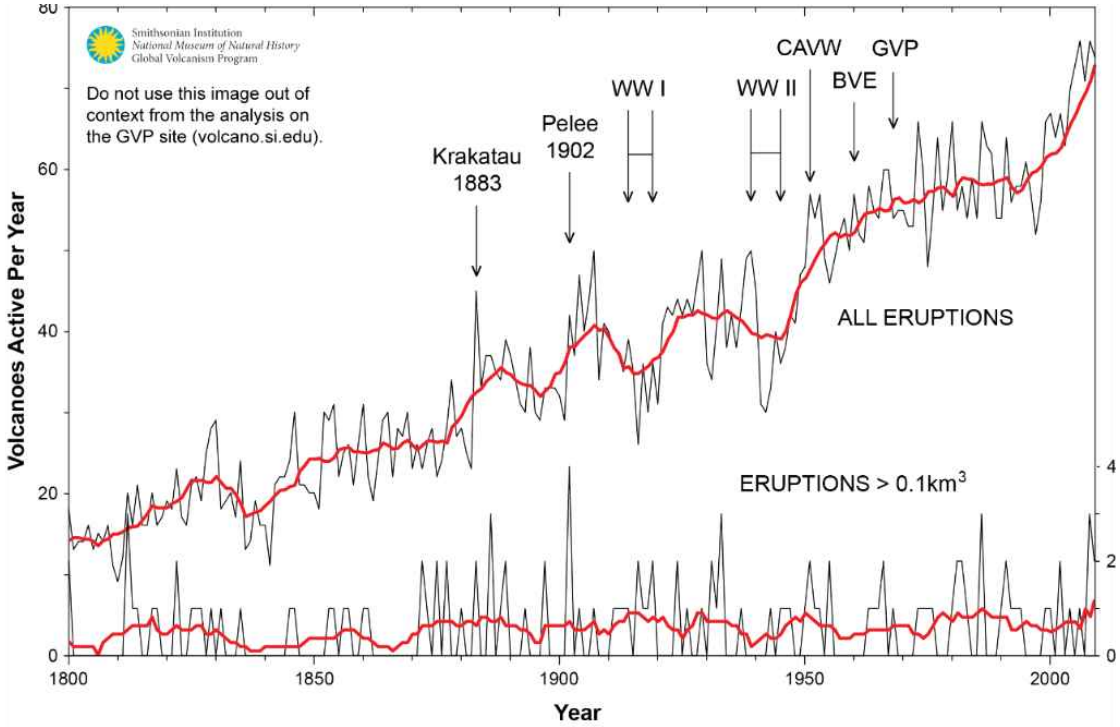
| NO | 성명 | 소속 / 학년 | E-MAIL |
|---|---|---|---|
| 1 | 김동영 | 경북소프트웨어고등학교<br>소프트웨어개발과 / 1학년 | |
| 2 | 변예현 | 경북소프트웨어고등학교<br>소프트웨어개발과 / 1학년 | |
| 3 | 황기성 | 경북소프트웨어고등학교<br>인공지능개발과 / 1학년 | |

# 설계 요약문

| | |
|---|---|
| **참가분야** | **공모주제 ( O ) / 자유주제 ( )** |
| **참가팀명** | Snoo.py |
| **설계제목** | 라디오존데를 활용한 AI 화산 예측 기구 |
| **대표자명** | 김동영 |
| **요약문** | 최근들어 전 세계적으로 화산 분출 사례가 증가하는 추세이다. 현재 화산 예측하는 방법은 여러 가지이지만 여러 한계적인 부분이 존재한다. 높은 기구 설치비용과 기구를 설치할 장소, 시설 관리 비용과 관리자 급여 등 비용적인 부분이 상당 수 발생한다. 그리하여 적은 비용으로 전 세계의 모든 국가에서 사용가능하도록 기상 관측 기기인 라디오존데를 활용하여 화산 분출 전 화산 가스 및 이산화탄소의 분출량이 급격하게 증가한다는 점으로 분출 예측과 분출 후 용암의 이동 반경을 분석하는 두 가지 AI를 개발하여 화산을 예측한다. 화산 예측 과정은 다음과 같다.<br><br><화산 분출 예측 과정><br>1. 헬륨 풍선에 헬륨을 주입하여 화산 상공까지 기구에 도달하게 한다.<br>2. 화산 상공에 도달한 기구는 화산 가스와 온습도를 각각의 센서를 통해 수집한다.<br>3. 수집한 데이터는 지상으로 수신하고 통신 장애를 대비하여 기구 내부의 SD카드에 저장한다.<br>4. 지상국에서는 수신한 데이터를 활용하여 절대습도를 구해 변화량을 비교하는 알고리즘을 구축하여 분출 가능성을 예측하는 AI로 가능성을 예측한다.<br><br><용암 이동 반경 예측 과정><br>1. 화산 상공에 도달하는 과정은 분출 예측 과정과 동일하다.<br>2. 화산 상공에 도달한 기구는 카메라를 통해 분출 중인 용암의 이미지를 촬영하여 지상국으로 송신한다.<br>3. 지상국에서는 화산 분출 후 용암이 흐르는 영상을 통해 학습된 AI를 통해 촬영한 이미지에 용암의 이동 반경을 예측하여 표현해준다.<br><br><활용방안 및 기대효과><br>전 세계의 활화산 수 약 850개정도 분포되어 있어 국내뿐만아니라 전 세계적으로 사용이 가능하며 화산 분출 예측을 통해 주변 인근 지역의 인명, 재산 피해를 최소화, 용암의 이동 반경 분석으로 경로 시각화를 통해 추후 피해 복구에 참조 가능하며,직접적으로 측정하여 정확한 연구데이터 추출로 세밀한 화산 연구가 가능하다. 또한 이러한 데이터들은 화산 연구에 귀중한 자원이다. |
| 설계프로젝트의 입상 이력 | |

## 1. 설계의 필요성 및 목적

전세계적으로 활화산의 수는 약 850개, 휴화산도 여러 존재한다. 이러한 화산은 분출 시 주변 인근에 막대한 재산과 인명 피해를 끼치고, 화산재로 인해 전 세계의 항공편이 마비되기도 한다. 전세계적으로 화산은 최근들어 분출 사례가 증가하는 추세이다.




[19~21세기까지의 전세계 화산 폭발 현황 그래프]

1800년대부터 꾸준히 활동 중이지만 1950년대부터 급격하게 화산 활동 사례가 증가한 것을 확인할 수 있다. 활동이 활발하다는 것은 화산 폭발 가능성이 증가 중이라는 것을 짐작할 수 있다. 전세계의 화산의 활동이 증가하는 추세인 상황에서 화산 분출 사례를 보면 피해가 막대한 것을 확인할 수 있다. 그리하여 화산 분출을 미리 예측하여 피해를 최소화하고자한다.

## 2. 설계 핵심 내용

### (1) 설계 문제의 정의

- 'AI를 통한 화산 예측 기구'는 활화산의 움직임을 관찰하여 화산 폭발 시점을 예측과 폭발 이후의 용암 예측을 통한 효율적 대피로 설계를 목적으로 한다.
- 화산 폭발 시점을 예측과 용암 경로 예측을 통하여 화산 인근 지역 마을의 막대한 인명 피해와 재산 피해를 조금이나마 줄이기 위해 설계한다.
- 기존의 화산 예측 시스템은 설치 및 운용에 대한 비용이 매우 비싸, 개발 도상국의 지역은 사용이 불가하다.
- 기존의 전세계 기상청에서 활용하는 Radiosonde를 활용하여, 우리의 기기를 추가 설치 가능하게 한다.
- 화산 분출 예측을 위해 MQ-8 센서를 통하여 분화구에서 방출되는 수소의 양을 측정한다.
- 화산 분출 예측을 위해 DHT-22 센서를 통해 온습도를 측정한다.

- 화산 폭발 전, 방출되는 화산 가스 내 수소의 농도와 대기 중 수증기가 증가함을 예측 원리로 이용한다.
- 용암 예측을 위해 가시광선 카메라를 통해 일정 시간 기준 움직임 데이터를 측정한다.
- 인공 신경망 기반의 AI를 통해 화산 폭발의 예상 시간과 가능성을 도출한다.
- 이미지 인공 신경망을 기반으로 용암의 이동 반경을 도출한다.
- 원격 솔레노이드 벨브를 장착하여 중력과 부력으로 Radiosonde의 고도를 Control한다.

## (2) 설계의 독창성 및 접근 방법

### 1) 설계 방법 및 배경

- 화산 예측을 위한 Data로 수소 농도, 온도, 습도, 지상 이미지를 기반으로 한다.
- 대기 중 수소 농도를 측정하고자, MQ-8 Sensor를 사용한다.
- 대기 중 온습도를 측정하고자, DHT-22 Sensor를 사용한다.
- 지상의 이미지를 촬영하고자, ov2640기반의 Camera를 사용하여, 촬영한다.
- 위 Data는 실시간으로 RF 통신을 통한 지상국 전송 및 자체 SD에 저장한다.

- 화산 예측은 폭발 시간 예측과, 용암의 이동 반경 예측으로 나누어 개발한다.
- 폭발 예측은 화산 가스 내 수소 농도 변화와, 온습도를 통해 도출한 절대 습도의 변화로 예측한다.
- 용암 이동 반경 예측은 이미지 인공신경망을 기반으로 지상의 용암을 촬영하여 범위를 예측한다.

- Radiosonde의 형식으로 공중에서 머무르며 데이터 수집이 가능하게 한다.
- 전자식 솔레노이드 벨브를 통해 Radiosonde의 He 풍선 내 기체의 양을 조절한다.
- 중력과 부력의 차이를 통해 기구의 고도를 조정하게 한다.

### 2) 설계의 독창성

- 기존보다 더 빠른 시간에 화산의 폭발을 예측하고, 대비할 수 있다.
- 자체적인 인공신경망을 구축하여 더욱 높은 정확도를 기대할 수 있다.
- 화산의 이른 대처로 물적 및 인적 피해를 최소화 할 수 있다.
- 전세계 기상청에서 사용하는 Radiosonde에 우리의 기구를 설치함으로써 도입 비용 최소화가 가능하다.
- 화산에 직접 접근하지 않고 가스, 이미지 같은 데이터를 받을 수 있어 안전하게 모니터링 및 예측이 가능하다.
- 주야에 관계없이 상공을 관측할 수 있어, 온도, 기후 등 외부적인 요소에 많은 영향을 받지 않는다.

### 3) 설계의 제약조건 및 문제 해결 방법

1. MCU Pin 중복: MCU는 제한된 데이터 통신 핀이 존재함. 하지만 MicroSD를 위한 기본 통신 핀과, nRF24 통신을 위한 기본 통신핀이 중복됨.

- nRF24와 MicroSD Card 사이의 중복핀인 Digital 13, 12, 11핀은 기본 세팅을 수정하여 타 핀으로 변경이 가능하도록 설계함. 또한 중복이 가능한 핀의 경우, 병렬 연결을 사용함.

2. 프로토타입 무게: 헬륨을 이용해 공중에 띄어 사용하는 기구로써 그 무게가 가벼워야 함. 만약 무게를 줄이지 못하거나 크기가 커진다면 운용에 장애가 발생함.

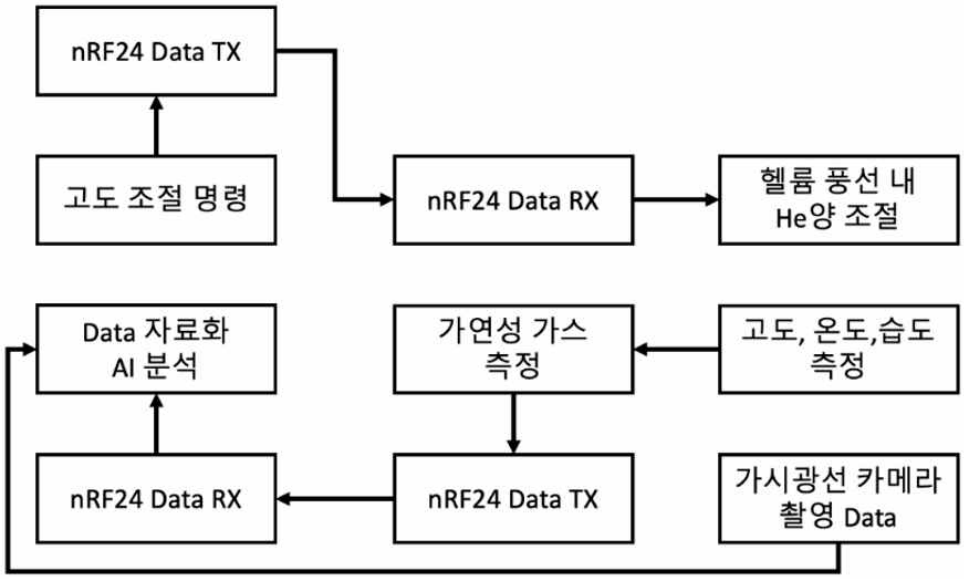
- 프로토타입의 무게를 해결하고자 프로토타입의 하우징은 제거한다. 꼭 필요한 부품만을 기반으로 PCB회로에 설계하여 무게를 조정한다. 각 고정핀과 케이블을 최소화하여 무게와 크기를 매우 효율적으로 줄임.

3. nRF24 전력 수급: 지속적인 RF통신의 특성상 배터리 및 전력 소모가 매우 큼. 이 경우 출력 가능한 송신 파워가 낮아져 유효 통신 거리가 감소하거나, 통신 안정화에 장애가 올 가능성이 매우 큼.

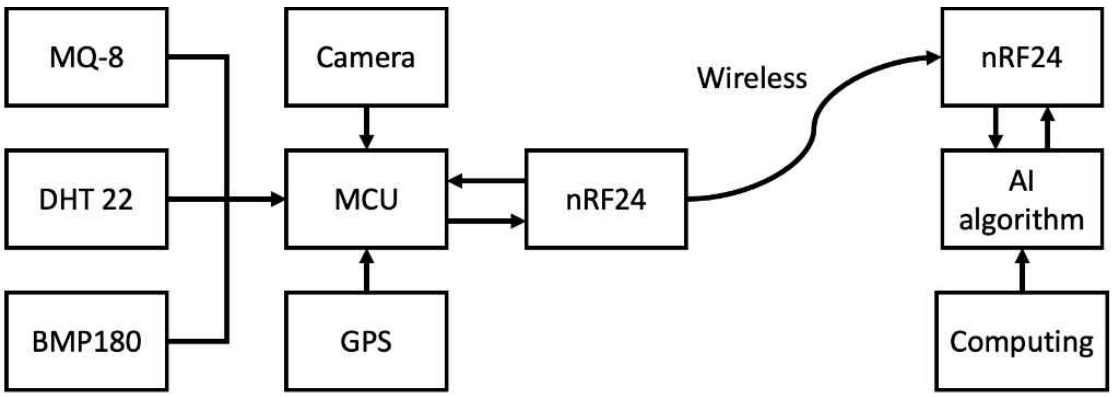
- nRF24는 기본 mcu 출력인 5v가 아닌, 3.3v를 필요로 한다,. 우리의 MCU의 경우 최대 출력가능 전력이 40mA 이다. 하지만 nRF24가 필요한 전력은 충분히 공급하지 못하는 문제가 존재한다. 전원 공급을 안정화 하기 위해 외장형 Power을 사용한다.

## (3) 설계 내용

**<화산 예측 기구 하드웨어 설계>**




[장치 구동 순서도]




[하드웨어 구조도]

지상국이 필요한 기능은 다음과 같다.

- nRF24를 통한 Radiosonde 장치로부터의 데이터 수신
- 실시간으로 수신받은 위성의 고도 및 온습도 데이터 확인
- 수신받은 데이터를 자료화
- 자료화된 데이터를 기반으로 AI 예측 ( 화산 분출 예측, 용암 경로 예측 )

- nRF24를 통한 Radiosonde 장치의 원격 컨트롤

장치에 필요한 기능은 다음과 같다.

- 실시간 고도 및 온습도, 가연성 가스, gps좌표 측정
- 지속적인 영상 또는 이미지 촬영
- 수집한 데이터를 nRF24 통신으로 전송 및 자체 SD 저장
- 지상국의 명령을 수신하여 수행 가능한 기술

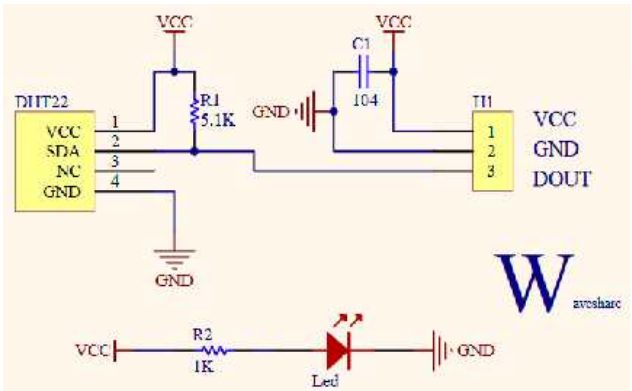
화산 예측 Radiosonde를 이루는 주요 센서와 부품은 다음과 같다.

① nRF24 RF 통신 모듈

nRF24 모듈은 2.4GHz 전파를 사용하여 점대점(2개의 네트워크 노드) 또는 네트워크(많은 네트워크 노드의 네트워크)를 연결할 수 있는 통신 모듈이다.

[nRF24 Module Image]

② DHT22, BMP180




[Left : DHT22 Circuit]

온도 및 습도를 측정하고자 DHT22 센서를 사용한다. 전자식 Sensor로 Digital 통신을 진행한다.

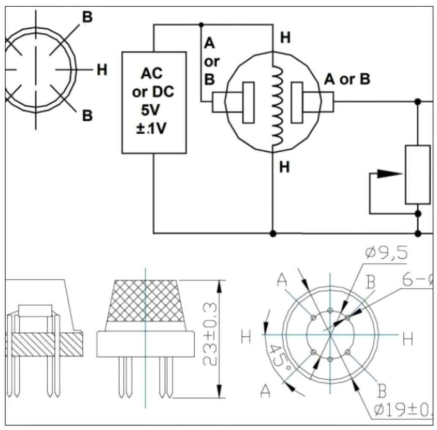
[Right : BMP-180]

공중에 떠있는 장치이므로 고도 측정을 위하여 BMP180 기압 센서를 사용하여
캔위성의 고도를 계산한다.

$$altitude = 44330 * \left( 1 - \left( \frac{p}{p_0} \right)^{\frac{1}{5.255}} \right)$$ [고도 산출 공식]

해수면 기준 기압 $p_0$과 센서에서 측정한 기압 p를 통하여 고도를 계산하는 방식이다.

③ 가연성 가스 측정 센서



[MQ-8 Sensor Circuit]

MQ-8 수소 측정 센서는 금속 산화물 반도체(MOS) 유형 센서이다.
이 센서는 수소 가스에 매우 민감하며 공기 중 수소 가스의 존재를 감지하는 데 사용됨.
아날로그식 센서로 Anolog 통신으로 데이터를 읽는다.

④ 솔레노이드 벨브

[솔레노이드 벨브 이미지]

솔레노이드 벨브는 전자식으로써, 전기 신호를 통해 Radiosonde의 He풍선 내 기체의 양을
조절 할 수 있다. 이 방식으로 기구의 고도를 Control한다.

지상국의 통신 원리는 다음과 같다.

해당 Radiosonde와 지상국과의 통신 및 데이터 전송은 전파법에서 특정소출력 (무선데이터통신시스템 용)으로 허가난 2.400 ~ 2.4835GHz의 대역에서의 데이터 통신을 진행한다.

이 방식으로 RF방식의 통신을 사용하여 센서에서 측정한 데이터와 이미지를 지상국으로 전송한다.

**<화산 폭발 예측 AI>**

- 화산 분출 예측 AI는 pandas 라이브러리를 사용해 설계함.
- 화산 폭발 예측 시스템은 가연성 가스 농도 증가 비율과 절대습도 증가 비율을 시각화해 예측을 수행함.
- 가연성 가스 농도는 화산 활동의 변화를 알 수 있는 지표이고, 절대습도는 화산 폭발 전 급격하게 변화하는 지표인 수증기의 실질적인 양을 구하기 위함임

1. 온도와 상대습도를 이용해 절대습도를 계산하는 함수를 정의하며 절대습도 계산 공식은 다음과 같다.

**상대습도**

$$H_r = \frac{\text{수증기압}}{\text{포화수증기압}} \times 100\% = \frac{P_W}{P_S} \times 100\%$$

```
# 온도와 상대습도를 이용하여 절대습도 계산 함수
def calculate_absolute_humidity(temperature, relative_humidity):
    e = 6.112 * np.exp((17.67 * temperature) / (temperature + 243.5))  # 포화 수증기 압력 계산
    absolute_humidity = (0.622 * e) / (pressure - (0.378 * e))  # 절대습도 계산
    return absolute_humidity
```

[온도와 상대습도를 이용한 절대습도 계산 함수 코드]

위 코드를 정의해 포화 수증기 압력을 계산한 후 절대습도 계산 코드에 변수를 input해 절대습도를 구함.

2. 가연성 가스 증가 비율은 화산 활동 변화를 알 수 있고, 이를 통해 화산 폭발 위험성을 예측할 수 있으며 다음과 같은 함수를 정의해 계산함.

```
# 가연성 가스 증가 비율 계산
pre_eruption_df["Gas_Increase_Rate"] = (imminent_eruption_df["Gas_Concentration"]
 - pre_eruption_df["Gas_Concentration"]) / pre_eruption_df["Gas_Concentration"]
```

[가연성 가스 증가 비율 계산 코드]

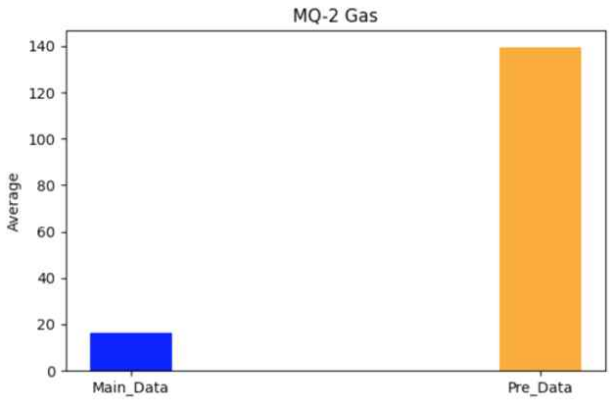
3. 위에서 함수를 정의해 가연성 가스 증가 비율, 절대습도를 계산한 후 계산 값을 시각화해, 변화량을 모니터링해 보다 빠르게 화산 폭발을 예측할 수 있게 설계함.

```python
# 시각화
plt.figure(figsize=(10, 6))

# 가연성 가스 농도 증가 시각화
plt.subplot(2, 1, 1)
plt.plot(pre_eruption_df["Gas_Increase_Rate"], label="Gas Increase Rate", color='blue')
plt.title("Gas Increase Rate Over Time")
plt.xlabel("Time")
plt.ylabel("Gas Increase Rate")
plt.legend()

# 절대습도 증가 시각화
plt.subplot(2, 1, 2)
plt.plot(pre_eruption_df["Absolute_Humidity"], label="Absolute Humidity", color='green')
plt.title("Absolute Humidity Over Time")
plt.xlabel("Time")
plt.ylabel("Absolute Humidity")
plt.legend()

plt.tight_layout()
plt.show()
```

[예측 결과 시각화에 관한 코드]

```python
if pre_data_avg['Ab Humi(g/m³)'] > main_data_avg
    (pre_data_avg['CO2'] / main_data_avg['CO2']
    explosion_probability = (ratio_difference).
    print(f"화산 폭발 가능성이 {explosion_probabilit
    time_to_100_percent = 100 - explosion_probal
    remaining_time = time_to_100_percent / 3

    print(f"폭발까지 남은 예상 시간: {remaining_time:
else:
    print("화산 폭발 가능성이 없습니다.")
```

```
화산 폭발 가능성이 85.84%로 있습니다!
폭발까지 남은 예상 시간: 4.72 시간
```




[Left : 폭발 예측 시간 도출 결과] [Right : Matplolib를 통한 데이터 시각화 결과]

**<용암 이동 반경 예측 AI>**

- 화산 폭발 후 가장 안전한 대피 경로를 찾기 위한 용암의 이동 반경을 예측하고자 함.
- CNN과 RNN 계열의 LSTM 모델을 사용해서 예측 인공신경망 AI 모델을 구축함.
- CNN은 이미지 처리에 유용한 기술인 공간적 계층 구조, 합성곱 연산 등으로 구성 되어있어 이미지의 특성을 추출하기 좋음.
- LSTM은 장기 의존성 학습을 통해 시계열 데이터 처리에 좋음.
- 원래는 실시간으로 객체를 잡아낼 수 있는 YOLO가 더 적합해 YOLO를 사용해 이동 반경을 예측하고자 했으나, 데이터 수가 적다는 문제가 있어 객체를 잡지 못할 것이라고 생각되어 사용하지 않음.

1. image 데이터의 크기가 맞지 않을 경우 예상치 못한 오류와 성능 저하 문제를 초래할 수 있기 때문에 image 데이터를 전처리 하는 과정을 수행함.

```
import cv2
import numpy as np

resized_images_data = []

for img in images_data:
    resized_img = cv2.resize(img, (360, 240))
    resized_images_data.append(resized_img)

resized_images_data = np.array(resized_images_data)

resized_images_data / 255.0
```

[이미지 사이즈 전처리 코드]

2. 훈련을 위한 AI 모델은 함수형 API를 사용해, CNN 모델과 LSTM 모델을 함께 사용함.
- 이미지 데이터 처리, 시계열 데이터 처리에 효과적인 모델을 만들었습니다.
- AI 모델은 데이터에 복잡한 패턴을 학습할 수 있게 해주기 위해 비선형 특성인 Selu activation을 추가함.
- 가중치 초기화 기법인 르쿤을 사용함.
- 모델을 최적화하여 빠르고 정확하게 수렴할 수 있도록 nadam optimizer를 사용함.
- 회귀 문제로 loss는 mse로 설정해 모델이 작은 loss에도 크게 반응하여 loss를 최대한 낮추도록 설정함.

```
from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D, Flatten, LSTM,
from tensorflow.keras.layers import concatenate, Dense, Dropout
from tensorflow.keras.models import Model

def create_model(image_shape = [240, 360, 3], time_steps = 10, num_features = 16):
    image_input = Input(shape=[240, 360, 3])  # 이미지 데이터 입력
    conv1 = Conv2D(32, (3, 3), activation='selu')(image_input)
    pool1 = MaxPooling2D((2, 2))(conv1)
    conv2 = Conv2D(64, (3, 3), activation='selu')(pool1)
    pool2 = MaxPooling2D((2, 2))(conv2)
    flat = Flatten()(pool2)

    time_input = Input(shape=(time_steps, num_features))  # 시계열 데이터 입력
    lstm1 = LSTM(32, return_sequences=True)(time_input)
    lstm2 = LSTM(16)(lstm1)

    merged = concatenate([flat, lstm2])  # CNN과 RNN 결합
    dense1 = Dense(128, activation='selu')(merged)
    dropout1 = Dropout(0.5)(dense1)
    dense2 = Dense(64, activation='selu')(dropout1)
    dropout2 = Dropout(0.5)(dense2)
    output = Dense(16, activation='linear')(dropout2)  # 회귀 문제이므로 linear 활성화 함수
```

[용암 이동경로 예측 모델 코드 중 일부]

3. 더욱 효과적인 학습을 위한 Epochs도출 방안을 적용함.
- epochs 증가에 따라 성능 향상이 없을때, 훈련을 멈추고 최적의 상태로 돌려주는 callbacks을 설정함.
- 만약 validation loss가 일정 epochs가 지나도 좋아지지 않아도, epochs를 진행하면 Overfitting 문제가 생길 수 있기 때문에 patience를 20으로 설정함.
- 이는 20번의 epochs 동안 validation loss가 줄지 않는다면 epochs를 중단하고 최적의 상태로 돌아감.

```
early_stooping_cb = keras.callbacks.EarlyStopping(patience=20,
                                                  restore_best_weights=True)

checkpoint_cb = keras.callbacks.ModelCheckpoint("save_my_keras_model.h5",
                                                save_best_only=True)
```

[Epochs callback 관련 코드]

4. 모델은 서브클래싱 API를 사용함.

- input과 validation data에는 시계열 데이터와 용암 이동 반경 이미지가 들어있는 list로 전달함.
- callbacks을 사용했기 때문에 epochs 하이퍼파라미터를 상관하지 않고 크게 지정함.

```
history = model_instance.fit([X_train, X_train_time_reshaped], y_train, batch_size=32,
                             epochs=1000000000, validation_data=([X_valid, X_valid_time_reshaped], y_valid),
                             callbacks=[tensorboard_cb, early_stooping_cb])
```

[서브클래싱 API 관련 코드]

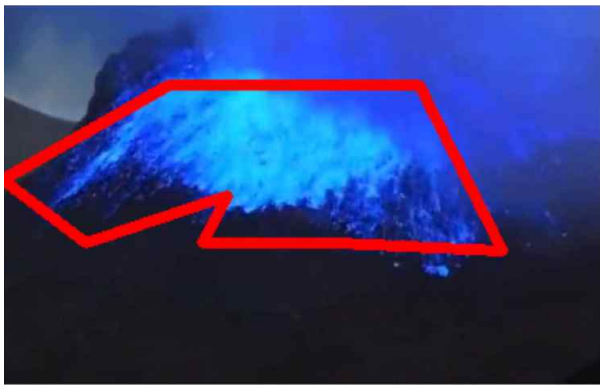
5. 훈련 후 예측을 잘 수행하는지 확인하기 위해 test data를 사용해 모델이 정상 작동 하는지 테스트함.

- 정상 작동 하는 것을 확인함.
- predict 메서드에 이미지 test와 시계열 test 데이터를 전달해 예측을 수행함.

```
1/1 [==============================] - 0s 127ms/step
array([[215.06795 , 135.33272 , 144.27023 , 130.21808 , 126.14896 ,
        134.7996  ,  95.0365  , 132.31268 ,  70.87385 , 129.1241  ,
         36.809635, 119.55839 ,  17.66604 , 122.40466 ,  10.484476,
        110.119545]], dtype=float32)
```

[도출 데이터의 좌표화 결과]

6. 여러 test를 해본 결과 AI 모델이 용암의 이동 반경 예측을 잘 수행한다는 것을 확인함.



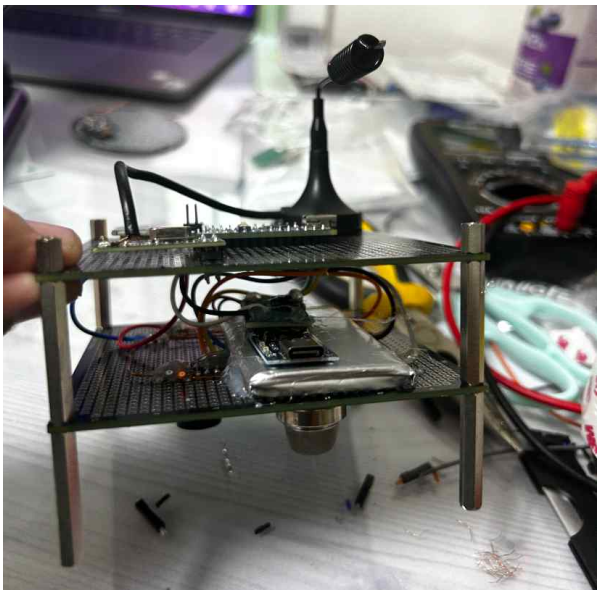
[Left : Test를 위한 국제 화산 연구소에서 제공된 화산의 용암 이미지]
[Right : 국제 화산 연구소의 이미지를 기반으로 범위를 예측한 결과]

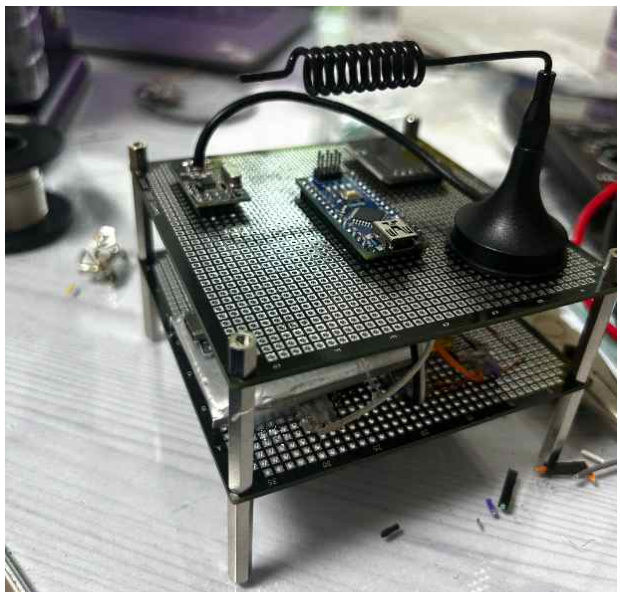
## 3. 설계 수행 일정

| 설계 진행 내용 | 4월 | 5월 | 6월 | 7월 | 8월 | 9월 |
|---|---|---|---|---|---|---|
| 아이디어 구체화 및 설계 | | | | | | |
| 회로 및 통신 설계 | | | | | | |
| AI모델, 소프트웨어 개발<br>( 중간보고서 ) | | | | | | |
| 하드웨어 프로토타입 실제 개발 | | | | | | |
| 하드웨어와 AI 연결 | | | | | | |
| 테스트 환경 구축 및 테스트 | | | | | | |
| 최종 보고서 작성 | | | | | | |

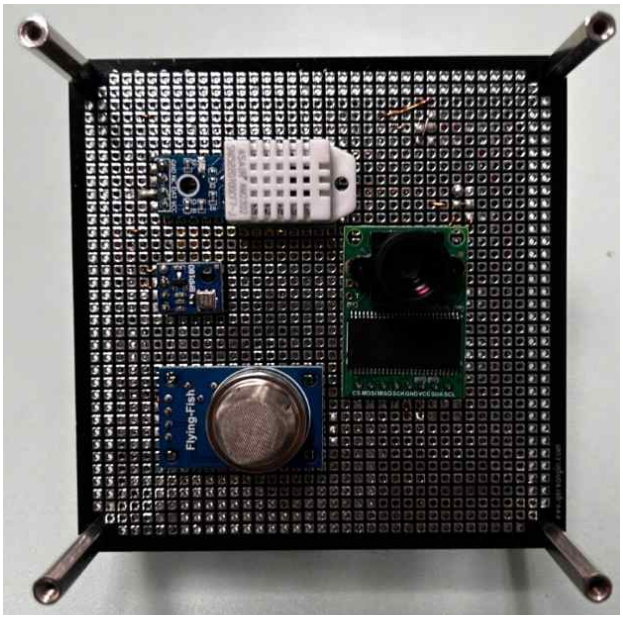
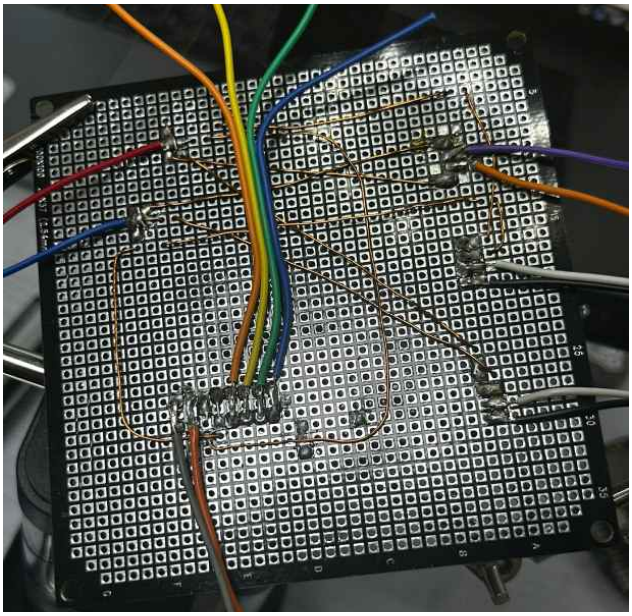
## 4. 설계 결과물

### (1) 최종 결과물 형상 및 작동원리





[Left : Radiosonde Side Image] [Right : Radiosonde Upper Image]

[Left: Radiosonde bottom image] [Right: Radiosonde bottom circuit]

- DHT22를 통한 온습도 측정, MQ-8을 통한 수소 및 가연성 가스 측정을 통해 화산의 수치화 데이터 수집
- Camera를 통한 지상 이미지 촬영 및 수집
- nRF24 통신을 통해 지상으로 수집된 Data를 전송함.
- 지상국에서는 화산 폭발 예측 AI모델과 용암 분석 AI모델을 통해 두가지를 분석함.
- 분석한 자료는 matplolib를 통해 시각화 하여 화산 대비 및 연구에 사용함.
- Arduino Nano를 MCU로 사용하여 개발함.
- C++ 기반의 알고리즘과 코드를 작성하여, PnP (Plug and Play) 방식으로 지상국과의 연결만으로 사용가능.
- nRF24와 솔레노이드 벨브의 높은 전력을 감당하고자 외부 Power를 장착함.
- 외부 Power는 Li-po Battery를 3.4v 1Cell에 C-Type의 BMS의 연결하고, 5v로 승압하여 사용함.
- 기존의 nRF24의 통신거리를 극대화하고자 자체 안테나를 사용하여, 예상 통신거리 40km로 계산됨.

## (2) 최종설계 결과물의 장단점 및 의의

- 화산 폭발을 미리 예측하는 장치와 기술이라는 측면에서 인명 및 재산 피해의 최소화가 가능함.
- 기존의 예측 방식은 도입 및 유지 비용이 수십억에 달하지만, 저예산으로 운영이 가능함.
- 전세계 기상청에서 이미 도입중인 Radiosonde 시스템이 결합하여, 도입 비용 최소화가 가능함.
- 직접적으로 측정하는 것이 아닌, 기구를 통한 반직접, 반간접적인 측정으로 운용자 안전성 보장이 가능함.
- 기존의 radiosonde와 다르게 솔레노이드 벨브를 통한 He풍선 고도 조절이 가능함.
- 하지만 AI의 특성상, 예측하고자 하는 화산 환경에 대한 측정 데이터가 다량 필요함.
- 정확도 향상을 위해 여러개의 기구를 통한 동시적 측정 또는 지속적인 데이터 축척이 필요함.
- 기존의 화산 예측 시스템과 다르게, 정확도와 운용도는 유지하며 가격을 획기적으로 낮추는 기술로써 개발 도상국과 같은 소외된 국가에서도 보급이 가능한 예측 기술이라는 의의가 있음.
- 기존의 Radiosonde와 같은 항공 측정 기술과 결합을 통해 더욱 발전이 가능하다는 의의가 있음.
- 화산 예측에 최적화된 자체 인공지능 신경망을 구현하여, 개발한 것에 대한 의의가 있음.

## 5. 활용방안 및 기대효과

- 이 기구를 통해 화산 예측을 하여 불시에 분출할 화산에 대한 걱정을 최소화 할 수 있을거라 기대함.
- MQ-2, 온습도 센서를 통해 수집된 센서 데이터 값을 기반으로 개발한 분출 예측 AI를 통하여 기존의 방법보다 효과적으로 화산 분출 예측이 가능함.
- 화산 분출 예측을 통하여 화산 인근에 거주중인 주민들에게 사전에 알려 인적 및 물적 피해를 최소화 가능.
- 화산 상공에서 지속적으로 촬영한 이미지를 통해 용암의 이동 반경 이미지를 지상국으로 송신하여 용암 이동 반경 예측 AI를 통해 용암의 이동 반경을 효과적으로 예측할 수 있음.
- 지속적으로 촬영한 이미지를 통해 용암의 이동 반경을 예측하여 피해 반경을 사전에 예측하고 분출 후 피해 복구에 참조하여 효율적으로 복구를 할 수 있음.
- 이러한 방식으로 얻은 연구 데이터는 추후 화산 연구에 귀중한 자원이 될 것이라 예상됨.
- 적은 비용으로 기구를 개발하여 개발도상국과 같은 나라도 상용이 가능하게 함.
- 기존의 Radiosonde System에 효과적으로 연동이 가능함.

## <참고문헌>

① 홍의석(Eui-seok Hong),박승신(Seung-sin Park),and 강왕구(Wanggu Kang). "관광용 계류식 헬륨기구 개발." 대한기계학회 춘추학술대회 2005.5 (2005): 1670-1675.

② 장대원. "화산재해 예측 시스템." 한국재난정보학회 학술대회 2012.1 (2012): 27-38.

③ 김혜림. "백두산 화산 분화 시 화산폭발지수에 따른 대기질 및 건강영향 예측." 국내석사학위논문 건국대학교, 2019. 서울

④ 김상곤,김홍일,and 이상철. "고고도 비행시험을 위한 기상 정보 획득 및 활용." 한국항공우주학회 학술발표회 논문집 2015.11 (2015): 959-962.

⑤ 김형진(Hyeong-Jin Kim),김동성(Dong-Seong Kim),and 이재민(Jae-Min Lee). "AI 기반의 무인항공기 임무 경로계획 기법 설계 및 성능평가." 한국통신학회 학술대회논문집 2020.8 (2020): 522-523.