

# 딥러닝을 적용한 CFRP 가공 결함 분석

Defect Analysis of CFRP Processing with Deep Learning

2016156026 이형석

2016156033 조현민

2016156039 최성훈

2016156047 황규빈

# 차 례

- 연구 동기 및 관련 연구
- 시스템 수행 시나리오
- 시스템 구성도
- 개발 환경 및 개발 방법
- 업무 분담
- 종합설계 수행일정
- 필요기술 및 참고문헌

# 연구 동기 및 관련연구

## 연구 동기

### Motivation



# KITECH

한국생산기술연구원

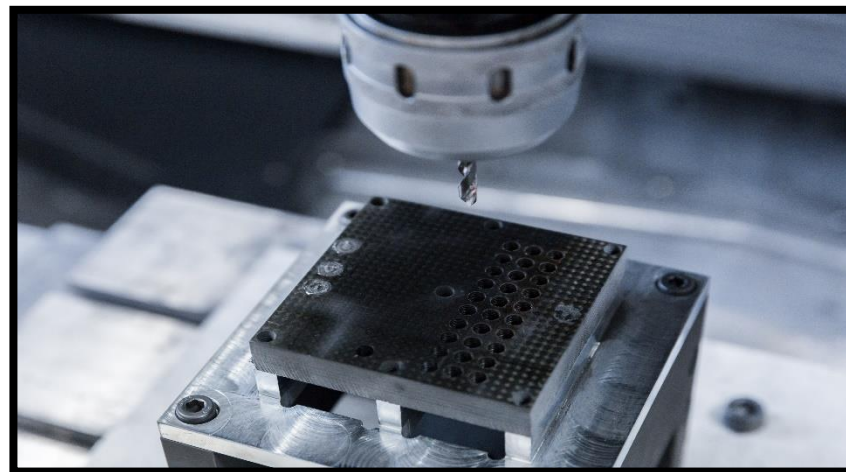
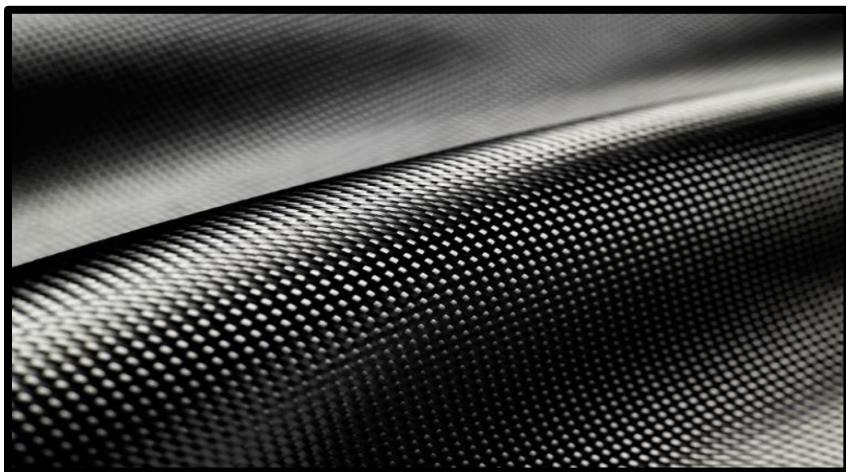
## 한국생산기술연구원 보도자료

- 2015년도부터 가공 기술 개발이 활발해짐
- 항공, 우주, 자동차 등의 산업에 여러가지 장점을 갖는 신소재로 CFRP가 떠오르고 있다는 사실 확인
- 해당 탄소섬유복합재 가공연구는 18년도에 성과를 냈지만 CFRP 소재 특성상 절삭이 어렵고 결함이 자주 발생한다.

# 연구 동기 및 관련연구

## 연구 동기

### Motivation



## CFRP(Carbon fiber reinforced plastic)

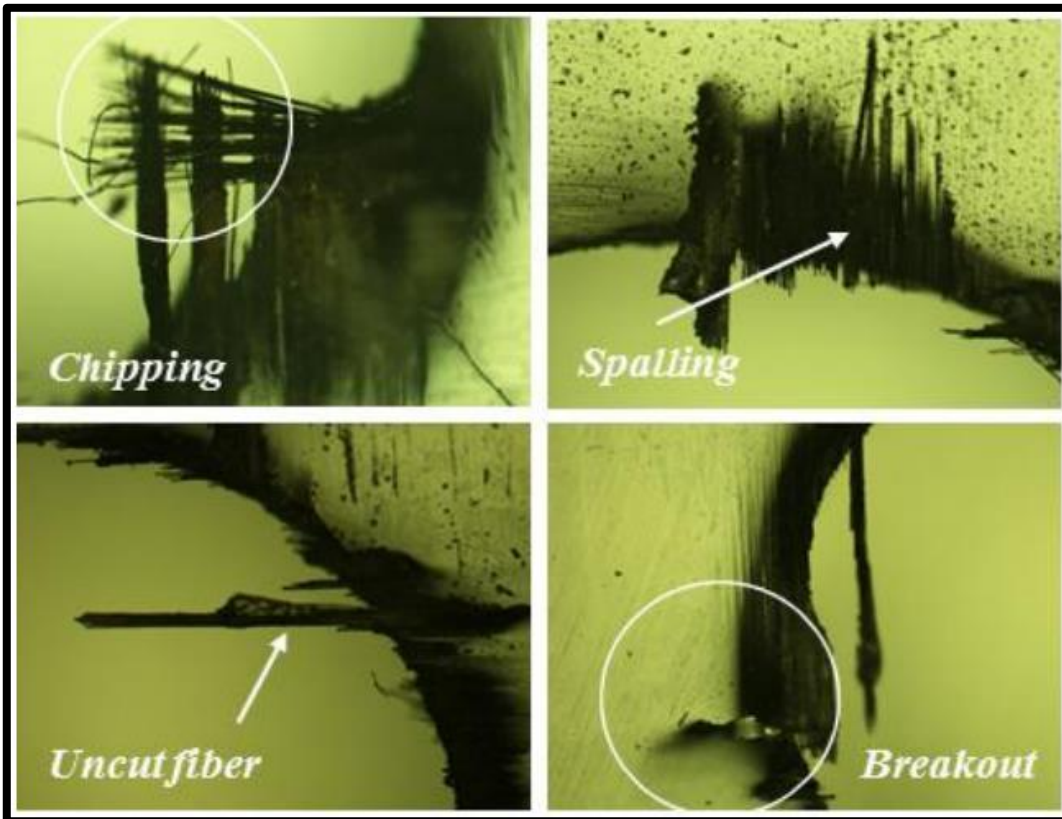
- 경량·고강도 소재인 탄소섬유복합재의 부품 요구 증대
- 우수한 기계적 물성 때문에 기계가공이 어려움
- 기계가공시 발생하는 오차를 검출하는 시스템 필요



# 연구 동기 및 관련연구

## 연구 동기

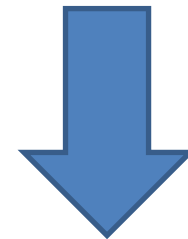
### Motivation



### 가공 성능에 치명적인 악영향

CFRP 드릴링 주요 가공결함

- chipping
- spalling
- uncut fiber
- breakout



후처리 가공을  
위한 검출 장비 필요!!

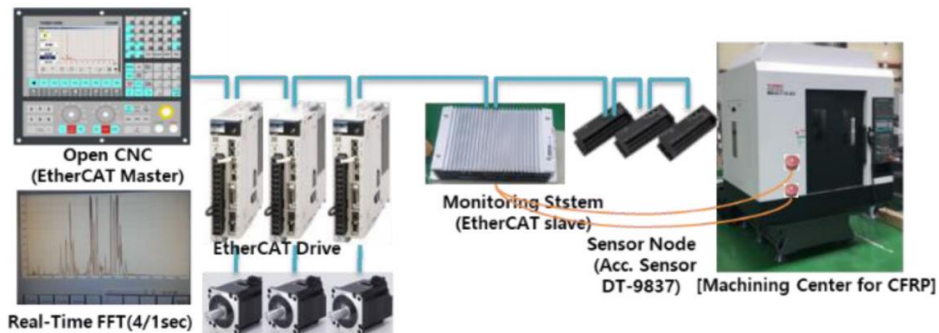
### Reference

Ki Moon Park : A Study on the Removal of CFRP Machining Defects by Various Tool Geometries(2017)

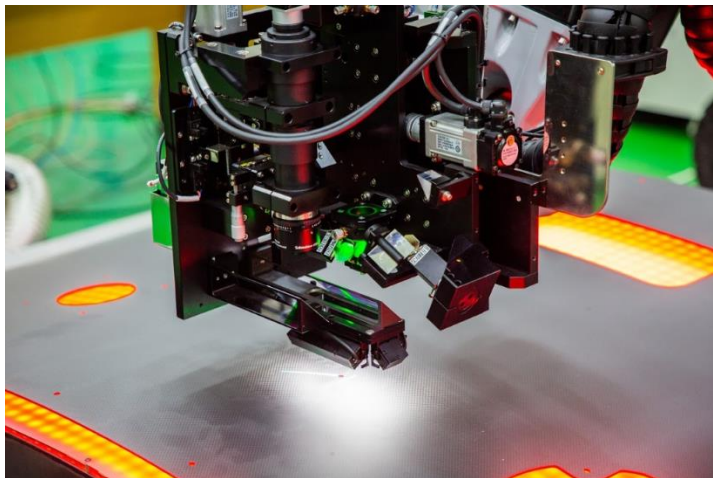
# 연구 동기 및 관련연구

## 관련 연구

### Related Research



<센서를 활용한 결함 분석 장비>



<광학검사 장비를 활용한 결함 분석 장비>

- 실시간으로 검사하기 때문에 특정한 신호에서 결함이 발생했는지 검사 가능
  - 복잡한 공정 과정이 추가됨
  - 전문가가 아닌 이상 다루기 힘들
- 
- 구멍 한개당 1초의 시간 소요로 빠른 시간안에 다양한 검사를 수행할 수 있음
  - 고가의 부품이 필요하여 중소기업에서 사용하기에 부적합

### Reference

In Hugh Choi : Open Controller Technology for the Process Monitoring of CFRP Machining(2019)

# 종합설계 개요

## 개발 목표 및 연구 개발 효과

### Objective

#### 연구 개발 목표

- 가공 이미지로 가공 결함 발생 여부와 결함의 종류, 결함 위치를 판별하는 딥러닝 모델 구현(CNN 알고리즘 적용)
- 실제 데이터를 사용하여 가공 결함 분석 실험을 통하여 95% 이상의 검출률을 가진 알고리즘 구현

#### 연구 개발 효과

- 딥러닝을 적용한 가공결함 분석을 통해 컴퓨터가 직접 판단하는 시스템을 구축
- 비용, 시간의 절약

# 시스템 수행 시나리오

CFRP 가공 결함 분석을 위한 공정 흐름도

Scenario

Reprocessing

Image Shooting



Image Processing



Server

Algorithm



Storage

Defect Result

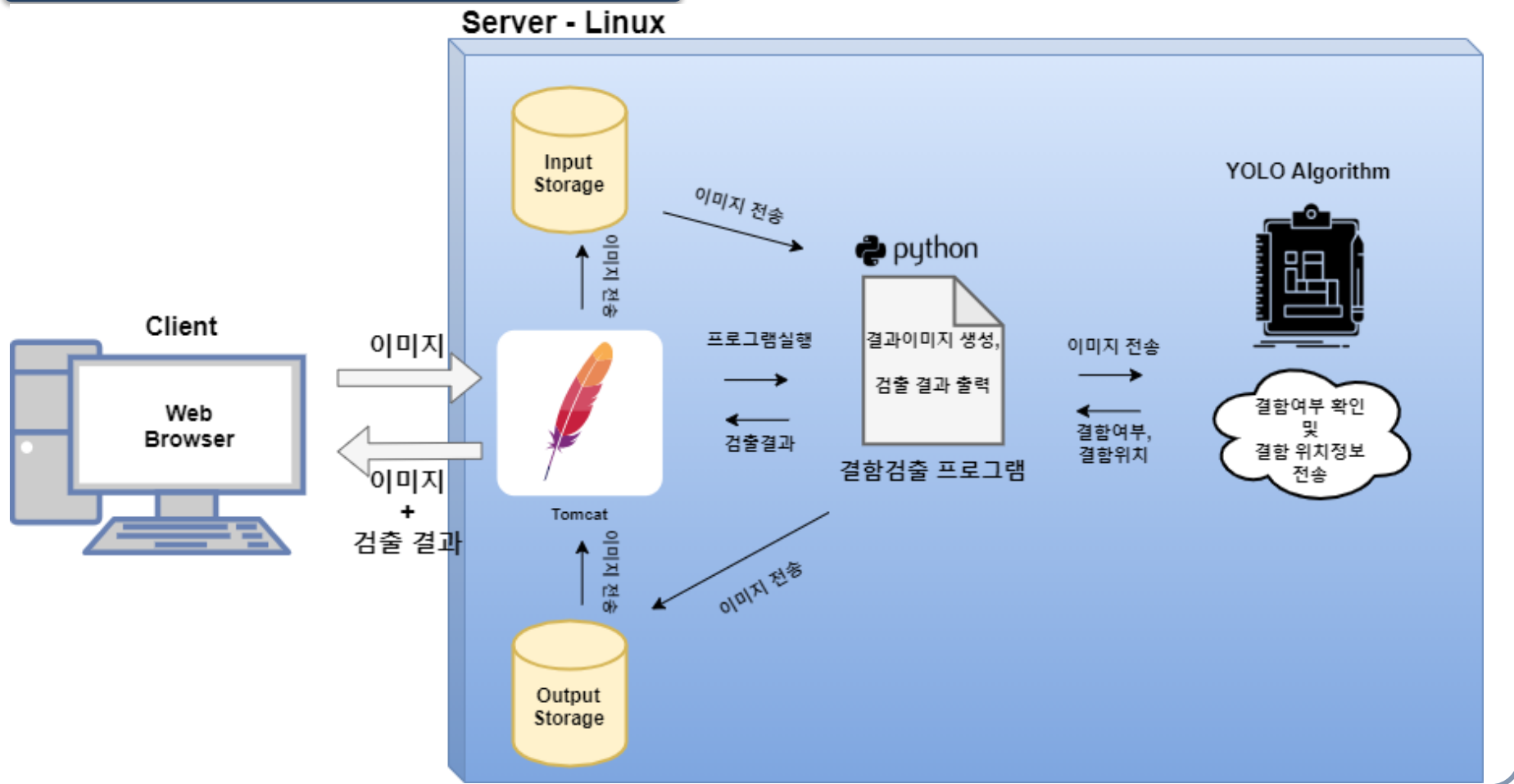




# 시스템 구성도

## CFRP 가공 결함 분석을 위한 시스템 구성도

Diagram



# 개발 환경

## 딥러닝 알고리즘을 위한 개발 환경

### Development Environment

서버 : Deep Learning AMI (Amazon Linux 2) Version 38.0 - p2.xlarge

운영체제 : Linux

웹 어플리케이션 서버 : WAS: Tomcat - 9.0.41

클라이언트 운영체제: Windows 10

사용 도구: Chrome 81.0

딥러닝 알고리즘 개발 환경

개발 도구: Pycharm community edition - 2020.1.3

사용언어: Python

웹 서버 개발 환경 개발 도구: Spring Tool Suite 4 - 4.9.0

사용 언어: Java

# 개발 방법

## 딥러닝 알고리즘을 위한 개발 방법

### Development Method

#### 1. 리눅스 톰캣 서버

- 서버 측에서 사용자에게 보여줄 JSP 페이지 구현
- 사용자로부터 전송 받은 새로운 이미지를 스토리지에 저장
- 결함 검출 프로그램으로부터 전송 받은 검출 완료된 이미지를 사용자 뷰 페이지에 표시
- 결함 검출 프로그램 실행 시 스토리지에 저장된 이미지를 가져온다.

#### 2. 결함 검출 프로그램

- 톰캣 서버로부터 전송 받은 이미지를 YOLO 알고리즘을 통해 결함 여부를 확인하고 그 결과를 생성하고 출력한다.
- YOLO 알고리즘을 통해 결함 여부와 위치에 대해 전송 받는다.
- 생성된 결과를 톰캣 서버에 전송하여 톰캣의 JSP 페이지의 사용자 뷰에 표시한다.

# 업무 분담

	이형석	최성훈	황규빈	조현민
요구사항 정의 및 분석				
시스템 설계	<ul style="list-style-type: none"> <li>❖구체적인 이미지 전처리 수행 절차 및 방식 결정</li> <li>❖기존 비전 알고리즘 장단점 분석</li> <li>❖팀에서 구현할 알고리즘 결정</li> <li>❖개발 환경 구축 (1. AWS 서버 2. Pycharm)</li> </ul>			
구현	- 서버와 통신 ❖String을 기반으로, 결함 여부를 확인 받고자 하는 이미지를 주고 결과를 돌려 받는 통신 기능 구현	- 알고리즘 모델 구현 ❖입력된 이미지를 기반으로 객체 인식 알고리즘 구현 ❖객체 인식 결과 별 라벨링을 추가하여 결과 사진으로 반환 구현	- 알고리즘 모델 구현 ❖입력된 이미지를 기반으로 객체 인식 알고리즘 구현 ❖객체 인식 결과 별 라벨링을 추가하여 결과 사진으로 반환 구현	- 이미지 전처리 동작 구현 ❖이미지 입력 후 데이터 가공 부분 구현
테스트	❖1. 테스트 데이터를 이용한 예측 결과와 실제 결과를 비교하여, 신뢰도 판단			

# 종합설계 수행일정

항목	추진사항	12월	01월	02월	03월	04월	05월	06월	07월	08월
사전 조사 및 요구사항 분석	📁 사전조사 및 요구사항 분석	—	—	—						
시스템 설계 및 상세 설계	📁 구체적인 이미지 전처리 수행 절차 및 방식 결정 📁 기존 비전 알고리즘 장단점 분석 📁 팀에서 구현할 알고리즘 결정 📁 개발환경 구축(1.AWS, 2.Pycham)		—	—	—					
데이터 수집 및 딥러닝 구현	📁 Training/Test를 위한 DataSet 수집 📁 입력된 이미지를 기반으로 객체 인식 알고리즘 구현 📁 객체인식 결과 각각에 라벨링을 추가하여 결과사진으로 반환			—	—	—	—	—		
시험 및 데모	📁 테스트 데이터를 이용한 예측 결과와 실제 결과를 비교하여 신뢰도 판단 📁 가공 시 촬영된 사진을 통해 결함 여부가 제대로 나오는지 확인							—	—	—
문서화									—	—



# GitHub

- <https://github.com/HwangGyuBin/Graduation-Work>

The screenshot shows the GitHub interface for the repository 'HwangGyuBin / Graduation-Work'. At the top, there's a dark navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the repository name is displayed with a 'Watch' button. A secondary navigation bar contains links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects' (with a count of 1), 'Wiki', 'Security', and 'Insights'. The 'Code' tab is selected. Below the navigation, there's a section for branches and tags, showing 'master' as the current branch, '1 branch', and '0 tags'. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button with a download icon. The main content area shows a commit history table with three entries, all by user 'HwangGyuBin' with the message 'MAKE FILA CHART'.

Commit Hash	Author	Message	Time
a874b3e	HwangGyuBin	MAKE FILA CHART	on 5 Oct 1 commit
	HwangGyuBin	MAKE FILA CHART	3 months ago
	HwangGyuBin	MAKE FILA CHART	3 months ago
	HwangGyuBin	MAKE FILA CHART	3 months ago

# 필요기술 및 참고 문헌

<https://www.kitech.re.kr/main/> 한국생산기술연구원 연  
도별 성과 – CFRP 가공 기술 개발

한국생산기술연구원 보도자료

<https://aws.amazon.com/> 딥 러닝 서버 플랫폼 비용과  
사양 측정

<https://keras.io/ko/> 파이썬 딥 러닝 라이브러리 CV 이  
미지 처리 알고리즘 기술

[www.springsource.org](http://www.springsource.org) Spring 프레임워크를 통한 서버  
통신 기술