

1. 업무수행 범위

1. 반도체 결함 검출 프로젝트 (HLSAM) 고도화

- 사용성 개선, 기능 개발을 위해 반도체 결함 검출 프로젝트 우선 개발
- 결함 검출 기능 모듈화 및 형상 분리
- 수행 기간: 2024.3월 초 ~ 2024.7월 초

2. METAIEYES 시계열 데이터 분석 서버 고도화

- 시계열 데이터 분석 서버의 재사용을 위한 모듈 추상화 필요성 식별
- 근처 디바이스들을 조절하여 목표 디바이스의 미래 값을 원하는 방향으로 조절하는 기능 구현
- 수행 기간: 2024.7월 중 ~ 8월 중

1. 업무수행 범위 - 반도체 결함 검출 프로젝트 (1/2)

반도체 결함 검출 프로젝트 목표

1. 다양한 소자/결함에 대응하는 검출 기능

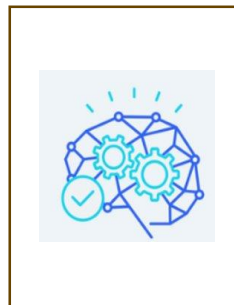
- 사용자의 소자, 결함 이미지 데이터를 이용하여 AI를 재학습
- 새로운 유형의 소자, 결함에 대응

2. 비 개발자가 사용 가능한 AI

- 결함 검출 AI 학습에 필요한 세부 설정을 자동으로 진행
- 사용자는 결함 데이터셋만 제공하면 AI 학습 가능



- 소자, 결함 데이터셋 제공



- 소자, 결함 검출 모델 생성



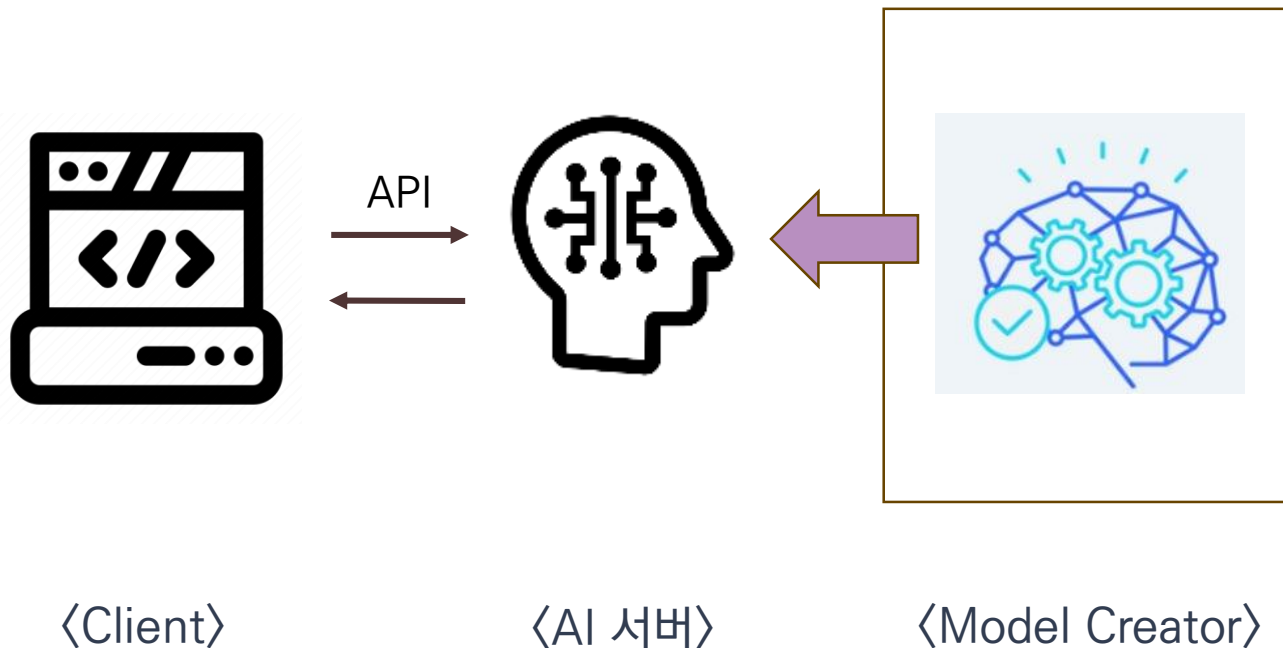
소자, 결함 데이터 라벨링

소자, 결함 학습 진행

결함 검출 수행

1. 업무수행 범위 - 반도체 결함 검출 프로젝트 (2/2)

반도체 결함 검출 프로젝트 개요



Model Creator

: 신규 소자/결함에 대한 모델 생성 및 추가 데이터 이용한 추가 학습

AI 서버

: 사용자 Application으로 부터 이미지를 전달받고, 모델로 추론, 추론한 결과를 사용자 Application에 전달

사용자 Client

: AI 서버에 검출 목표 사진을 전달하고, 추론 데이터를 전달받은 뒤 검출 결과를 분석. 사용자에게 결함 정보를 제공

1. 업무수행 범위 – METAIEYES 시계열 데이터 분석 서버

시계열 데이터 분석 서버 고도화

- METAIEYES 빌드 변경에 따른 모듈화 및 구현사항 수정 필요성 식별
- 데이터 시각화 및 AI 데이터 분석 (REQ-007) 요구사항 구현 진행
 - 유사도와 상관관계 기반으로 추이를 분석하고, 주변 환경을 최적화 할 수 있는 방법 제시

REQ-007	데이터 시각화 및 AI 데이터 분석	시스템은 수집한 데이터의 시각화를 제공한다.
		사용자는 시각화한 데이터의 가시화 모드를 제어할 수 있다.
		시스템은 수집한 데이터간 유사도와 상관관계를 분석하여 사용자에게 제공한다.
		시스템은 수집한 데이터로 향후 추이를 분석하여 사용자에게 제공한다.
		시스템은 수집한 데이터로 이상치를 탐지하여 사용자에게 제공한다.
		시스템은 분석한 데이터를 대시보드 형태로 전시한다.

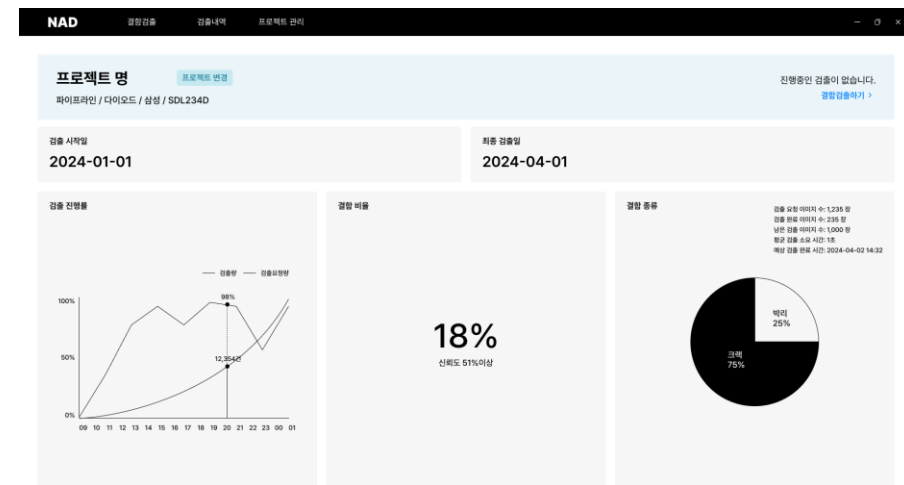
2. 업무수행 내용 - 반도체 결함 검출 프로젝트 (1/10)

반도체 결함 검출 프로젝트 (HLSAM) 개발 및 고도화

- 디자인 및 UI/UX 개선 진행
- 모델 학습 기능/적용 설정의 고도화 진행하고, 분석 기능 고도화하여 개선된 인사이트 제공



〈기존 화면 설계〉



〈변경된 화면 설계〉

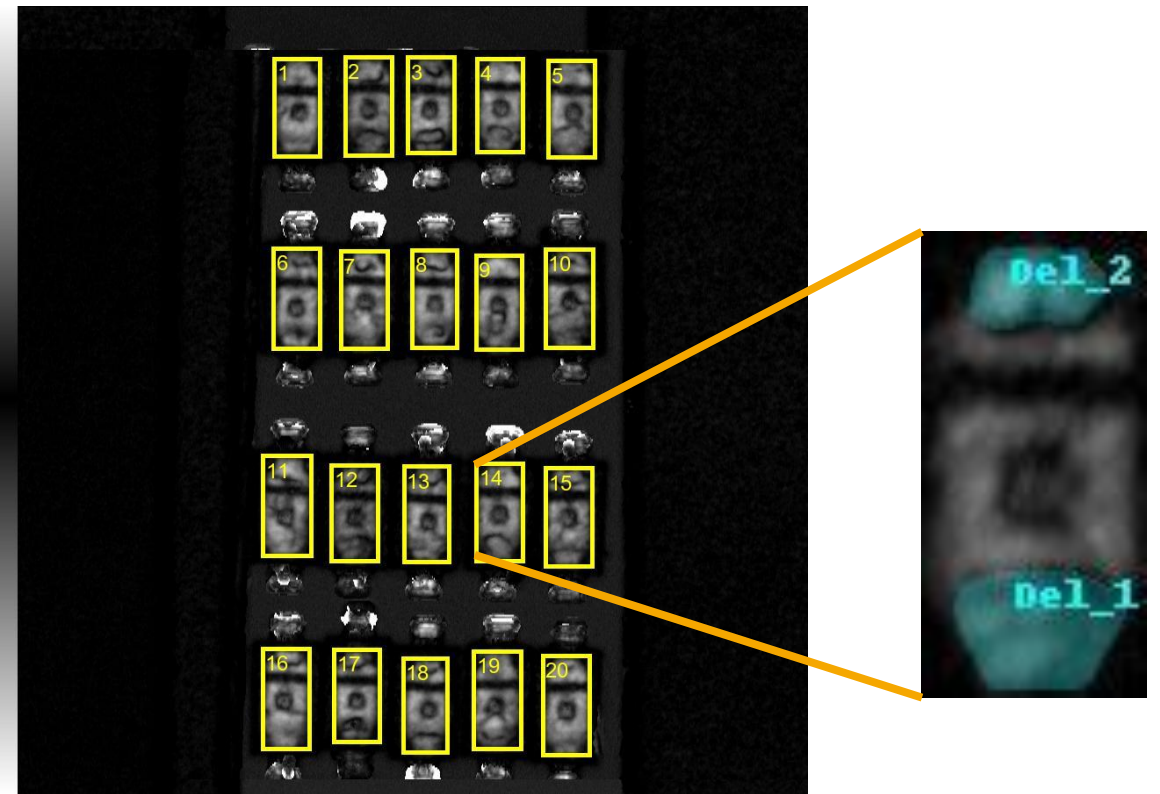
2. 업무수행 내용 - 반도체 결함 검출 프로젝트 (2/10)

초음파 이미지 기반 결함 예측

- 반도체의 초음파 이미지로부터 결함의 종류와 위치를 검출하는 기능 개발

기능

- 반도체의 결함 종류, 위치, 통계 분석
- 결함 데이터 직접 추가를 통한 결함 검출 성능 향상

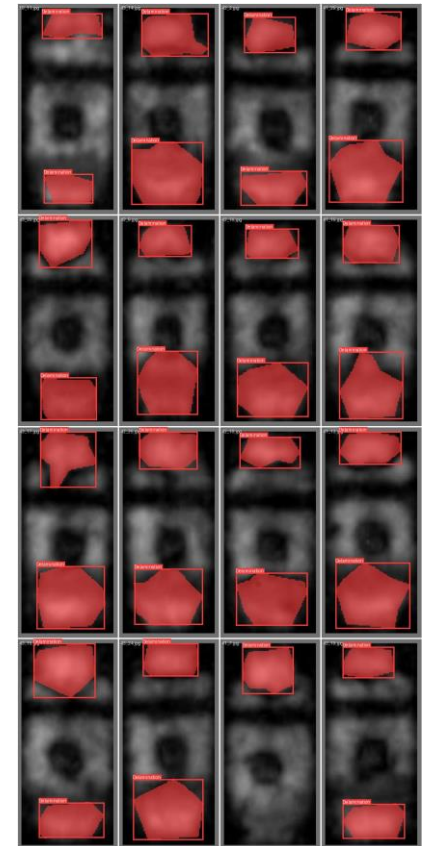
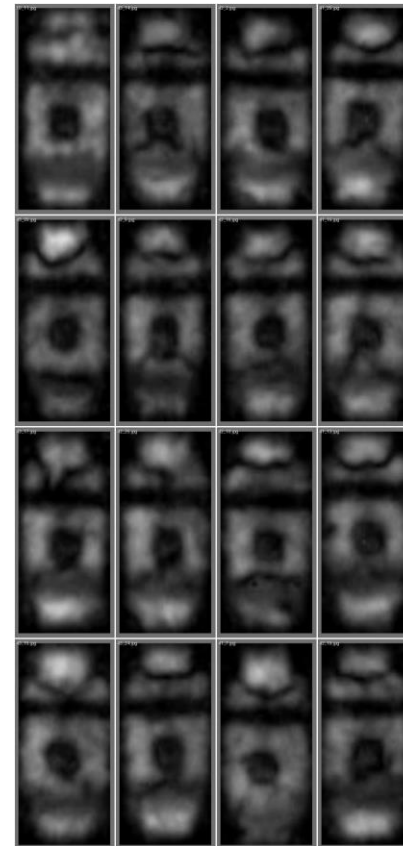


2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (3/10)

소자 / 결함 검출 AI 선정

- 검출 모델: yolov8 모델
- 각 소자, 결함에 대해 학습하여, 각 소자/결함의 종류와 위치를 판별
- 빠른 학습 속도와 검출 속도 가지며, 준수한 검출 성능 보유

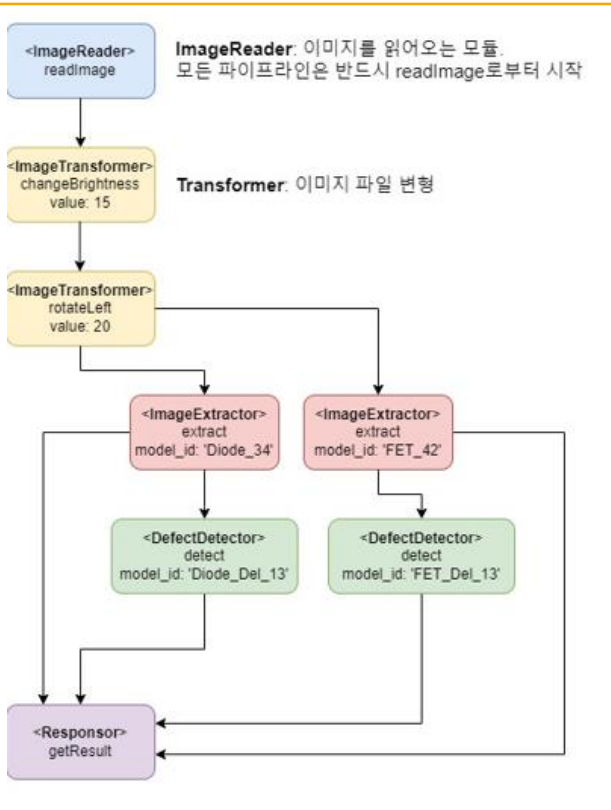
검출 성능	달성 정도	비고
검출 속도	0.38초	검출 장비(GTX 1080 Ti기준)
다이오드 – 박리 검출률	94%	
FET – 박리 검출률	82%	



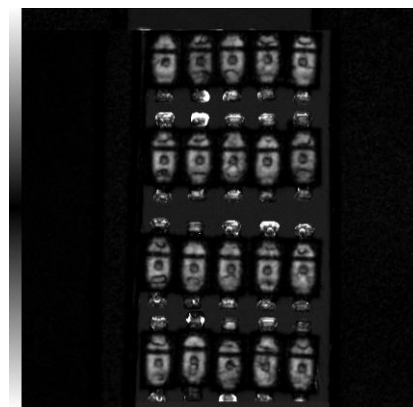
2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (4/10)

모델 파이프라인

다양한 소자, 결함을 검출하기 위해, 복수개의 AI 모델을 조합
소자 종류, 결함 종류를 설정하여 학습 -> 개별 검출 정확도 향상

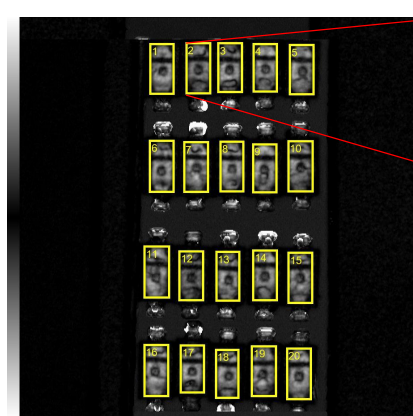


〈모델 파이프라인〉



<ImageReader>
readImage

소자 탐지



<ImageExtractor>
extract
model_id: 'Diode_34'

소자 이미지
분리



결함 탐지

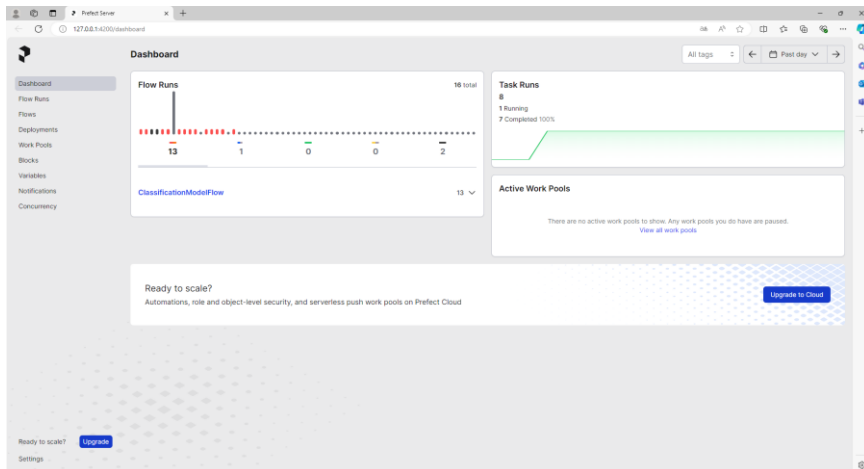


<DefectDetector>
detect
model_id: 'Diode_Del_13'

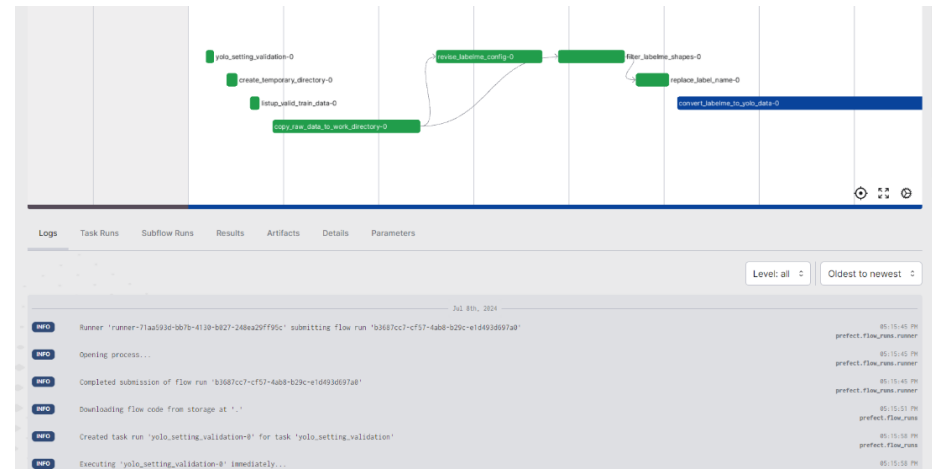
2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (5/10)

Model Creator

- 비 개발자의 AI 학습 지원을 위해, 오픈소스를 활용한 UI 제공
- 신규 소자/결함에 대해 학습하고, 모델 파이프라인 생성
- 비 개발자 사용을 고려하여, 파이썬 설치 없이 실행할 수 있는 Embeddable python 활용 -> Standalone으로 개발



Model Creator 메인 화면

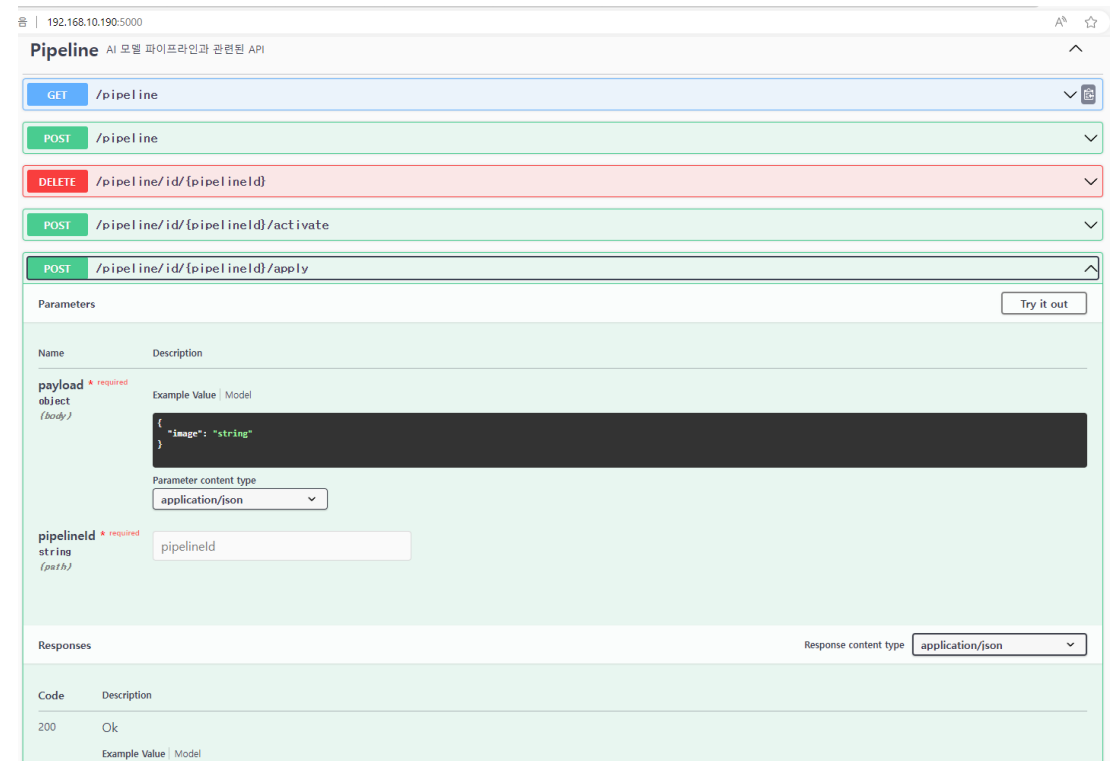
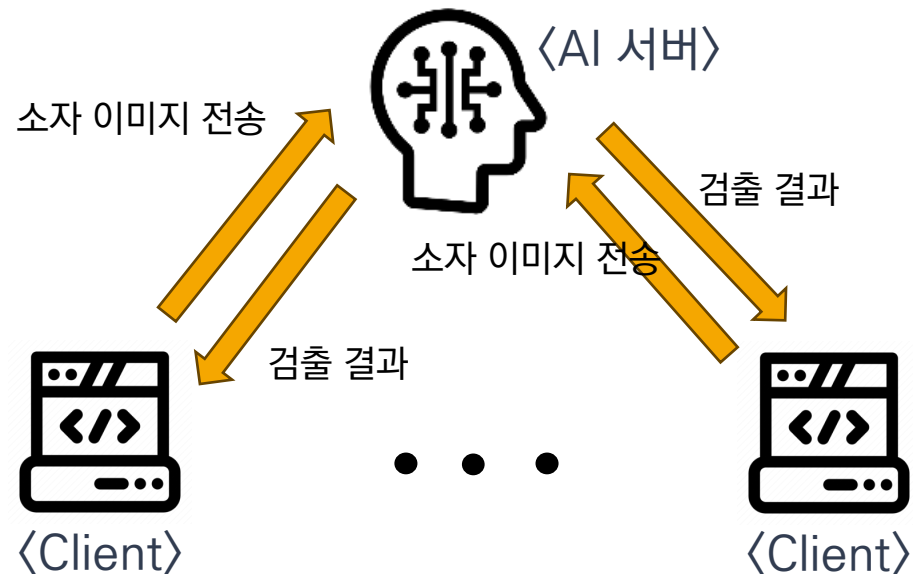


학습 진행 상황 전시

2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (6/10)

AI 서버

- Flask 라이브러리로 API 서버 구현
- Client와 통신하며, 결함 검출 결과를 전송
- 성과: 사용자 장치의 부담 최소화, 결함 검출 기능의 병렬화



2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (7/10)

사용자 Client

- 파이썬의 PySide6 라이브러리로 GUI 구현
- 단순한 조작으로 검출 결과 획득 가능



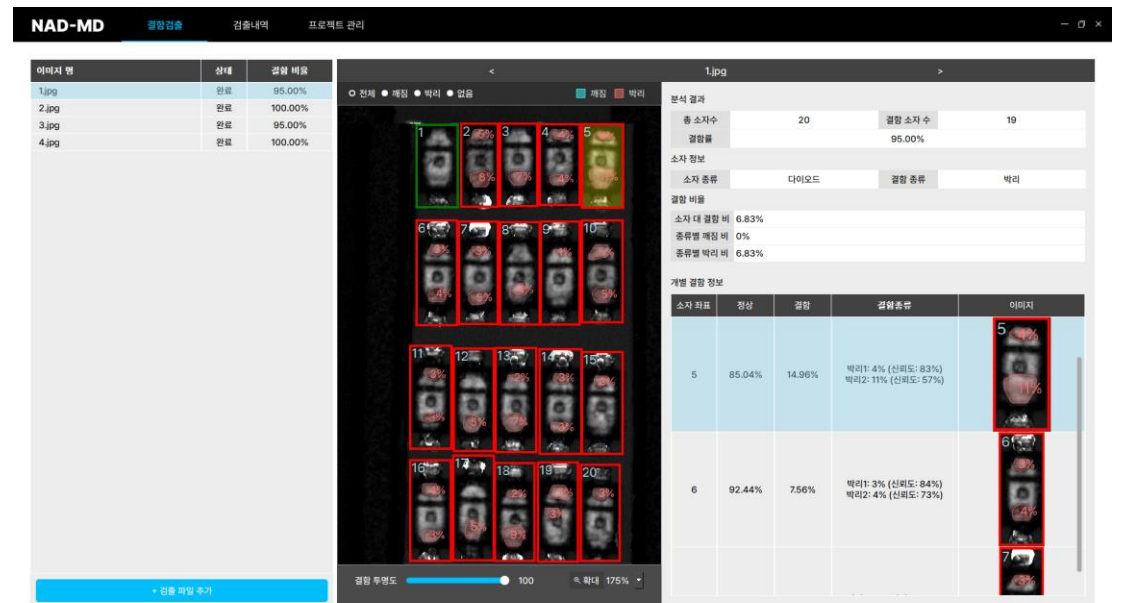
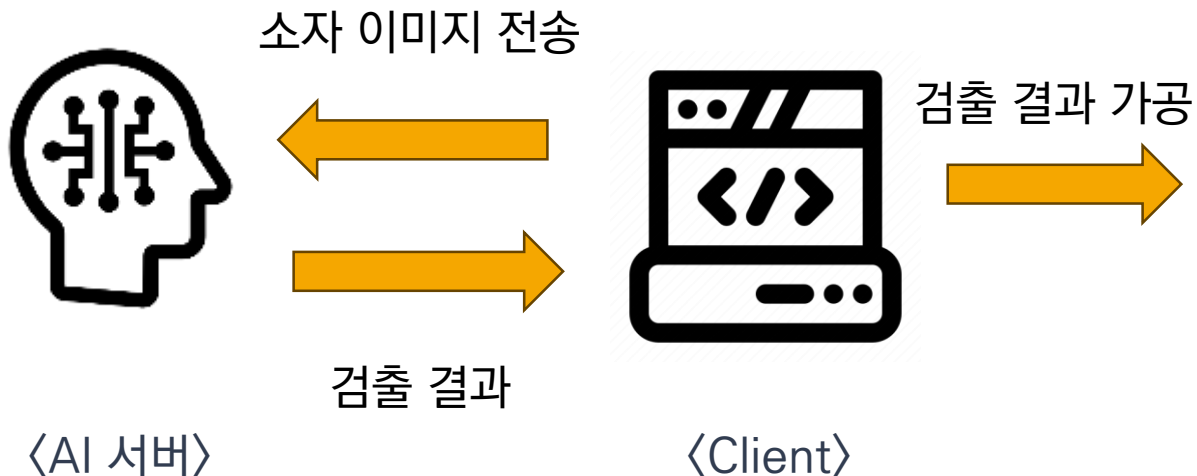
결함 검출 프로젝트 설정



2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (8/10)

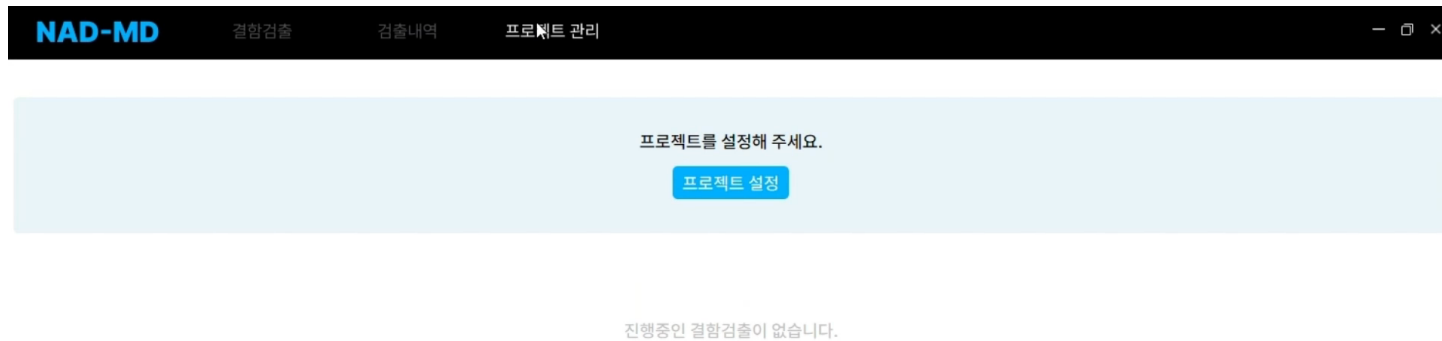
사용자 Client

- AI 서버에 이미지를 전송하고 검출 결과를 획득
- 결함 검출 결과를 가공하여 이미지 및 통계로 재구성



검출 결과에 기반한 이미지/통계 전시

2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (9/10)



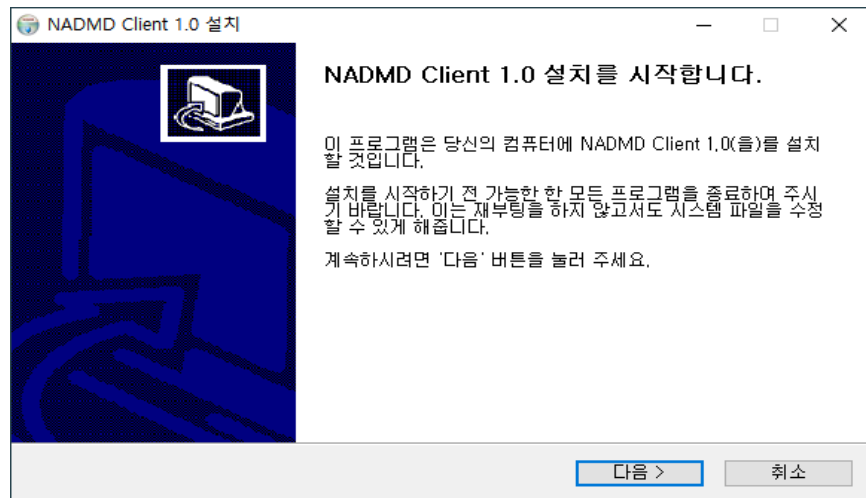
검출 결과를 재가공하여, 사용자에게
인사이드를 줄 수 있는 정보로 변형

- 소자 수
- 결함률
- 결함 종류
- 결함 면적 비
- 결함 이미지 생성 및 전시

2. 업무수행 내용 – 반도체 결함 검출 프로젝트 (10/10)

Client 파일 생성

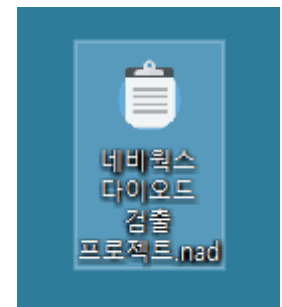
- NSIS 설치 마법사 이용하여 레지스트리 수정
- .nad 확장자 파일을 열어서 프로그램 실행 가능하도록 연결



Client 설치 마법사



레지스트리 수정

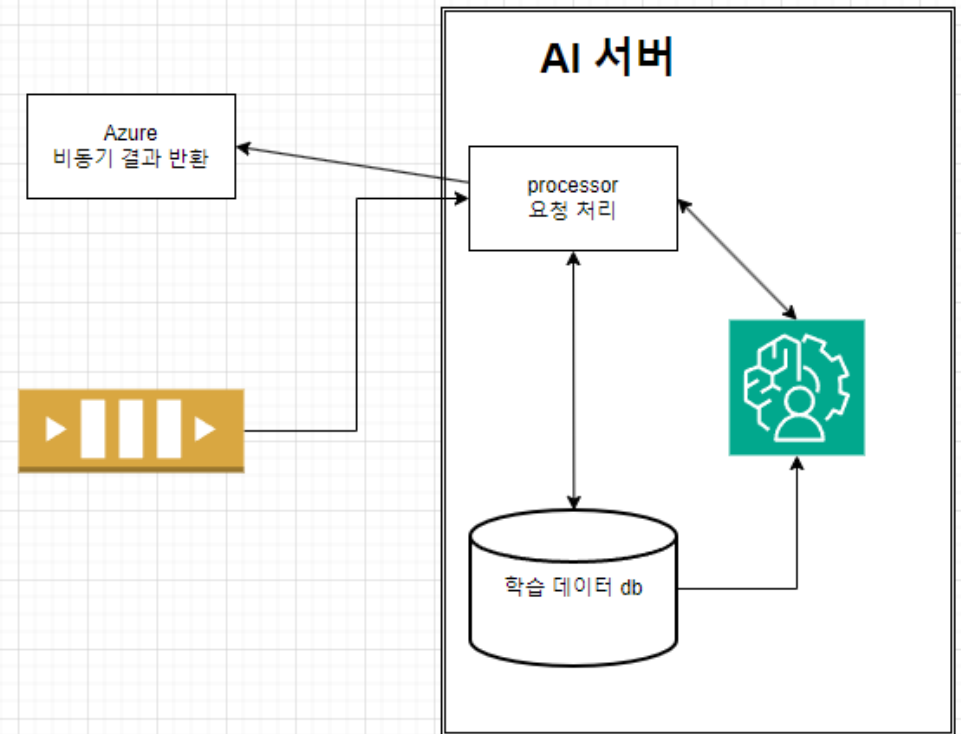


Client를 열 수 있는
확장자 파일 등록

2. 업무수행 내용 - 시계열 데이터 분석 서버 (1/3)

METAIEYES 시계열 데이터 분석 서버 고도화

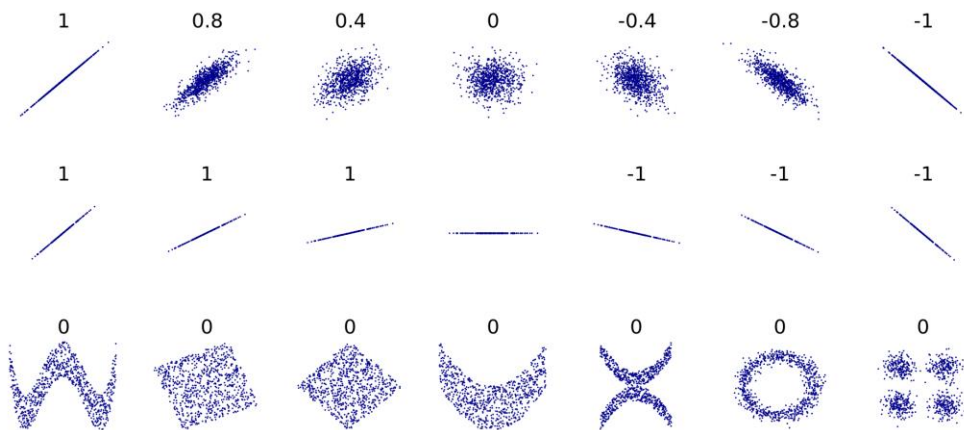
- IoT 분석 ai 서버 모듈화
 - AI 서버 구조 및 호출부 추상화
 - > 타 프로젝트에서 재사용하기 용이하도록 수정
- 데이터 시각화 및 AI 데이터 분석 (REQ-007)
 - 연관 디바이스 최적화 기능 구현



2. 업무수행 내용 - 시계열 데이터 분석 서버 (2/3)

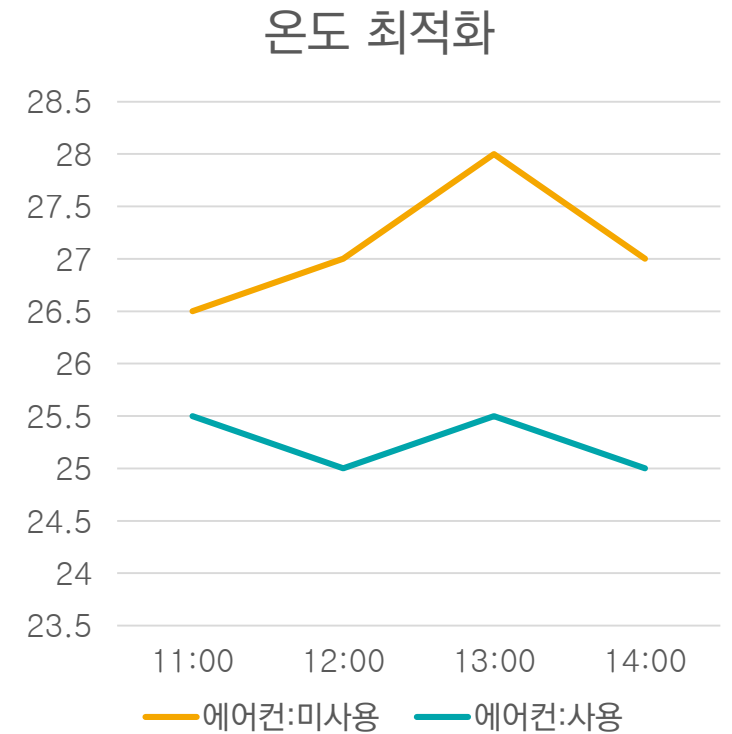
METAIEYES 시계열 데이터 분석 서버 고도화

- 디바이스 최적화 기능 (REQ-007 : 데이터 시각화 및 AI 데이터 분석)
 1. 상관계수 분석을 통한 연관 디바이스 탐색
 2. 연관 디바이스 값을 조절하여 목표 디바이스의 값을 예측



디바이스 간 상관관계 분석

상관계수 기반 변화 예측



연관 디바이스 변화에 따른
예측 생성

2. 업무수행 내용 – 시계열 데이터 분석 서버 (3/3)

METAIEYES 시계열 데이터 분석 서버 고도화

- 디바이스 최적화 기능 (REQ-007 : 데이터 시각화 및 AI 데이터 분석)
외부의 분석요청을 받아, 최적화 방법과 최적화시 예측 시계열 데이터를 생성

Produce Message Close

Partition
Partition #0

Key Serde
String

Value Serde
String

☐ Keep contents

Key
1

Value
1- [{"resultId": "uuid_format",
2 "groupId": "A",
3 "objectIds": ["8"],
4 "aggregation": "mean",
5 "objective": "target",
6 "targetValue": 2850,
7 }]
8]

Headers
1 {}

Produce Message

요청 메시지 전송



```
*****
Process 'optimization' Done. Save Result name 'test_result_id.json'
ConsumerRecord(topic='mie-aiot-optimize-requesting', partition=0, offset=8, timestamp=1723625384594, timestamp_type=0, key=None, value={
  'resultId': 'uuid format', 'groupId': 'A', 'objectIds': ['8'], 'aggregation': 'mean', 'objective': 'target', 'targetValue': 2850, 'headers': [], 'checksum': None, 'serialized_key_size=-1, serialized_value_size=151, serialized_header_size=-1)
ai-process-analysis:uuid format:body
Run Process 'optimization' with id 'uuid_format'
RESULT *****
{'statistic': {'individual': [{'graph': {'timestamps': [1715216400, 1715220000, 1715223600, 1715227200, 1715230800, 1715234400, 1715238000, 1715241600, 1715245200, 1715248800, 1715252400, 1715256000, 1715259600, 1715263200, 1715266800, 1715270400, 1715274000, 1715277600, 1715281200, 1715284800, 1715288400, 1715292000, 1715295600, 1715299200, 1715302800], 'data': [4384.476567333333, 4239.434952, 4090.9866165, 3919.10147775, 3784.541012, 3623.167446, 3113.0583915, 3104.62823225, 3293.3035605, 3593.74828400000002, 3969.1137668, 4210.20765425, 4395.717666, 4570.770922, 4684.978261, 4776.9622032, 4834.3356665, 4852.0, 4880.709981999999, 4925.997109, 4948.265774, 4980.912078, 4960.082446, 4681.062690333333, 3243.8333333333335], 'average': 4242.45584369, 'sum': 106061.39609225, 'min': 3104.62823225, 'max': 4980.912078, 'risk': '안전', 'objectId': '8'}]}, 'total': {'graph': {'timestamps': [1715216400, 1715220000, 1715223600, 1715227200, 1715230800, 1715234400, 1715238000, 1715241600, 1715245200, 1715248800, 1715252400, 1715256000, 1715259600, 1715263200, 1715266800, 1715270400, 1715274000, 1715277600, 1715281200, 1715284800, 1715288400, 1715292000, 1715295600, 1715299200, 1715302800], 'data': [4384.476567333333, 4239.434952, 4090.9866165, 3919.10147775, 3784.541012, 3623.167446, 3113.0583915, 3104.62823225, 3293.3035605, 3593.74828400000002, 3969.1137668, 4210.20765425, 4395.717666, 4570.770922, 4684.978261, 4776.9622032, 4834.3356665, 4852.0, 4880.709981999999, 4925.997109, 4948.265774, 4980.912078, 4960.082446, 4681.062690333333, 3243.8333333333335], 'average': 4242.45584369, 'sum': 106061.39609225, 'min': 3104.62823225, 'max': 4980.912078, 'risk': '안전'}, 'prediction': {'individual': [{'graph': {'timestamps': [1715302800, 1715306400, 1715310000, 1715313600, 1715317200, 1715320800, 1715324400, 1715328000, 1715331600, 1715335200, 1715338800, 1715342400, 1715346000, 1715349600, 1715353200, 1715356800, 1715360400, 1715364000, 1715367600, 1715371200, 1715374800, 1715378400, 1715382000, 1715385600], 'data': [4399.911720118289, 4130.9105205900805, 3865.35963870722, 3593.444610984693, 331.8345747674794, 3074.2529871091892, 2818.9640743364057, 2576.102673405104, 2350.8465661503833, 2130.63943416652, 1915.0336726514074, 1703.625939054347, 1494.9245806721167, 1274.903519359959, 1047.3253142996657, 813.8983151905788, 579.9360768358416, 344.19277229061805, 107.3927463180092, -133.19011896648544, -405.6076462361704, -726.6025992953283, -1050.9651373123743, -1329.3490975995217]}, 'average': 1579.4910473999178, 'sum': 37907.785137598024, 'min': -1329.3490975995217, 'max': 4399.911720118289, 'risk': '안전', 'objectId': '8'}]}, 'total': {'graph': {'timestamps': [1715302800, 1715306400, 1715310000, 1715313600, 1715317200, 1715320800, 1715324400, 1715328000, 1715331600, 1715335200, 1715338800, 1715342400, 1715346000, 1715349600, 1715353200, 1715356800, 1715360400, 1715364000, 1715367600, 1715371200, 1715374800, 1715378400, 1715382000, 1715385600], 'data': [4399.911720118289, 4130.9105205900805, 3865.35963870722, 3593.444610984693, 3331.8345747674794, 3074.2529871091892, 2818.9640743364057, 2576.102673405104, 2350.8465661503833, 2130.63943416652, 1915.0336726514074, 1703.625939054347, 1494.9245806721167, 1274.903519359959, 1047.3253142996657, 813.8983151905788, 579.9360768358416, 344.19277229061805, 107.3927463180092, -133.19011896648544, -405.6076462361704, -726.6025992953283, -1050.9651373123743, -1329.3490975995217]}
253142996657, 813.8983151905788, 579.9360768358416, 344.19277229061805, 107.3927463180092, -133.19011896648544, -405.6076462361704, -726.6025992953283, -1050.9651373123743, -1329.3490975995217]
*****
```

예상되는 측정값과, 최적화했을 때의 예상 시계열 데이터 생성

분석 요청

3. 업무수행 성과

반도체 결함 검출 프로젝트

1. 요구사항 추적표 작성
2. 결함 검출 분야 해외 기업 벤치마킹 및 분석보고서 작성
3. 소자, 결함 분리/검출 API 서버 구축
4. 각 형상 별 사용자 지침서 작성
5. 결함 검출 프로그램 (NAD-MD) 납품

METAIEYES

1. lot 분석 기능의 재사용을 위한 모듈 추상화
2. 상관계수 기반 디바이스 최적화 기능 구현