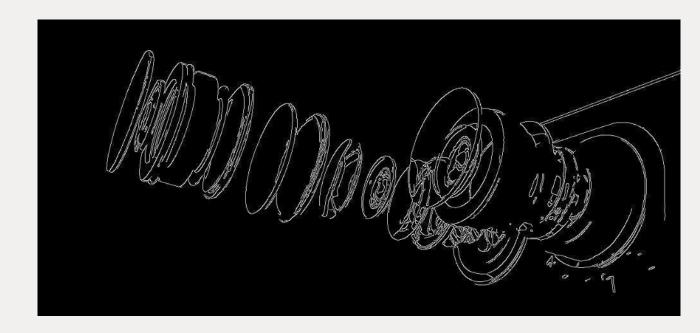
# LG Innotek Project

OpenCV와 Deep learning을 활용한 Overkill 모듈 검출





# BHAWKEYE

프로젝트 배경



### Part 1 프로젝트 개요

### 배경

- Epoxy 도포 불량 검출 공정에서 **Overkill**이 발생 (overkill: 양품을 불량으로 판정)
- Epoxy 부분에 **조명**이 반사되는 것이 주요 원인



### 목적

- Overkill 모듈들을 양품으로 판정하는 프로그램 개발
- 비용 절감 및 공정 수율 향상

### Part 1 구축 범위



소스 데이터

기업 제공 모듈 이미지 데이터



구축 범위

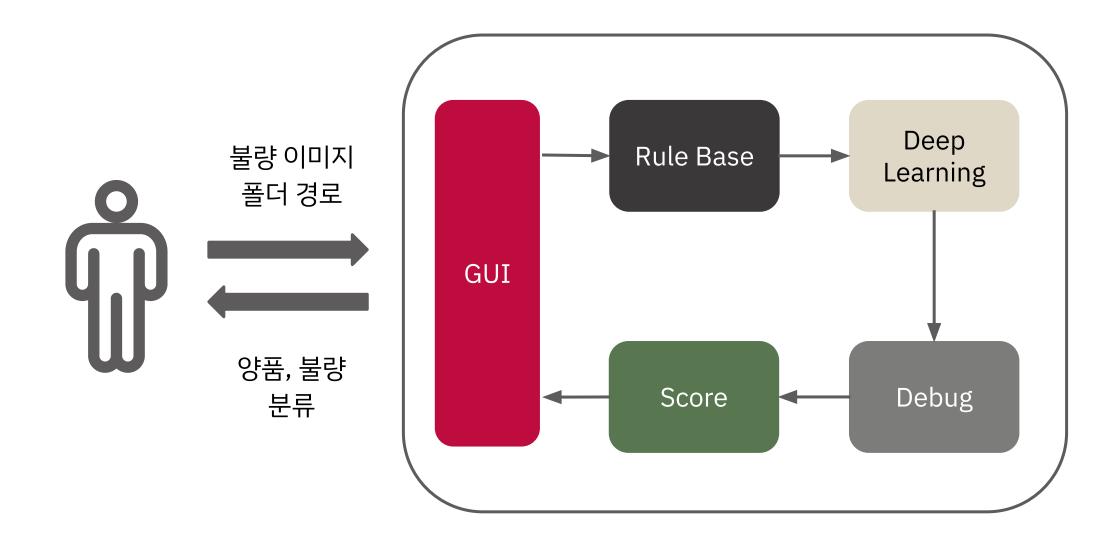
Overkill된 모듈 분류하는
Python 프로그램 release



기대 효과

공정의 불량률을 낮추고 이에 따른 **비용 감소** 예상

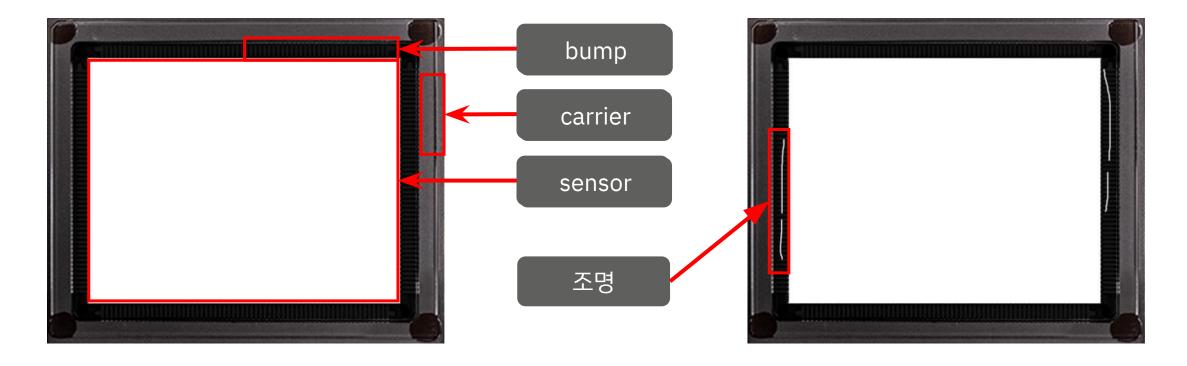
# Part 1 프로젝트 구성도



# Part 1 이미지 예시

양품 모듈

Overkill 모듈

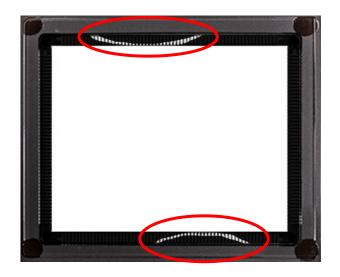


이미지 출처: https://www.jm-vistec.com/product/2u250mrxs\_250mp-cmos-sensor/

# Part 1 불량 조건

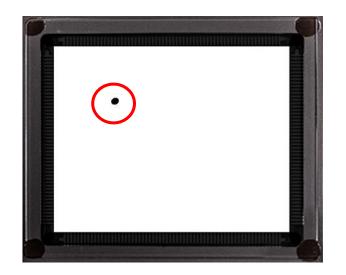
조건 1번

epoxy 도포 불량으로 인해 드러난 bump가 sensor와 **맞닿음** 



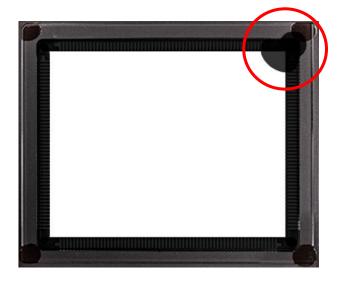
조건 2번

sensor 내부 **이물질** 발견



조건 3번

epoxy 과다 도포로 sensor, carrier에 침범



Part 1 우선순위

1

불량 이미지를 불량으로 판별

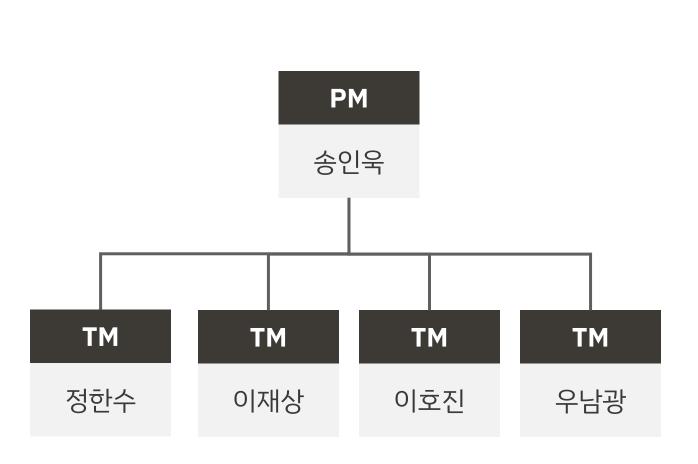
Overkill 이미지를 **양품**으로 판별

 Part 2

 프로젝트 팀 구성



### Part 2 모 역할



담당자	역할
송인욱	프로젝트 총괄 관리 Deep learning 모델
정한수	조건 3번 GUI
이재상	조건 1번
이호진	조건 2번
우남광	조건 2번

# Part 2 회의 사진





# Part 2 멘토링





### Part 2 간트 차트

CARTT			10월 2022 프로젝트 시작 11월 2022 개인 프로젝트 방표					1.2워 2022 프로젝트 전영현황 공유 및 가이드트 전영현황 공유 및 최충 보			
GANTT	<u> </u>		<u></u> 주 44	<u>로젝트 시설</u> 주 45	주 46	<u>개인</u> 주 47	프로젝트 말표 프로젝트 주 48	전염면왕 공유 및 가미드(1트 전염 주 49	연왕 공유 및 가이트및 등합 프로 주 50	역트 신영면왕 공유 및 [최종 도 주 51	
이름		료일 자원	22. 10. 23.	22. 10. 30.	22. 11. 6.	22. 11. 13.	22. 11. 20.	22. 11. 27.	22. 12. 4.	22. 12. 11.	
프로젝트 시작	22. 10. 27.	22. 10. 27. 이운기,전체	프로젝트 시작 🔷	_							
Opency 기초 교육	22. 10. 27.	22. 10. 27. 이운기,전체	Opency 기초 교육	<u> </u>							
Dpencv image processing 교육 - 1	22. 11. 3.	22. 11. 3. 이운기,전체	Opency	image processing 교육 - 1	7						
Dpencv image processing 교육 - 2	22. 11. 11.	22. 11. 11. 이운기,전체		Ope	encv image processing 교육 - 2						
개인 프로젝트 - log 파일	22. 10. 28.	22. 11. 3. 전체	개인 프로젝트 - log 파일		1						
개인 프로젝트 - opencv 구현	22. 11. 4.	22. 11. 16. 전체		개인 프로젝트 - opency 구현							
개인 프로젝트 발표	22. 11. 17.	22. 11. 17. 이운기,전체				개인 프로젝트 발표 🝑					
Rule base 조건별 검사 방식 구상	22. 11. 4.	22. 11. 22. 전체	3	Rule base 조건별 검사 방식 구상							
ule base 조건 검사 및 개선	22. 11. 23.	22. 11. 28. 전체				Rule b	pase 조건 검사 및 개선	:			
ule base 추가 조건 및 코드 통합	22. 11. 26.	22. 12. 1. 전체					Rule base 추가 조건 및 코드 통	<b>1</b>			
NN 모델 교육	22. 11. 17.	22. 11. 17. 이운기,전체				CNN 모델 교육	կ				
NN 모델 학습 및 검사	22. 11. 18.	22. 11. 25. 송인욱,정한수				CNN 모델 학습 및 검사		7			
도로젝트 진행현황 공유 및 가이드	22. 11. 23.	22. 11. 23. 이운기,전체				프로젝트 진	행현황 공유 및 가이드 👆				
한간 보고서 발표	22. 11. 25.	22. 11. 25. 송인욱					중간 보고서 발표 🬗	•			
NN 모델 개선 및 통합	22. 11. 26.	22. 12. 1. 송인욱,정한수					CNN 모델 개선 및 통	합			
로젝트 진행현황 공유 및 가이드	22. 11. 29.	22. 11. 29. 이운기,전체					프로젝트 진행현황	공유 및 가이드 🕹			
로젝트 중간 정리 및 통합	22. 12. 2.	22. 12. 2. 이운기,전체						프로젝트 중간 정리 및 통합			
ython program release 교육	22. 12. 2.	22. 12. 2. 이운기,전체						Python program release 교육	<u> </u>		
UI 프로그래밍 및 릴리즈	22. 11. 28.	22. 12. 7. 전체					GUI 프로그래딩	빙 및 릴리즈			
종 보고서 작성 및 발표 준비	22. 12. 2.	22. 12. 13. 송인욱						최종 보고서 작성 및 발표 준비		-	
로젝트 진행현황 공유 및 가이드	22. 12. 8.	22. 12. 8. 이운기,전체						프로젝트	진행현황 공유 및 가이드 🔶		
종 보고서 발표	22. 12. 14.	22. 12. 14. 송인욱								최종 보고서 발표 🦫	

### Part 2 일정

일	월	화	수	목	금	토	일	월	화	수	목	급	토
				10 / 27							11 / 03	11 / 04	
												l J	
										i de			
							개인 프로젝	트 (log 파일	)				
				멘토링							멘토링		
					11 / 11					11 / 16	11 / 17	11 / 18	
	Į	Į.			D.1		H 7J 11 HL 1	7.11	1		ļ	l J	
	I		I		Rule	e base 조건	열심사망식	T'8	Ī	I	I	1	
				ᆌᅁᄑ	l 로젝트 (open	[ ] [전 구성)			1				
		S		/11년 ==	멘토링	(   L)	ĺ	*		Ĭ	멘토링		
·	i i	11 / 22	11 / 00		100000000000000000000000000000000000000	11 / 20		44 / 20	11 /00		댄포이	10 / 00	,
		11 / 22	11 / 23		11 / 25	11 / 26		11 / 28	11 /29			12 / 02	
3	Rule base 조건 검사 및 개선									조기 비-			
	CNN 모델 학습 및 검사					통합 파일 작성					중간 발표		
						GUI 프로그래밍 및 릴리스							
			멘토링						멘토링			멘토링	
			12 / 07	12 / 08					12 /13				
									최종 발표				
	I.												
			멘토링										

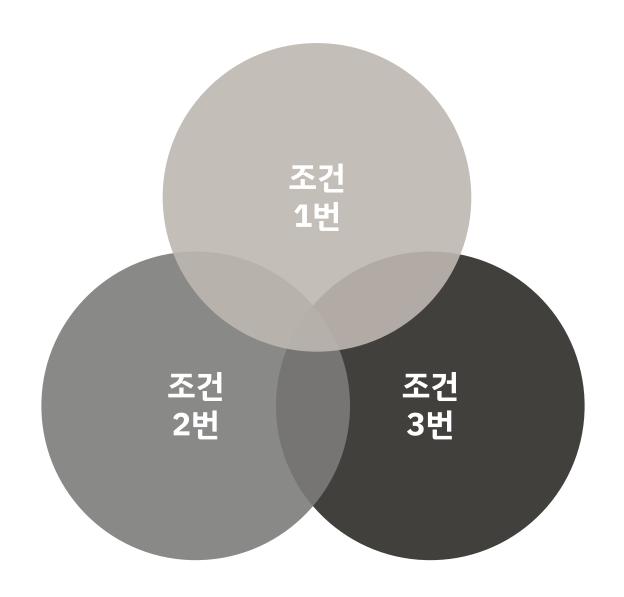
프로젝트 진행



Part 3 Rule base

### 불량 이미지 검출 프로세스

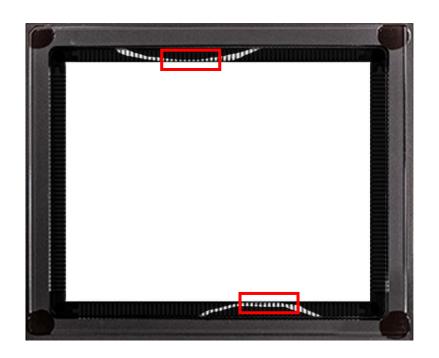
OpenCV를 활용한 image processing



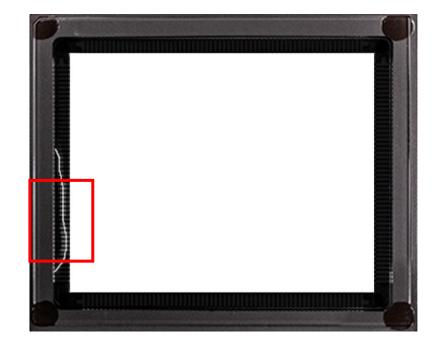
### 조건 1번 - 아이디어

불량 모듈에서 발견되는 패턴을 통해서 검출 시도

불량 모듈은 드러난 범퍼가 센서와 맞닿아 있음



Overkill 모듈은 조명 반사로 범퍼가 보여도 센서와 닿지 않음

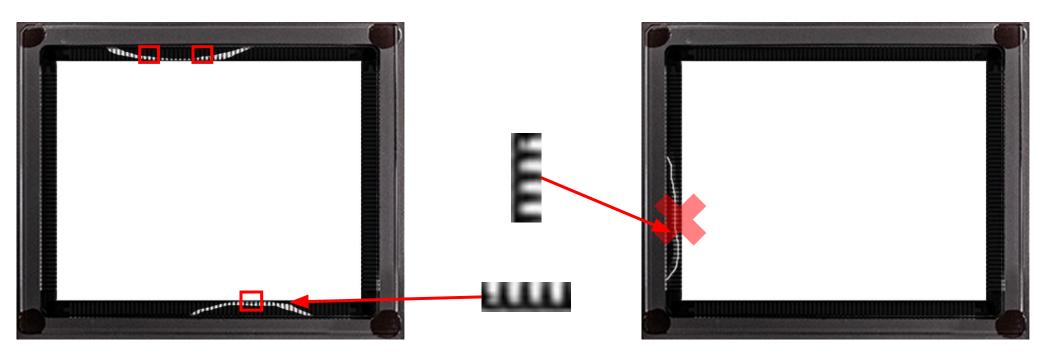


# Part 3 조건 1번 - 검사 방법

불량 모듈

범퍼와 센서가 닿는 부분을 **탬플릿 매칭**으로 찾아냄 Overkill 모듈

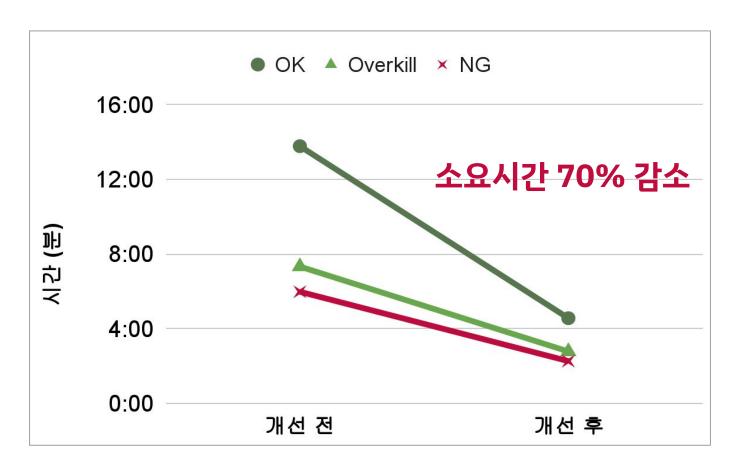
Overkill 모듈은 매칭되지 않으며 매칭되더라도 낮은 정확도를 보임



### Part 3 조건 1번 - 속도 개선

### 템플릿 매칭 방식 변경

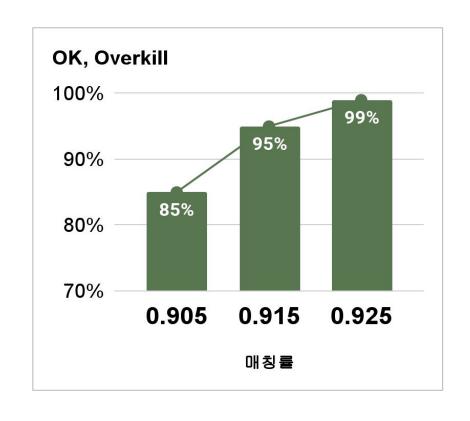
전체 이미지 검사 —— bump의 상하 좌우만 잘라서 검사

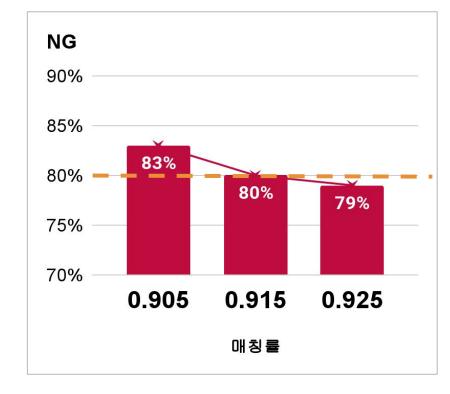


### Part 3 조건 1번 - 성능 개선

### 템플릿 매칭률 최적화

템플릿 매칭 시 제품과 템플릿 이미지 간의 **정확도** 설정값 변경



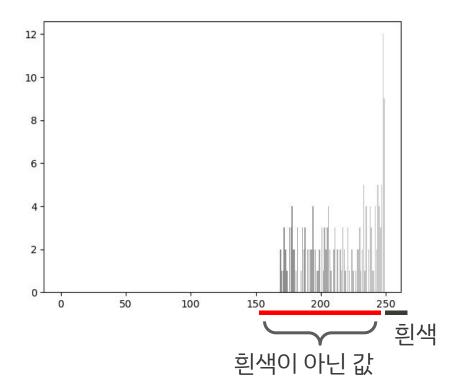


조건 1번 불량 비율 (80%)

### 조건 2번 - 아이디어

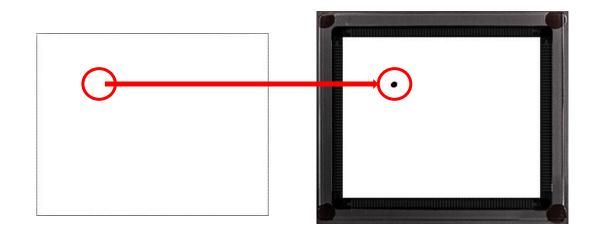
#### 방법 1

센서 내부에 히스토그램으로 검은색 **픽셀의 개수** 구해서 이물질 검출



#### 방법 2

정상 센서와 비교하여 다른 부분 색칠 후 **히스토그램**으로 이물질 검출



### 조건 2번 - 검사 방법

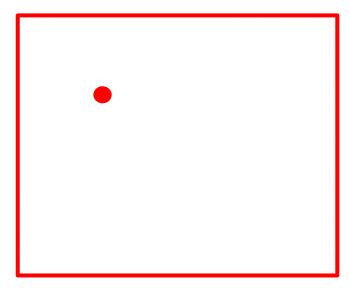
방법 1

흰색 센서를 찾아내면 안쪽의 **검은색 픽셀**의 개수를 구할 수 있음

•

방법 2

매칭시킨 후 센서를 잘라내면 정상 센서와 **안 맞는 픽셀** 개수를 구할 수 있음



### Part 3 방법1 - 성능 개선

기존 방식

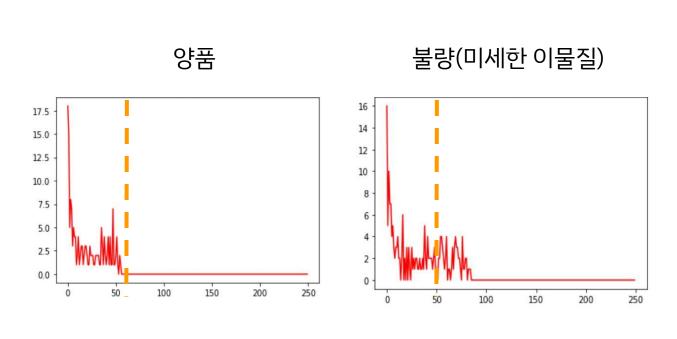
개선된 방식

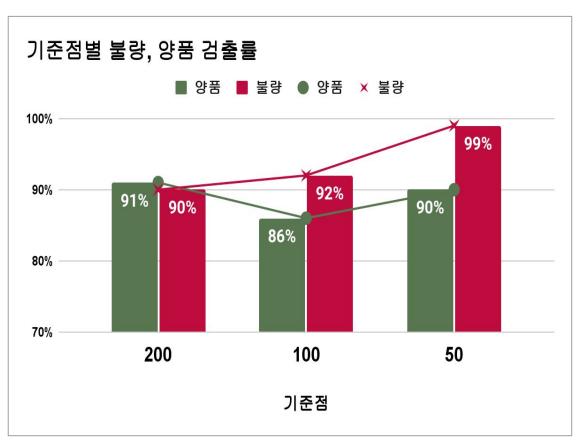
#### Overkill 검출률



### Part 3 방법2 - 성능 개선

### 히스토그램 색상별 빈도 수 기준점 조절

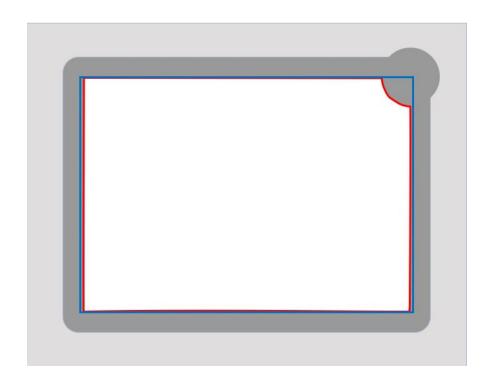




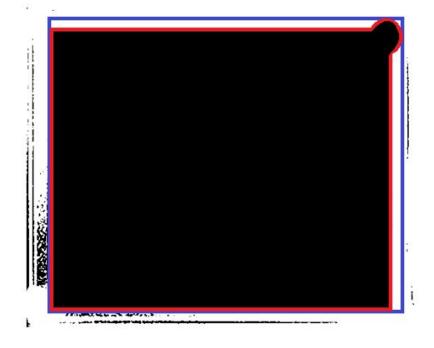
### 조건 3번 - 아이디어

### 컨투어와 사각형의 넓이 차이 비교

센서 컨투어와 내부 사각형의 넓이 확인

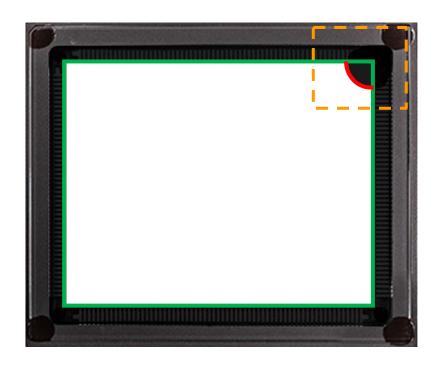


epoxy 영역 컨투어와 외부 사각형 넓이 확인 (otsu threshold 사용 대비 조절)

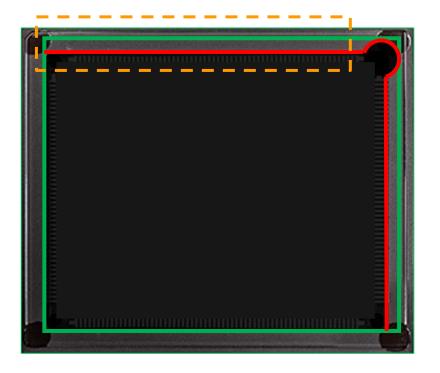


### 조건 3번 - 검사 방법

내부 침범시 컨투어의 넓이가 내부 사각형의 넓이보다 **작음** 



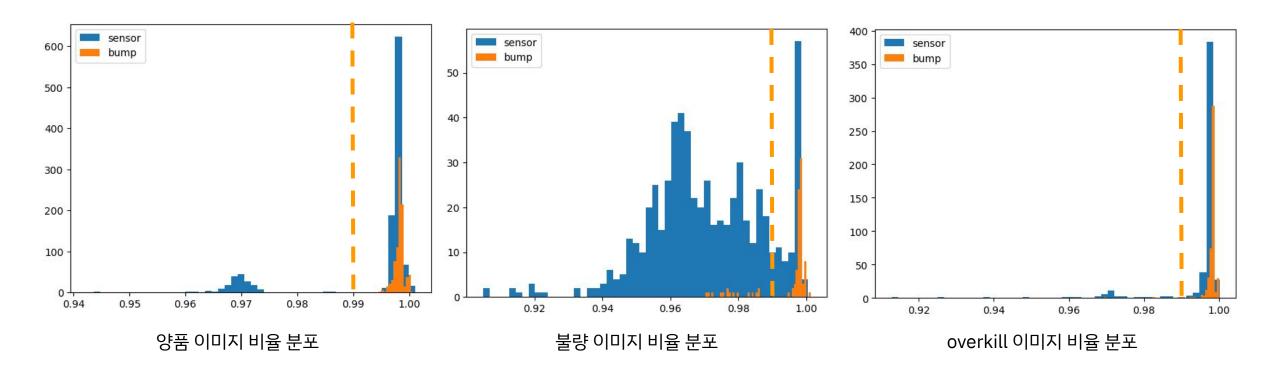
외부 침범시 컨투어의 넓이가 외부 사각형의 넓이보다 **작음** 



### Part 3 조건 3번 - 성능 개선

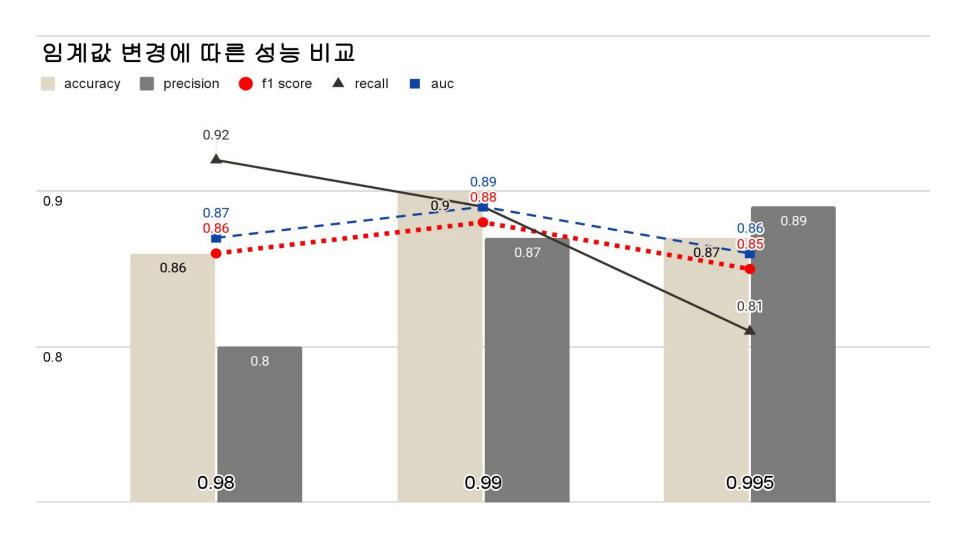
크기 비율 임계값 조절

실제 양품, 불량, overkill 이미지의 **비율 분포** 확인



Part 3 전 3번 - 성능 개선

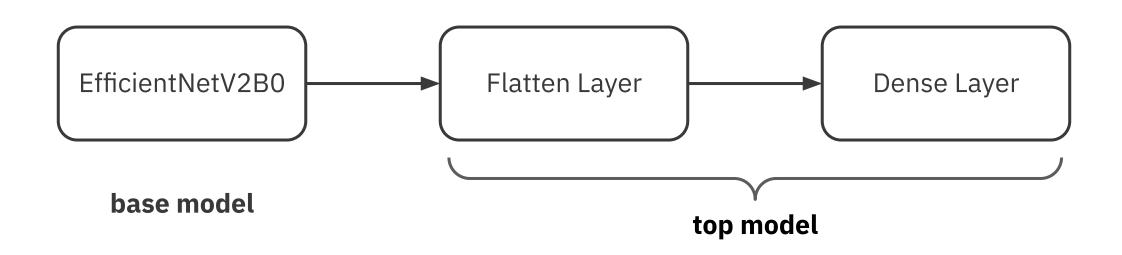
### 크기 비율 임계값 조절



### Deep learning – 모델 구성

EffientNet V2 를 활용한 **Transfer Learning** 

**Efficient Net Fine Tuning** 

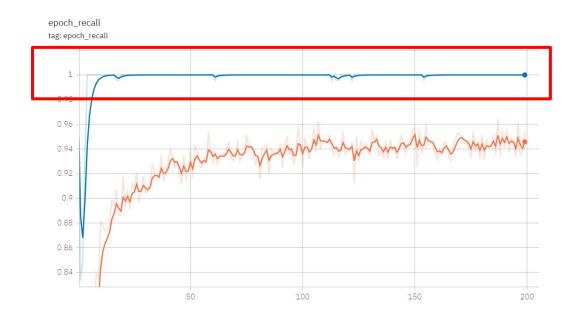


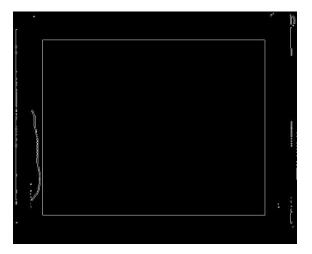
# Part 3 Deep learning - 전처리

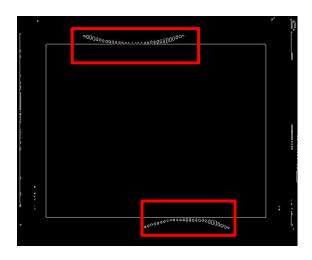
resize 과정에서 **과적합** 발생 ──── resize 후 **Canny** 사용하여 특징 강조

train recall: 0.94

val recall: 1







# Part 3 Deep learning - 모델 학습

top 모델 학습 ───── base 모델 fine tuning

train acc: 0.88

val acc: 0.91

epoch\_accuracy
tag: epoch\_accuracy

0.92

0.9

0.88

0.86

0.84

0.82

0.8

0.78

0.76

0.74

0.72

50

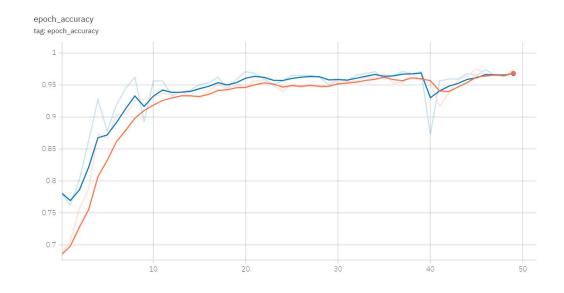
100

150

200

train acc: 0.96

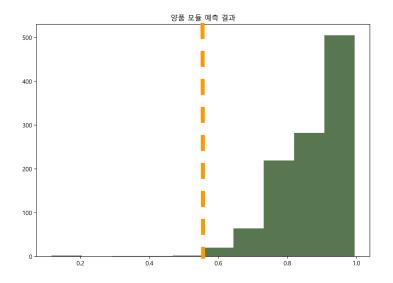
val acc: 0.96

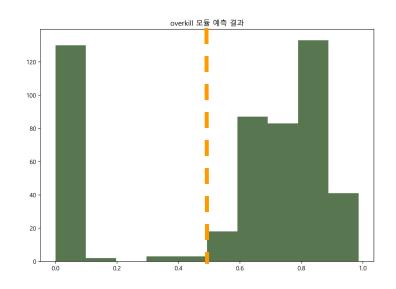


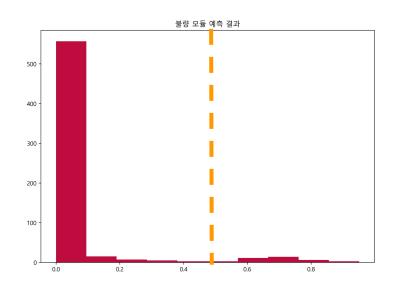
### Deep learning – 검사 결과

양품 모듈 예측 결과 정확도 98% overkill 모듈 예측 결과 정확도 78%

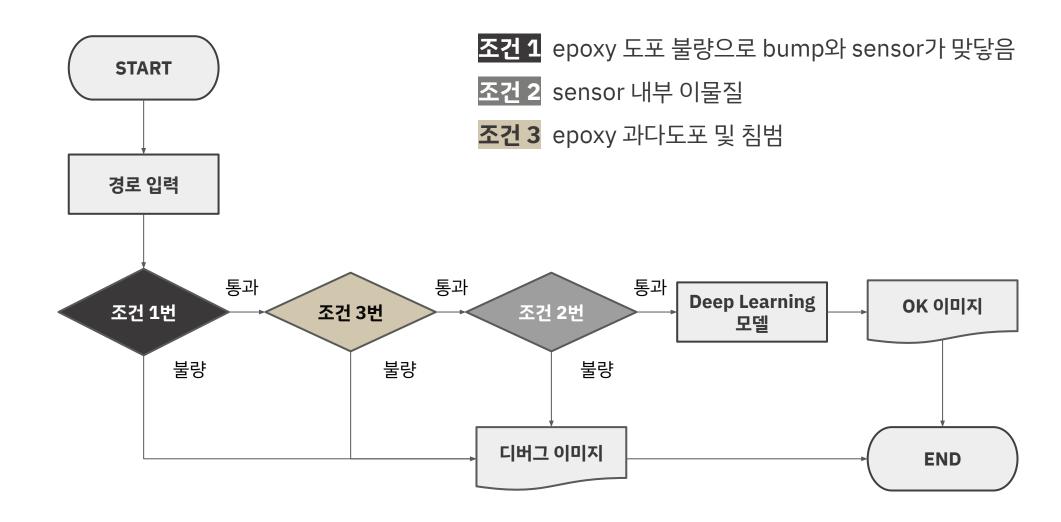
불량 모듈 예측 결과 정확도 92%







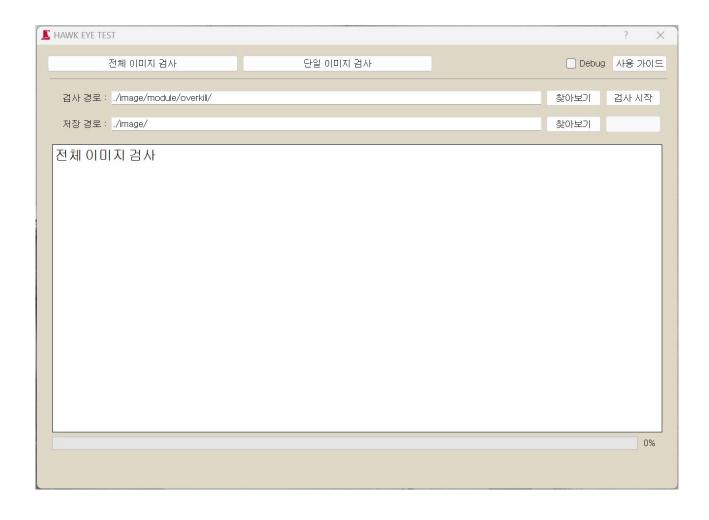
Part 3 GUI – Flow Chart



Part 3 GUI - 화면 설계서

Page Title.	전체 이미지 검사	Group Title.	Description	n.	Page.	1
			1	검사 선택 창		
		2	옵션 선택			
	1	1 2	3	검사, 저장 경로 지정		
	3	4	4	검사 시작		
	3		5	검사 로그, 결과 출력		
			6	진행바		
			7	Help 페이지		
	Ę	5				
	6					

# Part 3 GUI



### Part 3 프로그램 통합

검사 및 디버깅 Rule 조건별 검사 실행하여 불량 판정하고 불량 판정시 디버그 이미지 출력 Base Deep **GUI** Learning

### 현장에서 사용하는 UI

현장에서 사용하기 위한 인터페이스 디버그 저장 옵션 및 로그 저장

#### Overkill 이미지 검증

양품 모듈에서 모델 예측 결과를 출력해 Overkill 이미지 한번 더 검증

### Program Release

Pyinstaller 사용하여 exe 실행파일로 배포







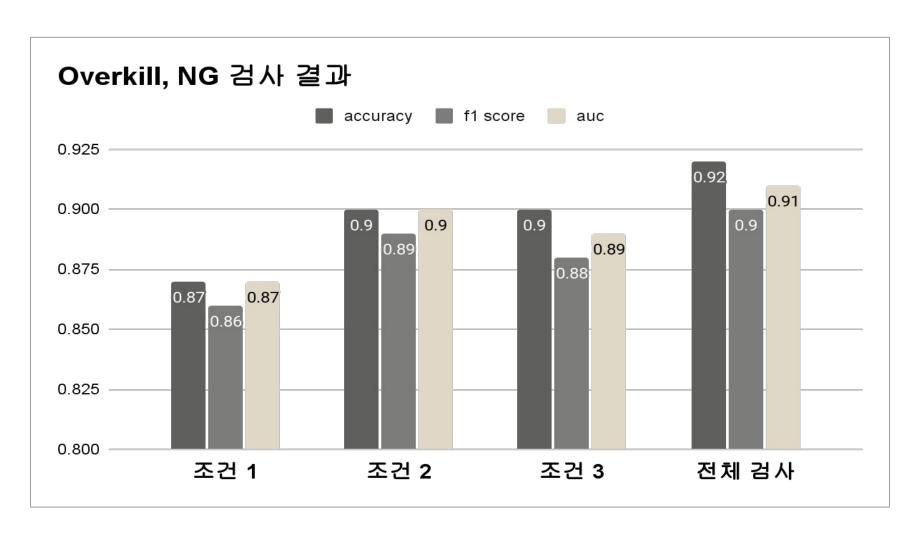
 Part 4

 프로젝트 결과

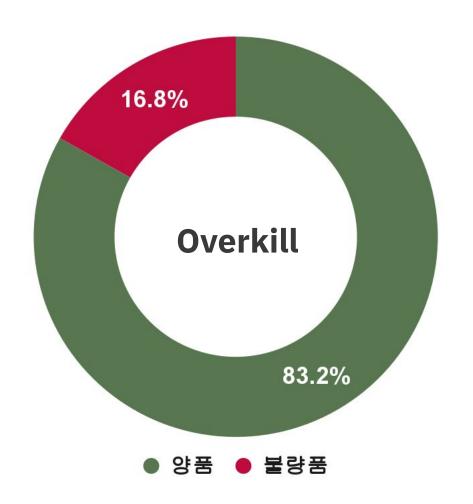


Part 4 조건별 성능

모듈 개수 = Overkill 500개, 불량 600개, 양품 1100개



Part 4 프로그램 성능

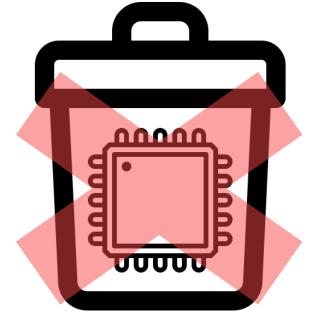




#### Part 4 프로그램 기대 효과

양품 모듈 불량처리 방지 Overkill 모듈 **83.2%** 감소

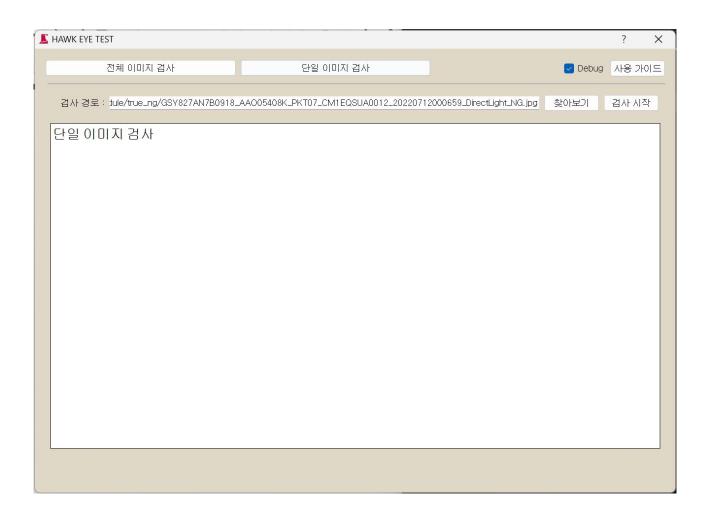




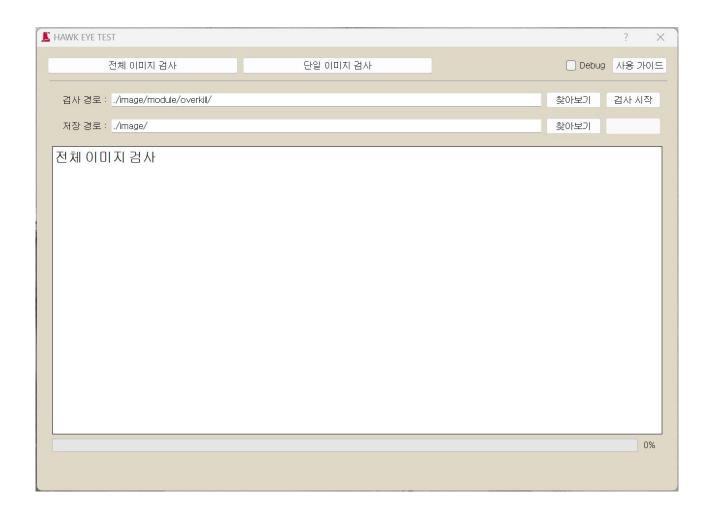




# Part 4 단일 이미지 검사



# Part 4 전체 이미지 검사



### **Thanks**

Do you have any questions?



Extra COlor

