졸업 설계

2차 발표

2016156026 이형석
2016156033 조현민
2016156039 최성훈
2016156047 황규빈

목차

CONTENTS

01 작품선정 배경 및 필요성

저비용 CFRP 가공 결함 검출

- 항공, 우주, 자동차 등의 산업에 CFRP가 떠오르고 있다.
- CFRP 소재는 가공이 어려워 결함이 잦게 발생하며, 가공 결함을 검출하는 장비는 매우 고가이기 때문에 규모가 작은 기업에서는 저비용의 결함 검출 방법이 필요하다.

육안보다 더 빠른 방법

- 육안으로 검사하는 방법은 상대적으로 많은 시간이 필요함.
- 가공 후 즉시 카메라를 이용한 결함 검출을 진행한다면, 육안으로 검출하는 방법보다 훨씬 더 빠른 검사 진행이 가능하다

01 기존 연구 분석

한국생산기술연구원	• 광학 스캐너를 활용해 가공된 부품 표면과 내부 불량을 1초 만에 파악하는 '3D 광학 고속 검사 기술'을 세계 최초로 개발
코그넥스	• 자동화 프로세스를 위한 머신 비전 시스템, 소프트웨어, 표면 검사 등을 제조 및 판매
다트비전	• 반도체, 디스플레이, 모바일 및 제조업과 다양한 분야에서 머신 비전 솔루션을 공급
앤비젼	• 형상 및 결함 추적과 표면, 색상 검사 머신 비전 기술 지원 서비스 제공 • XGS 센서 기반 카메라로 수치 측정과 같은 애플리케이션에 활용

01 기술동향 분석

이미지 분석 기술

복잡한 성형 표면 및 결함 처리에 탁월

머신 비전 시스템

제품의 결함, 오염, 기능 결점, 이상을 감지하기 위해 트레이닝

기존의 알고리즘으로 처리 불가능한 검사, 분류 및 위치 지정 애플리케이션을 해결

국방, 항공 우주, 연구, 생체인식, 의료 등에서 머신 비전을 활용 중

01 개발 목표

서비스 개발

- 웹, 앱을 통한 CFRP 가공 이미지 결함 검출 결과 확인
- 핸드폰 카메라를 통한 간이 이미지 결함 검출 앱 개발
- 검출된 결함에 대한 Box 표시 후 재가공 판단

웹 개발

- 페이지 관리자용 웹 개발
- 사용자 결함 검출 이미지 확인용 웹 개발
- 사용자 이미지 업로드 웹 개발

01 개발 목표

서버 개발

- 웹과 딥 러닝 프로그램 사이 이미지 전달 서버 개발

개발 방향

- 빠른 프로토타입 출시를 통해 애자일 방법론을 따라서 추가적인 요구사항에 대처
- 개발 시 선행 지식이 부족한 딥 러닝 Yolo v5 알고리즘을 클라우드 서비스에서 구현
- 사용되거나 표시될 이미지의 서버와 프로그램 간에 송수신에 초점을 맞춘다.
- 데이터 송수신 이후에 웹과 앱의 UI에 대해 초점을 맞춘다.

01 팀 역할 분담

이형석

- 프론트(웹페이지) 구현
- 실시간 검출 히스토리 노출
- 서버 구현
- AWS EC2 서버 구축

조현민

- 이미지 전처리
- YOLO MARK를 활용한 이미지 라벨링

01 팀 역할 분담

최성훈

- 알고리즘 모델 구현
- Yolo v5 기법 구현 및 응용

황규빈

- 알고리즘 모델 구현
- Mask R CNN 기법 구현 및 응용
- 프론트(앱) 구현
- 이미지 업로드 기능 구현
- 결과 반환 및 노출

01 소통방법

개발 방법론	• 점진적 개발의 장점을 살리면서, 요구사항의 변화를 주기적으로 수용하는 애자일 방식 활용 • 2주 단위로 기존에 계획한 요구사항과 전 스프린트에서 나온 변경사항을 비교하고, 검토를 거쳐 우선순위화하여 반영 • 분석, 설계, 구현, 테스트 같은 공정들을 동시적으로 접근하여 중간 산출물을 최소화하면서 품질을 높임
프로젝트 진행관리	• 스프린트 관리를 위해 Trello 도구를 활용 • 협업 및 버전 관리를 위해 GitHub를 활용 • 팀원간 커뮤니케이션을 위해 Slack을 활용
팀룰	• 매주 수요일 스프린트 회의를 진행, 회의 간 프로젝트 진행 상태 및 문제점 파악

01 개발 일정

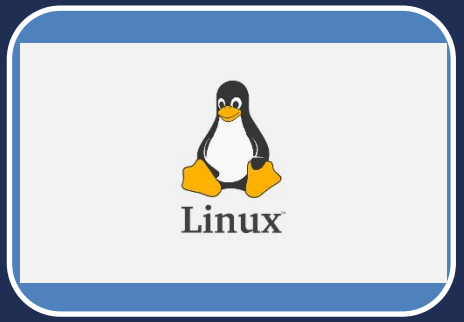
항목	추진 사항	12월	1월	2월	3월	4월	5월	6월	7월	8월
사전 조사 및 요구사항 분석	사전조사 및 요구사항 분석									
시스템 설계 및 상세 설계	데이터 학습에 사용될 알고리즘 선정									
	웹 구조 설계									
	모바일 앱 구조 설계									
	DB 설계									
데이터 수집 및 딥러닝 구현	결함이 있는 CFRP 소재의 이미지 데이터 수집									
	데이터 라벨링									
	라벨링된 데이터를 기반으로 학습									
	결함 검출 정확도 향상									

01 개발 일정

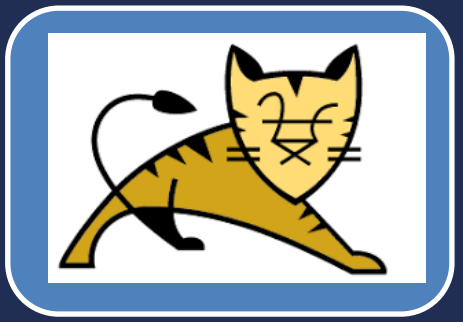
항목	추진 사항	12월	1월	2월	3월	4월	5월	6월	7월	8월
시스템 설계 및 상세 설계	모바일 앱 서버 업로드 및 검출 결과 출력 구현									
	웹 요청에 대한 검출 결과 응답 구현									
	검출 데이터 관리 기능									
	모바일 UI									
테스트	기능 단위 테스트									
문서화	최종 보고서 작성									

01 개발 환경

Server



OS



Web Server

Client OS

개발 도구

Web 개발 도구

02 개발 내용

Step1 개발환경 구축

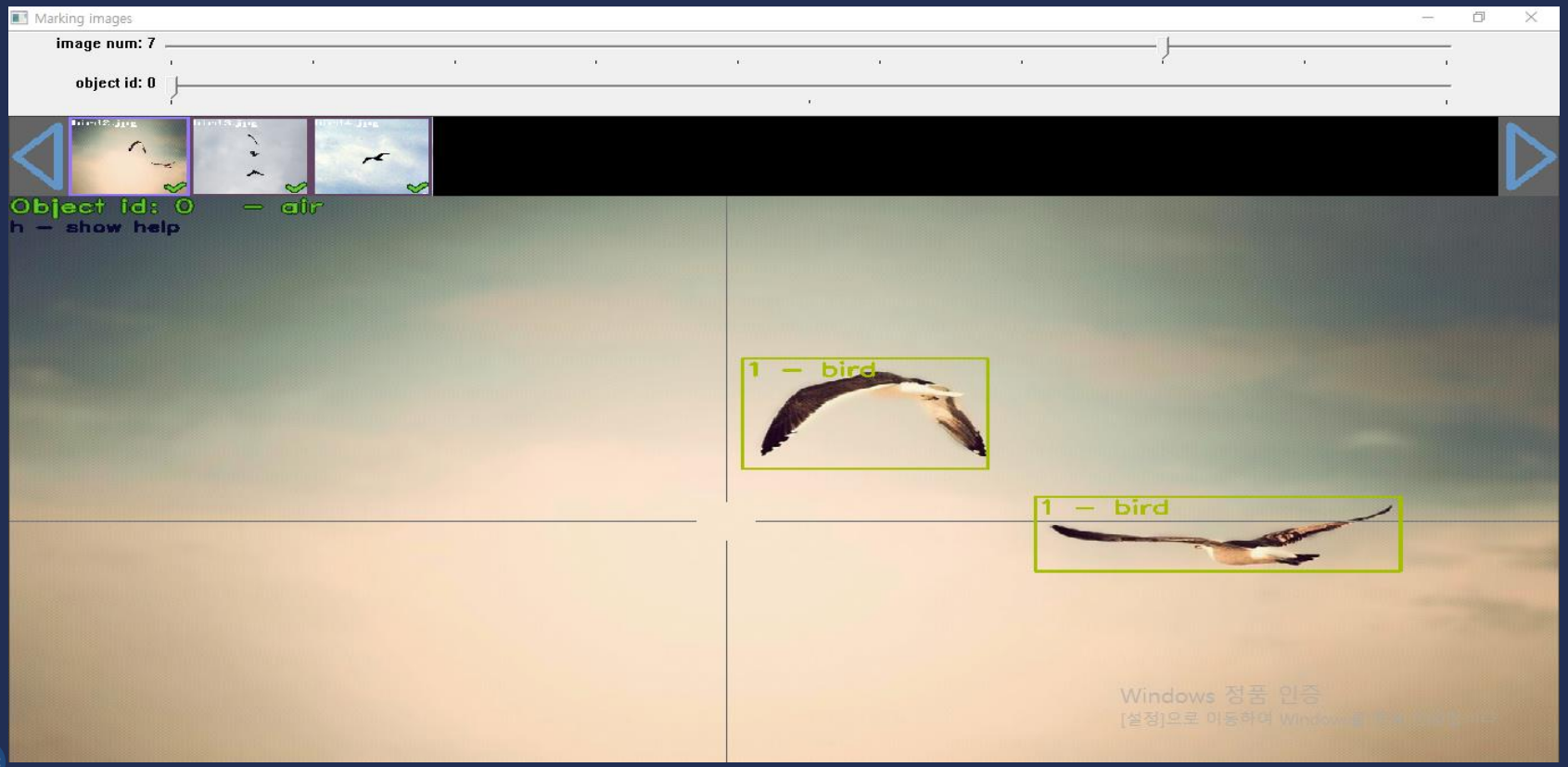
※ 이미지 라벨링과 GPU사용을 통한 학습속도 증진위한 개발환경

- Anaconda3, Python(3.5v이상)
- Pytorch
- CUDA 9.0 version
- OpenCV

02 개발 내용

Step2 Labeling

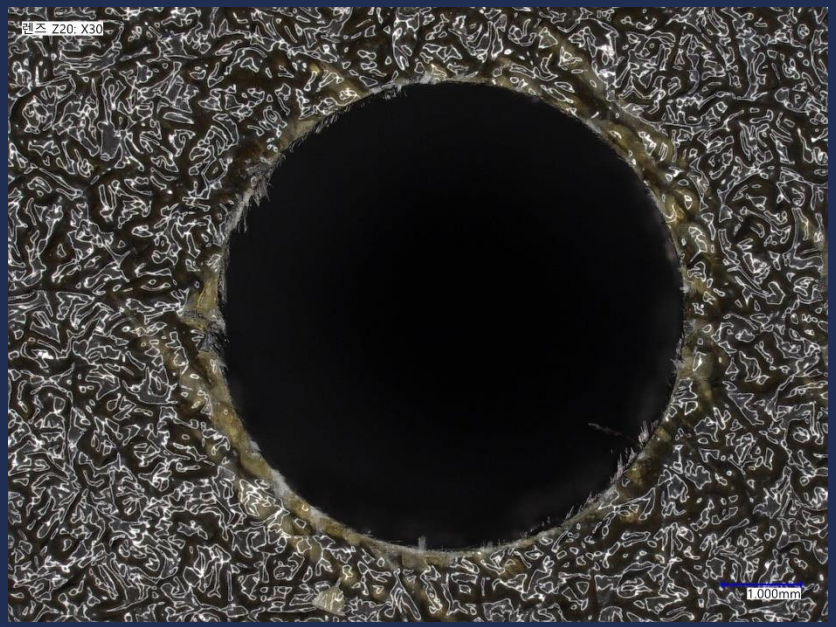
- 객체에 마우스 드래그를 하여 박스처리를 하고 박스처리 된 객체의 좌표를 저장한다.

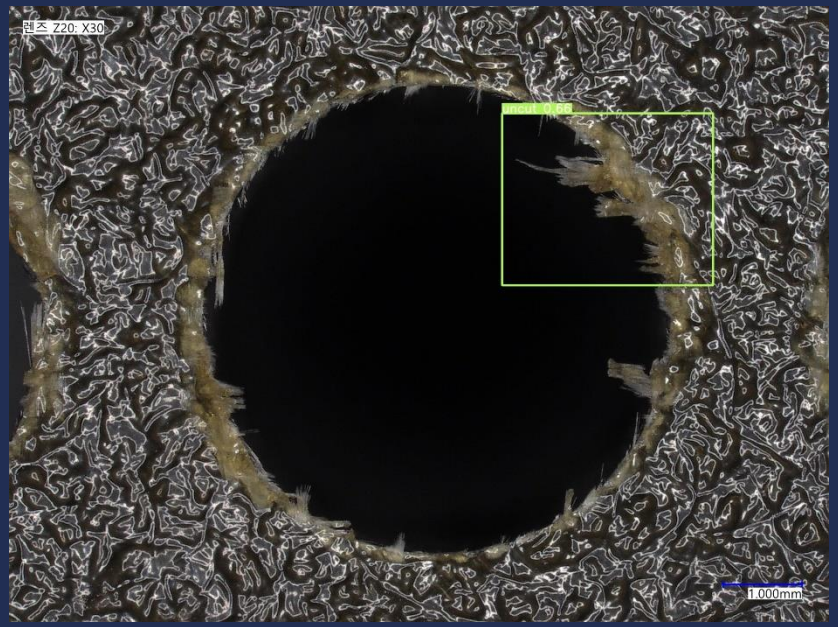


02 개발 내용

Step3 Training

- Step2에서 라벨링한 좌표들을 YOLOv5알고리즘의 파라미터로 전달 해 준다.
- 현재 보유하고 있는 데이터의 일부인 약 500장 가량의 이미지를 학습시켰다.





02 문제 및 해결방안

문제

부족한 Training Data로 인하여 작은 사이즈의 결함부분은 잡아내지 못하여 낮은 결함 검출율을 보인다.

해결방안

현재 소유하고 있는 이미지의 양을 늘리기 위해 전처리를 하여 더 많은 Training Data를 만들어 다시 학습시키면 높은 결함 검출율이 나올 것이라고 기대된다.

02 문제 및 해결방안

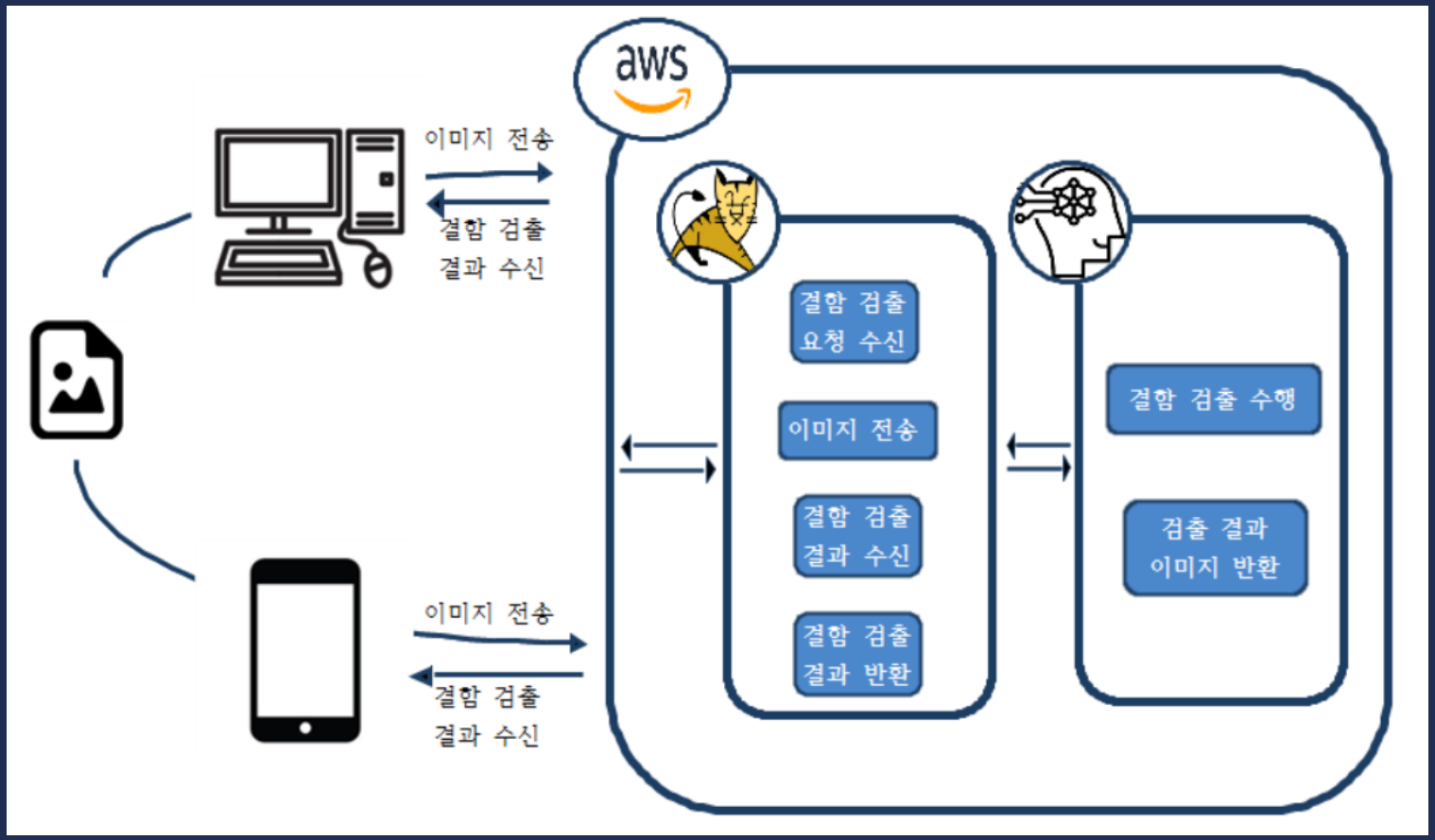
문제

더 높은 검출율과 빠른 응답시간을 구축하기 위해 여러 가지 머신러닝 기법들을 사용 해보고 비교 분석하고 있다. 여기서 실제 머신러닝을 사용하는 회사의 전문가에게 피드백을 받은 기법 중 Mask RCNN의 기법이 있는데 이 기법은 라벨링을 할 때 결함 부분을 정확하게 라인을 그려야 하지만 우리가 검출하고자 하는 결함들의 경계선이 모호하여, 학습을 위해 이미지 라벨링을 하는 과정에서 영역 선정에 어려움을 겪고 있습니다

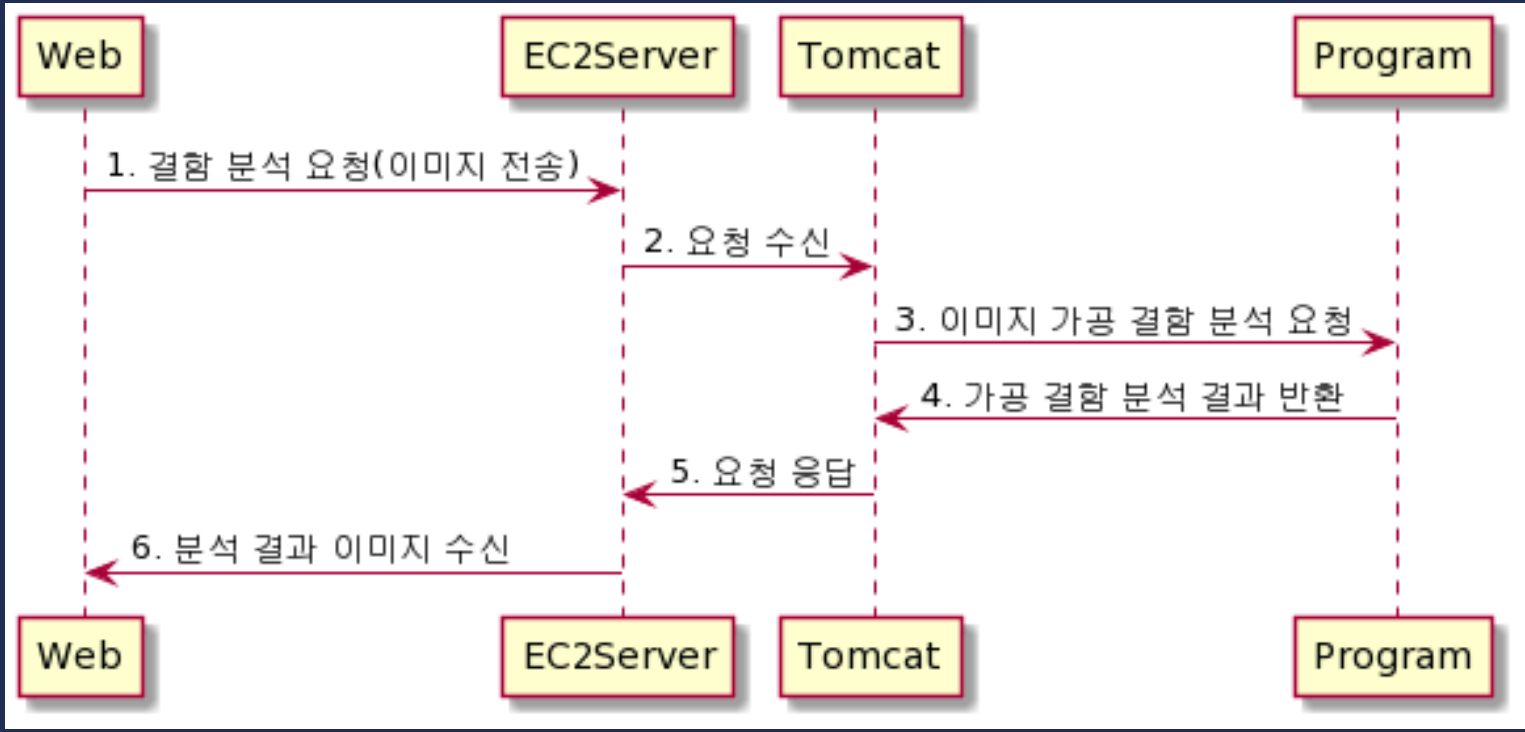
해결방안

위의 문제에 대해 전문가에게 피드백 요청 또는 Mask RCNN처럼 매우 정확한 결함 위치가 도출 되어야 검출에 성공한 것인지 아니면 어느 정도 오차가 있어도 허용이 되는지 만약 오차가 있어도 된다면 허용 범위는 어느 정도인지 한국생산기술연구원의 조언을 구할 예정

02 시험 시나리오

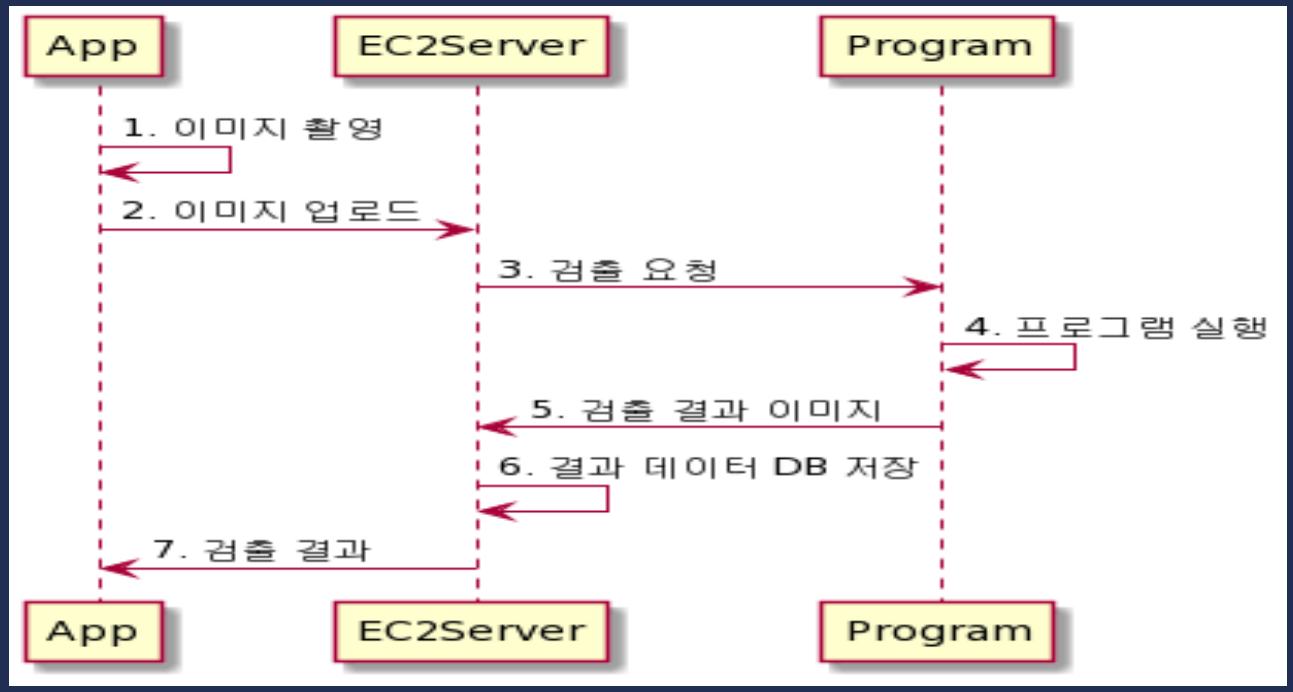

aws
이미지 전송
결함 검출
결과 수신
이미지 전송
결함 검출
결과 수신
결함 검출
요청 수신
이미지 전송
결함 검출
결과 수신
결함 검출
결과 반환
결함 검출 수행
검출 결과
이미지 반환

02 상세 설계 – 웹 시퀀스 다이어그램



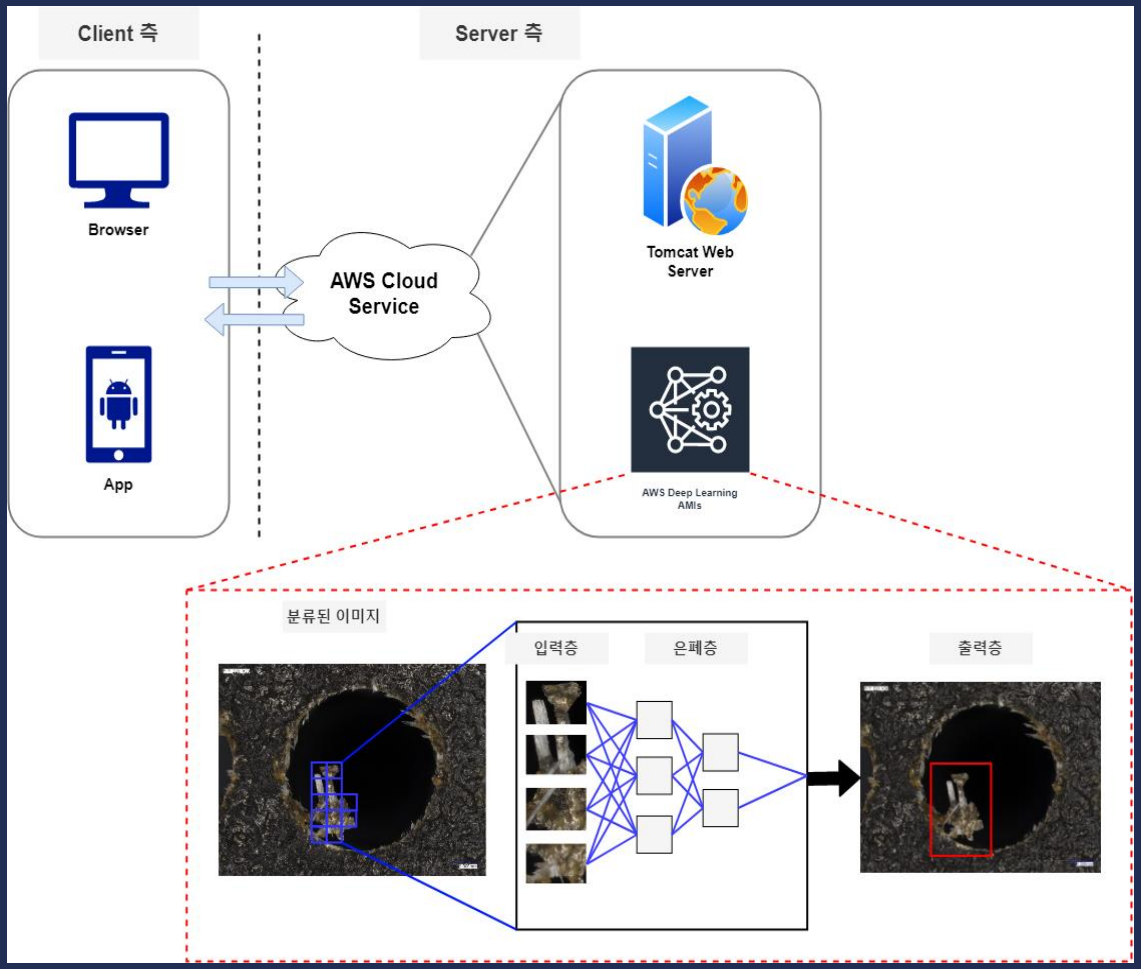
순서	내용
1	웹 페이지에서 이미지를 업로드 받아, 서버로 결함 분석을 요청
2	요청을 받은 서버는, 웹에서의 요청을 수행하기 위해 Tomcat으로 전달
3	Tomcat에서 요청과 함께 전달된 이미지를, 분석을 위해 프로그램으로 전달
4	가공 결함 검출 결과를 Tomcat으로 응답
5	결함 검출 결과 요구에 대한 응답
6	서버는 결함 결과를 Web으로 응답

02 상세 설계 – 앱 시퀀스 다이어그램



순서	내용
1	모바일 앱의 카메라를 이용해 결함 검출 대상을 촬영
2	모바일 앱에서 웹 서버로 이미지를 전송
3	서버는 결함 검출 프로그램을 실행
4	이미지에 결함 여부를 확인해 결함이 있다면 결함 부분에 바운더리 박스를 만들어 새로운 이미지를 생성
5	결함 검출 프로그램의 결과 이미지를 서버에 저장
6	서버는 결과 데이터를 데이터베이스에 저장
7	결함이 검출된 이미지와 검출 결과 데이터를 모바일에 응답

02 상세 설계 – 앱 시퀀스 다이어그램



항목	내용
웹 페이지 준비	● Apache Tomcat 9.0 서버에서 사용자에게 제공할 JSP 페이지 준비
앱 준비	● Android Studio를 통해 사용자가 이미지를 송신하고 수신받을 애플리케이션 준비
AWS Cloud Service 준비	● AWS Cloud Server에는 Tomcat Server와 Deep Learning Server가 존재 ● Tomcat Server를 통해 웹 페이지 서비스 제공 ● Deep Learning Server를 통해 전송 받은 이미지 결함 검출 후 사용자에게 결과 전송
딥러닝 알고리즘 검출률 확인	● CNN Model의 Yolo v5 알고리즘으로 이미지 결함 검출 후 결함 부분에 Boundary Box 표시
결과	● CNN Yolo v5 알고리즘을 통한 딥러닝 훈련으로 낮은 검출률이지만 미절삭 섬유와 박리 현상이 정확히 구분됨 ● 이미지를 송수신하기 위한 애플리케이션 구현, 내부 데이터 송수신 로직은 추후에 구현