

2023

프로젝트 계획 & 결과 보고서



프로젝트명 : 차량 데이터를 이용한 번호판 예측

수행 기간 : 2023.8.21 ~ 2023.9.22

팀원 : 김재민(백엔드)

김은진(프론트엔드)

김민범(데이터분석)

최호진(데이터분석)

CONTENTS

01 프로젝트 계획

02 데이터 탐색

03 모델 선정

04 YOLOv5

05 ESRGAN

06 OCR

07 CNN

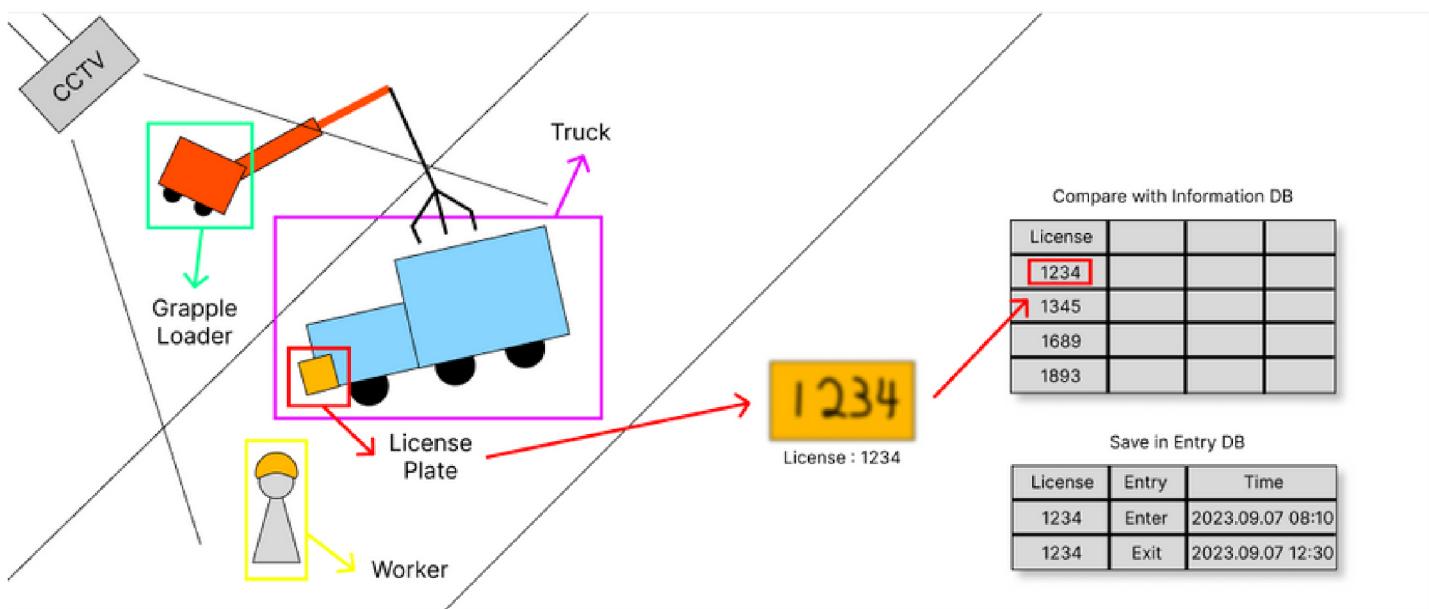
08 프로젝트 결과

09 프로젝트 소감

1. 프로젝트 계획

동연SNT에서 고철장에 있는 트럭의 번호판을 식별하여 번호판의 번호를 예측하여 차량 정보를 조회하고 차량의 입출입을 관리하는 시스템을 제안하였다. 이에 우리는 주어진 데이터를 활용하여 차량 번호판을 예측하는 프로젝트를 진행하고자 한다.

1. 프로젝트명 : 차량 번호판 식별 및 예측 시스템 개발
2. 팀 구성 : 김재민, 김은진, 김민범, 최호진
3. 제안자 : 동연SNT
4. 기술 분야 : 컴퓨터 비전, 머신러닝, 딥러닝, 데이터 분석 및 처리
5. 목적 : 차량 정보 확인 및 작업 증빙 활용
6. 일정 : 2023.8.21(월) ~ 2023.09.22(금)
7. 세부 과정
 - CCTV를 이용하여 촬영 (목적 : 작업 현장 모니터링)
 - 촬영한 현장 사진에서 번호판 추출
 - 추출된 번호판 번호 예측
 - 예측 결과와 사전에 기록된 정보 DB와 비교
(기록이 있는 경우 => 입출입 DB에 기록)
(기록이 없는 경우 => 정보 DB에 신규 추가)



2. 데이터 탐색

프로젝트를 진행하기 위해 필요 데이터를 수집하고 탐색적 데이터 분석(EDA)를 통해 수집한 데이터를 분석하였다. 이를 통해 데이터의 정보를 파악하고 여러 문제점들을 발견할 수 있었다.

No. 1 데이터 수집

1. 수집 데이터

- 현장 사진(약 30GB), 번호판 이미지(약 300GB)

2. 수집 방법

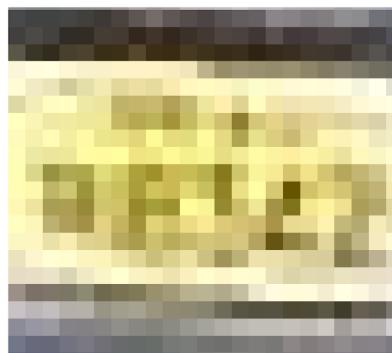
- 동연SNT에 데이터 요청
- 현장에서 촬영한 사진 및 번호판 이미지



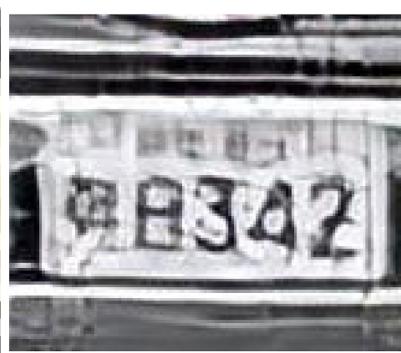
현장 사진



번호판 이미지



번호판 추출 사진



화질 개선 결과

No. 2 문제점

1. 화질 문제

- 현장 사진에서 추출한 번호판 화질이 너무 낮음
- 번호판 객체 식별 및 번호 예측의 어려움 발생

2. 라벨링 문제

- 수집된 이미지 데이터에 라벨링 정보가 없음

3. 파일 형식 문제

- 확장자가 지정되지 않은 이미지 파일 제공(file)
- 이로 인해 파일의 열람, 편집 및 처리의 어려움 발생

No. 3 해결 방안

1. 화질 해결 방안

- 화질을 극복할 수 있는 모델 제작

2. 라벨링 해결 방안

- 라벨링 수량 지정
- 현장 사진 1000장, 화질 개선 번호판 700장

3. 파일 형식 해결 방안

- 파일명 뒤에 확장자를 추가하는 코드 제작

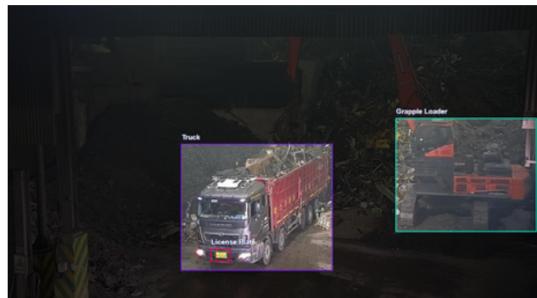
```
jpg_converter.py
1 import os
2
3 def rename_folders_and_files(folder_path):
4     # 현재 폴더 내의 파일 및 폴더 목록 가져오기
5     contents = os.listdir(folder_path)
6
7     for item in contents:
8         item_path = os.path.join(folder_path, item)
9
10    if os.path.isfile(item_path): # 형식이 파일인 경우
11        if not item.endswith(".jpg"): # 이름 뒤에 jpg로 끝나지 않는 경우
12            new_item_name = item + ".jpg" # 이름 뒤에 jpg를 붙인다
13            new_item_path = os.path.join(folder_path, new_item_name)
14            os.rename(item_path, new_item_path)
15
16    elif os.path.isdir(item_path): # 폴더인 경우
17        rename_folders_and_files(item_path) # 재귀적으로 하위 폴더 내의 파일 및 폴더도 처리
18
19 # 상위 폴더 경로 설정
20 parent_folder_path = "./snapshot" # 상위 폴더 경로를 적절히 수정해주세요
21
22 rename_folders_and_files(parent_folder_path)
23
24 print("작업 완료!")
```

jpg 변환 코드

3. 모델 선정

프로젝트를 진행하기 위해 총 4가지 모델을 제안하였다.

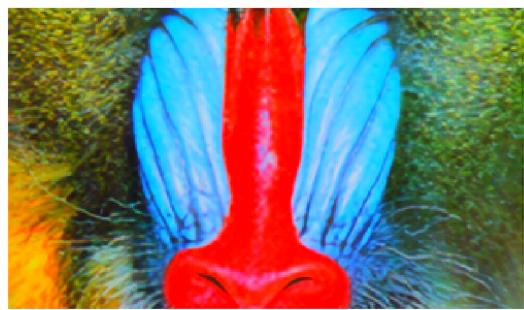
Model 1 YOLOv5



객체 인식 모델 (번호판 추출)



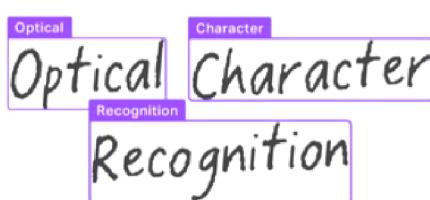
Model 2 ESRGAN



화질 개선 모델 (선명도 상승)



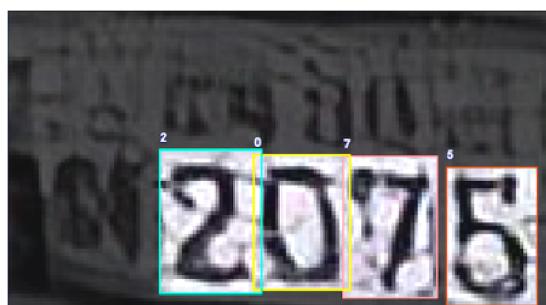
Model 3 OCR



숫자 인식 모델 (번호 식별)



Model 4 CNN



숫자 분류 모델 (번호 예측)



4. YOLOv5

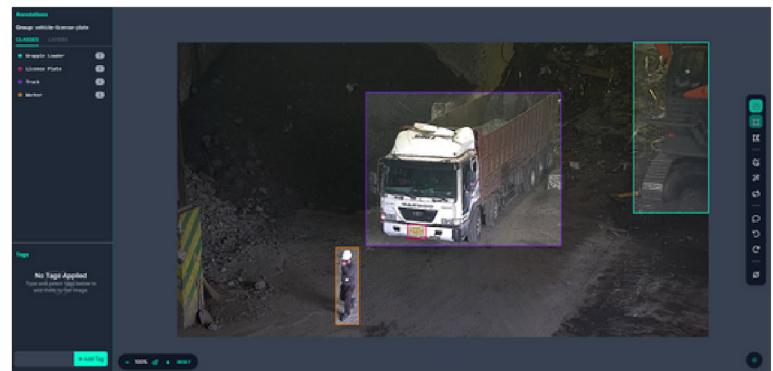
YOLOv5를 이용하여 객체 식별 모델을 제작하였다.

1. 데이터 라벨링(roboflow)

- 클래스 : 4개 (Grapple Loader, License Plate, Truck, Worker)
- 총 데이터 : 현장 사진 2400장 (train : 2100장, val : 200장, test : 100장)

2. 데이터 전처리

- 이미지 크기 : 640x640 조정
- 자르기 : 가로 25-75%, 세로 25-75%
- 뒤집기 : 수평 방향
- 회전 : 좌우 15° 방향
- 전단 변환 : 상하좌우 10° 방향
- 밝기 : -20% ~ 20%



3. 모델 학습

- epoch = 200, batch size = 32로 지정
- weights = yolov5s로 지정
- 학습 결과가 가장 좋은 가중치인 best.pt를 저장
- 식별한 번호판 이미지 저장

```
Training started at 2023-09-11 10:45:00. Best results are saved in best.pt. Best model saved in model.pt.
To update EarlyStoppingPatience(100) pass a new patience value, i.e. "python train.py --patience 300" or use "--patience 0" to disable EarlyStopping
63 epochs completed in 1.137 hours.
initializer stripped from runs/train/junkyard/model/weights/last.pt, 14.5KB
initializer stripped from runs/train/junkyard/model/weights/best.pt, 14.5KB
Validating runs/train/junkyard/model/weights/best.pt...
using layers...
YOLOv5 summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
    Class   Images Instances   P   R   mAP50   mAP50-95: 100% 4/4 [00:04<00:00, 1.04s/it]
      all     200     270   0.859   0.765   0.787   0.613
      Grapple Loader  200      8   0.706   0.604   0.621   0.503
      License Plate  200     74   0.948   0.976   0.981   0.628
      Truck        200    180   0.959   0.978   0.985   0.867
      Worker        200      8   0.921   0.5   0.559   0.454
Results copied to runs/train/junkyard/model/weights/best.pt
```

4. 문제점

- 특정 class (Grapple Loader) 분류 오류
 - 트럭으로 오분류하거나 미분류하는 현상 발생
 - 가벼운 가중치 사용 원인 추정 (YOLOv5s)
- 분류한 번호판의 크기가 너무 작음
 - 화질이 깨지거나 흐릿하게 보이는 현상 발생
 - 심한 경우 유판으로 확인이 불가능한 정도
- 해결 방안
 - 이미지의 화질을 올리는 작업이 필요
 - ESRGAN 모델을 프로젝트에 맞게 제작



5. ESRGAN

SUPER RESOLUTION

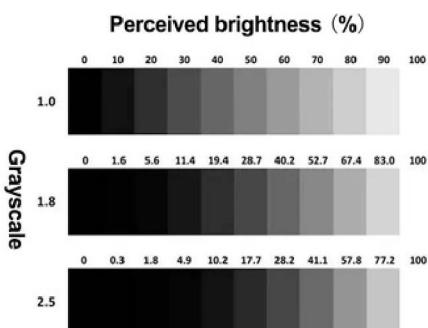
문제점 : 이미지 번호판의 품질 문제로 인해 모델 학습이 어려운 상황에 직면했다.

해결 방안 : 1. 선명도 문제 해결을 위해 ESRGAN 모델을 이용한다.
2. Grayscale 기능을 추가하여 OCR, CNN 모델 학습의 효율을 올린다.



No. 1 이미지 선명도 증가

저해상도의 이미지를 고해상도로 변환하는데 중점을 둔 모델이기 때문에 이미지 선명도 상승



No. 2 흑백 변화

기존의 ESRGAN 모델에서 grayscale 기능을 추가하여 모델의 학습 효율을 올릴 수 있도록 모델 설정

No. 3 적용 결과

기존 번호판 이미지에 비해
월등히 개선된 모습을 확인



6. OCR

주차장에서 흔히 사용되는 OCR(광학 문자 인식) 기능을 이용하여 번호판에 기재된 번호를 식별한다.
처음에는 tesseract를 이용하여 프로젝트를 진행하였으나 좋은 결과를 얻지 못하였다.
이후, 우리는 다른 대안인 easy-ocr을 시도해보았고, OCR 중에서는 가장 좋은 결과를 얻을 수 있었다.

1. OCR 결과

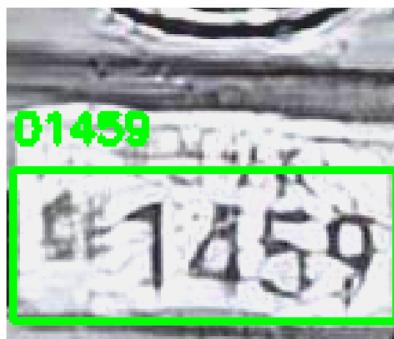


Plate 1 : 1459

Class : 01459

Confidence : 0.32

Class: 01459, Confidence: 0.32

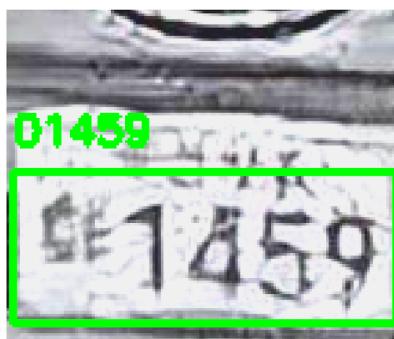


Plate 2 : 1211

Class : 313

Confidence : 0.14

Class: 313, Confidence: 0.14

2. 문제점

- ESRGAN을 이용하고도 화질이 좋지 않은 이미지인 경우, OCR만으로 정확한 값을 얻기 어려움
=> 번호판의 각 번호를 예측하는 정확도가 높은 모델 개발 필요
- 번호판에서 각 숫자를 정확하게 추출하기 위한 CNN 기반 모델 고려
=> CNN을 이용하여 번호판 이미지를 입력으로 받고, 번호판의 각 숫자를 예측하는 모델 개발 진행

7. CNN

Fast-R-CNN, Mask-R-CNN을 사용하여 번호판 번호 예측 모델을 학습한다.

학습 과정은 다음과 같다.

1. 데이터 라벨링(roboflow)

- 클래스 : 10개 (0 ~ 9까지의 숫자)
- 총 데이터 : ESRGAN 번호판 사진 1546장 (train : 1353장, val : 129장, test : 64장)

2. 데이터 전처리

- 이미지 크기 : 224x224 조정
- 회전 : 좌우 15° 방향
- 전단 변환 : 상하좌우 15° 방향
- 밝기 : -10% ~ 10%
- 흐림 : 최대 1.5px
- COCO & tf_record 형식으로 다운



3. 문제 발생

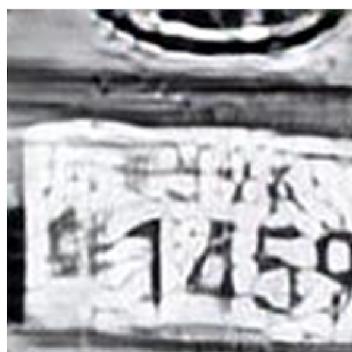
- 모델의 학습이 제대로 진행되지 않는 현상 발생
- loss값이 0만 반복해서 출력하는 현상 발생

4. 해결 방안

- CNN을 기반으로 하되 Roboflow 사이트에서 학습 진행
- Model Type : Roboflow 3.0 Object Detection (Fast)
- Checkpoint : COCO

5. 학습 결과

- 각 class의 문자를 합치고 평균 confidence로 전환
- OCR보다 인식률이 높고 정확한 결과값 도출



class: 1, confidence: 0.87
class: 4, confidence: 0.94
class: 5, confidence: 0.95
class: 9, confidence: 0.76

Class : 1459

Confidence : 0.88

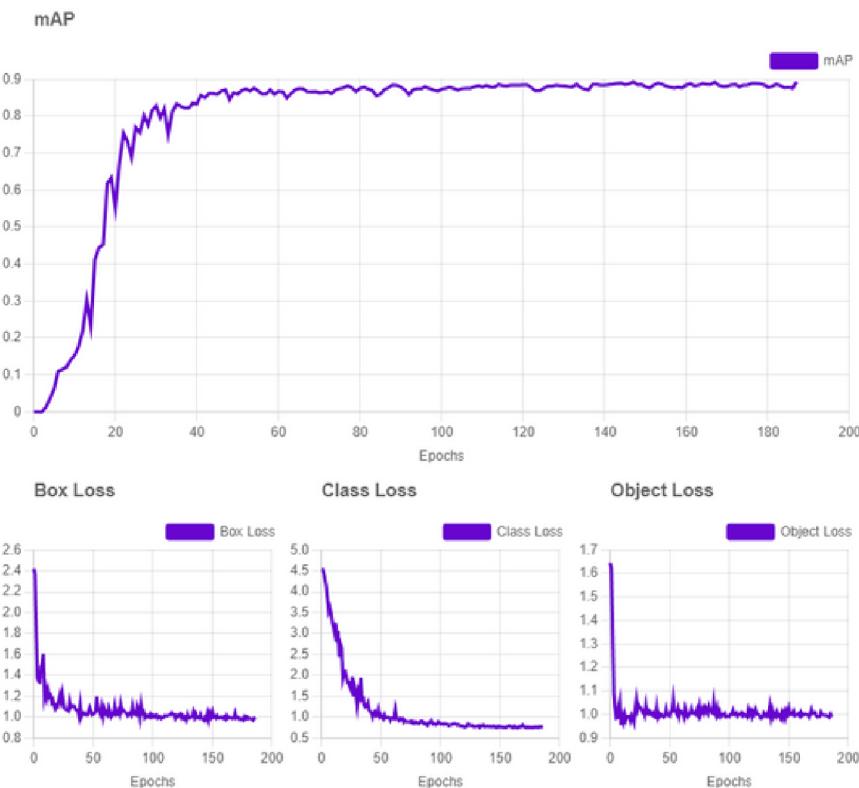


class: 1, confidence: 0.66
class: 2, confidence: 0.89
class: 1, confidence: 0.58
class: 1, confidence: 0.66

Class : 1211

Confidence : 0.70

Training Graph



7. 결과 해석

- positive : 양성, 모델이 이미지나 비디오에서 올바르게 감지하고자 하는 객체
- precision (86.3%) : 정밀도, 모델이 positive로 예측한 객체 중 86.3%가 실제로 해당 객체인 경우
- recall (84.4%) : 재현율, 모델이 실제 positive 객체 중 84.4%를 올바르게 감지한 경우
- mAP (89.3%) : 클래스에 대한 평균 precision
- F1-score (85.3%) : precision, recall의 조화평균
- 고성능 모델, precision, recall을 고루 고려하여 양호한 결과를 도출하고 있는 것을 확인

8. 그래프 해석

- mAP 그래프
 - 학습 중 모델의 성능 변화를 시각적으로 보여주는 그래프
 - 학습 시간에 따른 mAP의 변화를 추적하여 모델이 어떻게 향상되는지를 확인
 - 높은 mAP 값이 지속적으로 유지 중 (이상적인 결과)
- Box Loss 그래프
 - 객체 위치와 경계 상자에 대한 손실을 나타내는 그래프
 - 모델이 객체 위치를 정확하게 예측하고 경계 상자를 학습하는 데 효과적임을 나타냄
- Class Loss 그래프
 - 클래스 예측에 대한 손실을 나타내는 그래프
 - 모델이 객체의 클래스를 정확하게 분류하는 데 효과적임을 나타냄
- Object Loss 그래프
 - 객체의 존재 여부를 예측하는 데 사용되는 손실을 나타내는 그래프
 - 모델이 객체의 존재 여부를 정확하게 파악하는 데 효과적임을 나타냄

8. 프로젝트 결과

RESULTS

100장의 무작위 현장 사진 샘플을 각 모델에 적용하였다. 적용 결과를 사전 DB와 비교하여 실제 정확도를 비교하였다.

1. YOLOv5 : 96%

- 현장 사진 (100장) 중 번호판 식별 (96장), 식별 불가 (4장)

2. ESRGAN : 100%

- 번호판 사진 (96장) 중 96장 모두 적용 (화질 증가)

3. OCR : 00%

- 개선된 번호판 사진 (96장) 중 번호 인식 (00장), 식별 불가 (00장)
- 인식된 번호 (00장) 중 정답 (00장), 오답 (00장)

4. CNN : 00%

- 개선된 번호판 사진 (96장) 중 번호 인식 (00장), 식별 불가 (00장)
- 인식된 번호 (00장) 중 정답 (00장), 오답 (00장)

9. 프로젝트 소감

OPINION

1. 예상치 못한 복잡성

- 이미 오래 전부터 연구되어 온 분야였기 때문에 비교적 간단하게 처리될 것이라는 안일한 생각
- 실제로 데이터를 받아 보니 이 문제는 예상보다 훨씬 복잡한 문제
- 특히 데이터 전처리 과정에서 많은 어려움 발생

2. 데이터의 중요성

- 프로젝트의 성공과 미래 프로젝트에 대한 마음가짐을 크게 변화
- 초기에는 이미 존재하는 데이터셋을 활용하여 모델을 개발하려고 노력했으나 실패하여 좌절
- 기업으로부터 데이터를 제공받은 이후, 모델의 학습 결과가 향상되어 안정감을 가졌음
- 데이터의 양과 질을 향상시키기 위한 기업의 투자하는 노력과 데이터 관리의 중요성을 알게 되었음
- 데이터 확보에 대한 더 신중하고 철저한 접근을 채택할 것을 다짐

3. 연구/개발에 대한 인내와 노력

- 모델의 성능을 높이기 위해 끊임없이 실험하고 데이터를 분석하는 작업은 지루할 수 있음
- 모델의 성능을 향상시킨 순간, 이 모든 인내와 노력이 얼마나 가치있고 의미 있는 일이었는지 되돌아봄
- 이를 통해 느낀 성취감은 노력의 모든 가치를 대변하는 순간이었다고 느낌

4. 느낀 점

- 다양한 환경 조건에 대한 대응력이 필요
- 문제를 해결하기 위해서는 고도의 기술과 데이터가 필요
- 기술력 향상과 데이터 관리의 중요성 인지