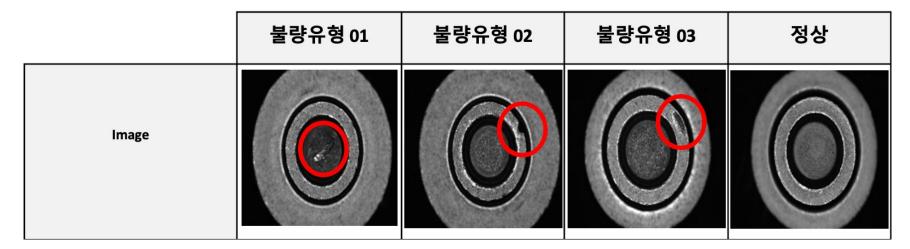
PROJECT TITLE

초정밀 배터리 단자 불량 유형 판정

<초정밀 베터리 단자 데이터>





서 론

Notations

MODEL

실험방법

실험결과

향후연구



INTRODUCTION

연구배경

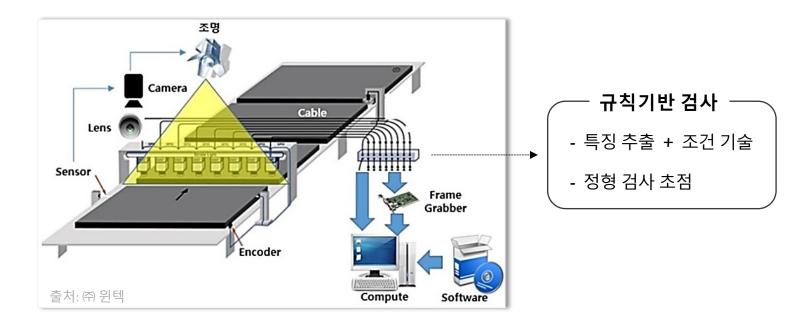


- ✓ 미래 산업의 변화는 전동화(Electrification) · 무선화(Cordless)가 핵심
- ✓ 모든 사물이 이차전지로 움직이는 시대 개막
- → 그에 따라, 베터리 시장의 규모는 점차 커지며, 베터리의 품질에 대한 관심은 더욱 커짐

연구배경

서론

- "머신 비전(machine vision)"이란?
 - → 이미지 캡처, 프로세싱, 판정하는 H/W 및 S/W 어플리케이션



✓ 초정밀 부품(10mm 이하)의 경우 육안 검사가 매우 어려움"육안 검사 대체 → 품질 및 생산성 향상에 기여"

연구개발의 필요성

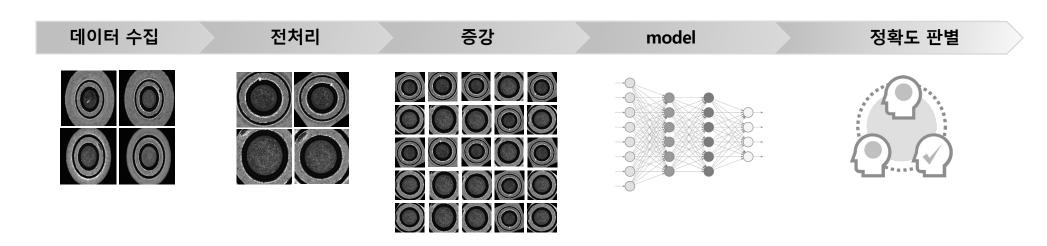
✓ 기존 머신 비전의 한계

: 규칙 기반의 비전 검사 주류를 이루었기에 복잡하거나 다양한 상황 조건에 따라 다른 판단이 요구된다.

✓ AI 머신 비전

: 형상 판정 정확도↑, 불량 유형 분류 가능, 비정형/색상 판별 가능

데이터 수집 및 과정



불량 유형 및 정상 데이터

<초정밀 베터리 단자 데이터>

	불량유형 01	불량유형 02	불량유형 03	정상
Image				
불량 내용	내부 원 영역 불 량	중간원 찍힘, 깨짐 ^{142장}	중간 원 갈림 118 장	정상 38 장
	139 장			J

3개 유형의 불량, 1개의 정상으로 양/ 불량 판정



NOTATIONS

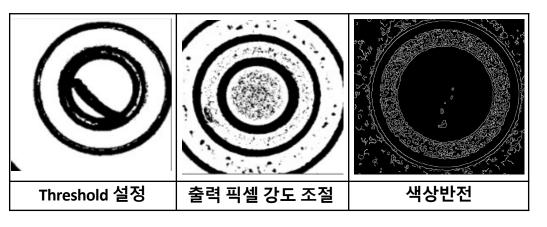
1. 데이터의 전처리(Preprocessing)

- 데이터의 결측치 확인 및 제거하고 불일치 되는 부분을 일관성 있는 데이터의 형태로 전환하는 전 과정
- 대용량 데이터에서의 이미지 분류과정에서는 데이터 전처리에 따라 성능이 크게 좌우된다.
- 보다 **정확하고 효율적인 기계학습**을 위해서는 데이터의 **전처리 과정이 요구된다.**

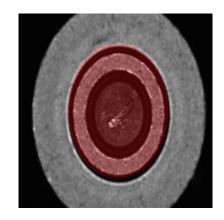


* Overfitting: 기계 학습 모델이 학습 데이터에 대한 정확한 예측을 제공하지만 새 데이터에 대해서는 제공하지 않을 때 발생하는 바람직하지 않은 기계 학습 동작

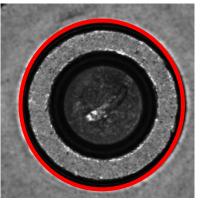
1. 데이터의 전처리(Preprocessing)



- ✓ 모든 class에 **일관된 적용 불가**
- ✓ 여러 이미지 전처리 기법을 사용해보았지만 Crop이 가장 적합



<전처리 前>



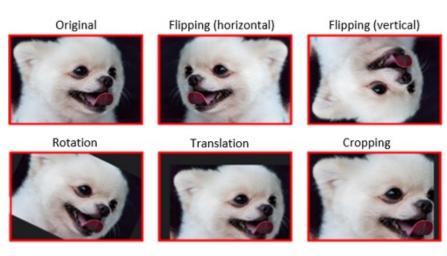
<전처리 後>

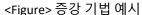
- ✓ 나사의 불량 부위를 담고 있는 부분만을 학습시키고자 함
- ✓ 유의미한 특징을 담고 있는 부분만을 Crop하여 전처리 진행

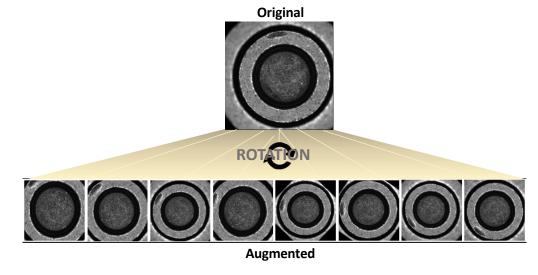
2. 데이터의 증강(Augmentation)

서론

- ✓ 갖고 있는 데이터셋을 여러 가지 방법으로 Augment하여 실질적인 학습 데이터셋의 규모를 키울 수 있는 방법
- ✓ 머신/딥 러닝을 수행하기 위해서는 기본적으로 훈련을 위한 데이터의 양이 매우 많이 필요
- ✓ 훈련용 데이터의 양이 적으면 Overfitting 문제 발생
- ✓ Machine Learning을 위한 패턴인식을 위해서는 학습 데이터의 양이 많을수록 그 성능이 향상됨



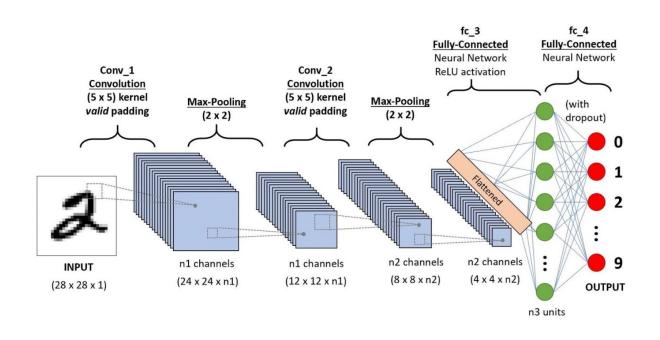




<Figure> 증강된 베터리 단자 데이터



알고리즘 선정 과정



국내외 학회 논문들을 찾아보면, image feature extract/object-detection에 CNN 알고리즘을 활용하여 좋은 성과를 냈다는 연구결과를 발견하였습니다.

MLP(완전연결신경망) /. CNN(합성곱 신경망) 차이: **Feature 추출 방식**

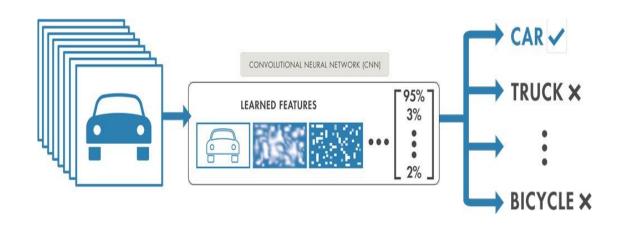
MLP는 이미지 행렬을 **1차원 배열**로 만들고 신경망에 입력으로 넣어 가중치를 계산해서 **정보 손실이 큰** 반면,

CNN은 이미지 행렬을 **2차원 배열**에서 가로, 세로 축 전부 합성곱 연산과 풀링 연산으로 특징점을 추출하기에 정보 손실이 적습니다. (* **풀링으로 추출한 특징점에 대한 가중치만을 연산)**

3. 불량 판정 모델

✓ CNN

이미지를 분석하기 위해 패턴을 찾는데 유용한 알고리즘으로 데이터에서 이미지를 직접 학습하고 패턴을 사용해 이미지를 분류한다.

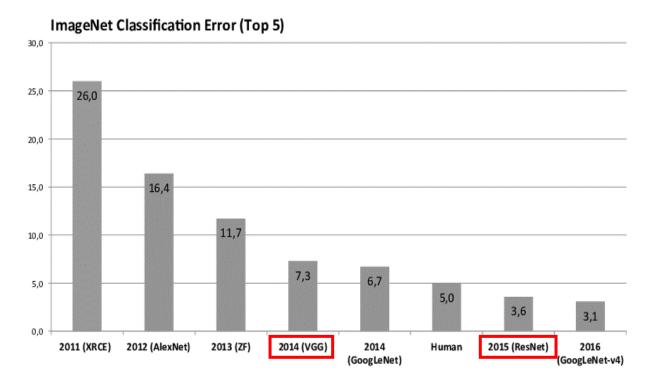


CNN이 자동으로 특징을 학습하여 객체를 분류

✓ CNN 알고리즘 중에서 VGG16, ResNet50, ResNet101 선택

3. 불량 판정 모델

✓ ImageNet Challenge

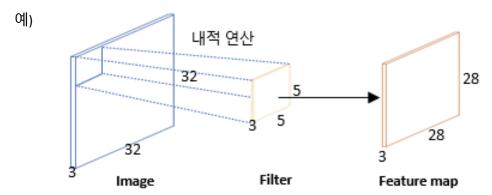


✓ VGG와 ResNet이라는 기본구조를 가져와 이미지 데이터에 적합하게 모델링하여 실험을 진행했다.

3. 불량 판정 모델

✓ Convolution 이란?

- 이미지의 특징을 판단



✓ Pooling 이란?

- 이미지의 특징을 강화 (값들 중 특정 값들만 유지)

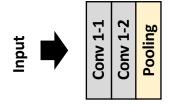
MODEL

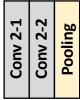
예) Max pooling

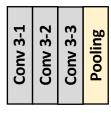
1	2	2	4			
5	6	7	8		6	8
3	2	1	0	Max pooling with 2x2 filter, stride 2	3	4
1	2	4	3	ZAZ IIICI, SUIGC Z		

불량 판정 모델

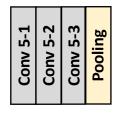
✓ Architecture of Vgg-16

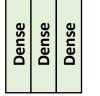










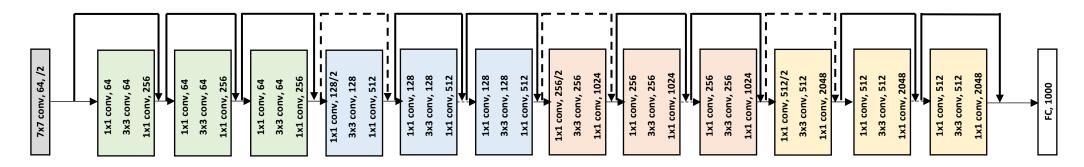




- ✓ VGG는 간단한 구조로도 깊이에 따라 높은 성능을 보일 수 있다.
- ✓ VGG는 좋은 성능과 사용하기 쉬운 구조로 GoogLeNet보다 인기 있는 모델
- ✓ 파라미터가 많기 때문에 학습이 매우 느리고 용량이 엄청나게 크다.

불량 판정 모델

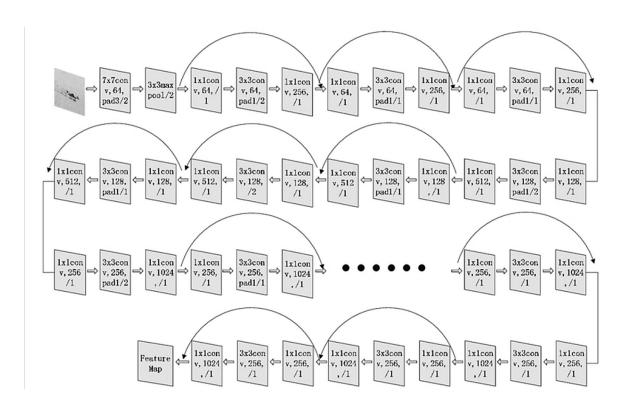
✓ Architecture of ResNet-50



- ✓ 기존의 VGG 네트워크보다 더 깊고 Residual Block을 활용해 복잡도와 성능이 더 개선되었다.
- ✓ 구현이 간단하며, 학습 난이도가 매우 낮아진다.
- ✓ 깊이가 깊어질수록 높은 정확도 향상을 보인다.
- ✓ 입력 제약이 매우 크고, 충분한 메모리(RAM)가 없으면 학습 속도가 느릴 수 있다.

불량 판정 모델

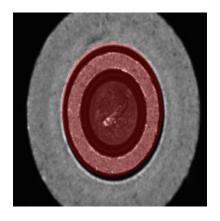
✓ Architecture of ResNet-101



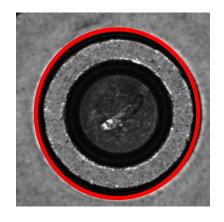


실험방법

1. 데이터의 전처리(Preprocessing)



<전처리 前>



<전처리 後>

- ✓ 나사의 불량 부위를 담고 있는 부분만을 학습시키고자 했다.
- ✔ 유의미한 특징을 담고 있는 부분만을 Crop하여 전처리 진행하였다.
- ✔ Crop으로 전처리하여 Overfitting을 줄였다.

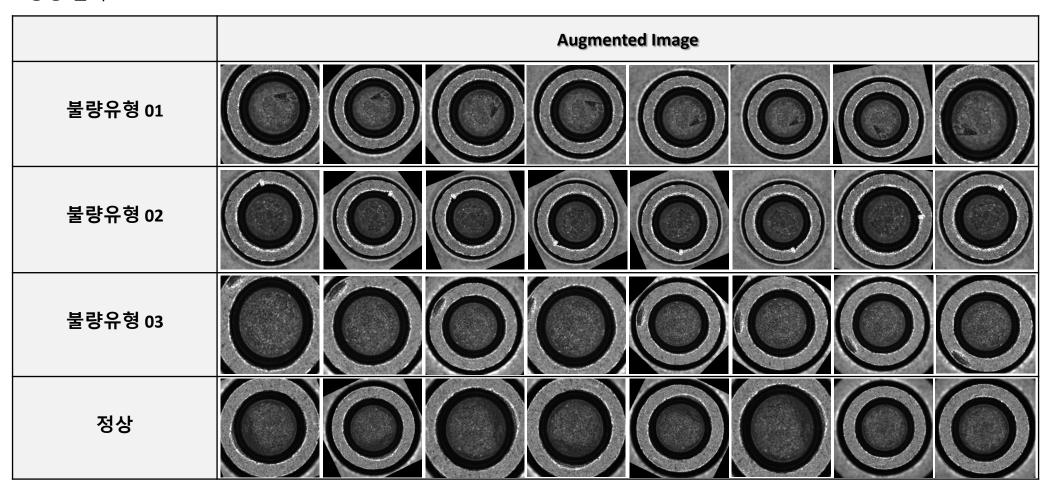
2. 데이터의 증강(Augmentation)

	불량유형 01	불량유형 02	불량유형 03	정상
Image				
불량 내용	내부 원 영역 불량 139 장	중간 원 찍힘, 깨짐 142 장	중간 원 갈림 118 장	정상 38 장

- ✓ 기존 데이터로는 머신러닝 모델에 학습시키기에 양이 매우 적음.
- ✓ 데이터 특성상 원의 형태임을 고려하여 'Rotation (회전)'만을 증강 옵션으로 설정.
- ✓ Class 별로 1,500장 씩 증강
- ✓ 여러 종류의 증강 기법을 사용
- ✓ 증강 데이터의 성능이 증강 전 데이터의 성능보다 좋음

2. 데이터의 증강(Augmentation)

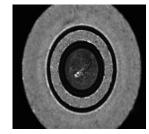
<증강 결과>



3. 데이터 수집 및 데이터셋 구성

dataset	전처리	증강		
dataset1	Crop	증강X		
dataset2	전처리X	증강X		
dataset3	전처리X	증강O		
dataset4	Crop	증강O		

- ✓ 증강은 Augmentor의 rotation 기능을 활용하여 증강함. (1,500 장/class)
- ✓ 전처리는 유의미한 특징을 담은 부분까지 crop함.



<전처리 前>

<전처리 後>

4. 실험방법

	Dataset 1			Dataset 2		Dataset 3			Dataset 4			
전처리	Crop	Crop	Crop	전처리x	전처리X	전처리x	전처리X	전처리X	전처리X	Crop	Crop	Crop
증강	증강X	증강X	증강X	증강X	증강X	증강X	증강0	증강0	증강0	증강0	증강0	증강0
모델	vgg-16	RN50	RN101	vgg-16	RN50	RN101	vgg-16	RN50	RN101	vgg-16	RN50	RN101
실험횟수		5회 반복 실험										



실험결과

실험결과

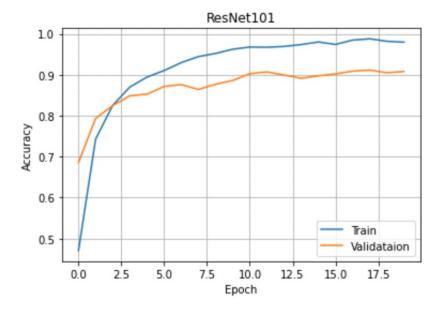
	Dataset 1			Dataset2		Dataset3			Dataset4			
전처리	Crop	Crop	Crop	전처리x	전처리x	전처리x	전처리x	전처리X	전처리x	Crop	Crop	Crop
증강	증강X	증강X	증강X	증강X	증강X	증강X	증강0	증강0	증강0	증강0	증강0	증강0
모델	vgg-16	RN50	RN101	vgg-16	RN50	RN101	vgg-16	RN50	RN101	vgg-16	RN50	RN101
정확률 (1)	92.7%	86%	93.3%	78%	84%	84%	87.3%	87.5%	89.5%	92.6%	83.9%	95.8%
정확률 (2)	86%	84%	78%	76%	78%	82%	89.8%	90%	90.5%	88%	89.7%	92.5%
정확률 (3)	74%	76%	78%	76%	78%	80%	86.7%	89.5%	88%	86%	88.3%	90.5%
정확률 (4)	80%	74%	78%	80%	74%	74%	86.3%	89.7%	88.3%	84.5%	88.8%	89.7%
정확률 (5)	78%	84%	84%	80%	82%	76%	87.5%	91%	89.8%	87.7%	89.7%	89.3%
Avg	82.14%	80.8%	82.26%	78%	79.2%	79.2%	87.52%	89.54%	89.42%	87.76%	88.08%	90.56%

실험결과 및 결과해석

<인자 조합별 정확도>

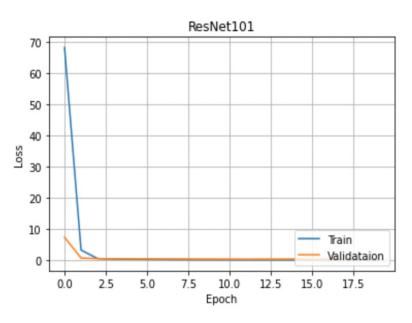


▶ 전처리, 증강 여부와 3개 모델의 정확도 중 "전처리O/증강O/Resnet101"이 가장 높은 정확도



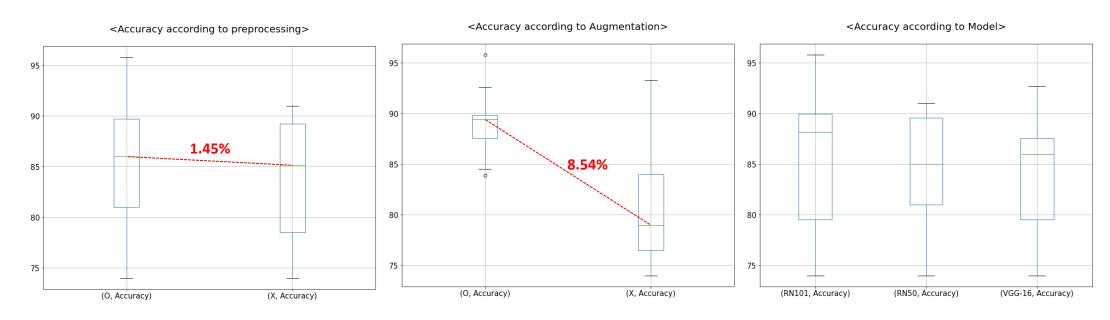
Training max: 98.77%

Validation max : 91.13%



실험결과 및 결과해석

인자 별 정확도 (Boxplot)



- ➤ 전처리 여부에 따른 정확도의 평균 차이는 약 1.45%이며, 전처리를 하지 않은 data는 편차가 크다는 것을 알 수 있었음 그에 따라, 전처리는 모델의 강건함에 영향을 미친다는 결론 도출
- ▶ 또한, 증강 여부에 따른 정확도의 평균 차이는 약 8.54%이며, 증강에 따라 정확도와 강건함의 정도가 크게 차이 남

실험결과 및 결과해석

	df	sum_sq	mean_sq	F	PR(>F)
Preprocessing	1.0	37.762667	37.762667	2.369534	0.4793
Augmentation	1.0	1112.842667	1112.842667	69.828708	0.0001
Model	2.0	26.610333	13.305167	0.834873	1.6191
Preprocessing : Augmentation	1.0	27.202667	27.202667	1.706914	0.7269
Preprocessing : Model	2.0	17.492333	8.746167	0.548805	2.1381
Augmentation : Model	2.0	9.220333	4.610167	0.289279	2.7594
Preprocessing : Augmentation : Model	2.0	5.082333	2.541167	0.159453	3.1281
Residual	48.0	764.964000	15.936750	NaN	NaN

▶ 유의수준(1-α): 0.05

▶ 교호작용은 없으며, 증강에 따라 정확도가 영향을 미친다는 것을 알 수 있음



향후 연구

▶ 연구결과

- 전처리, 증강, 모델에 따른 정확도 비교에 관한 연구 결과 증강에 따른 정확도의 차이가 큰 것을 알 수 있었음
- 전처리에 따른 정확도의 차이는 있었으나, 크게 영향을 끼치지 않았음
- 모델의 정확도 측면에서는 ResNet 모델이 전이 학습 모델이므로 정확도, 속도 면에서 강세를 보임

▶ 향후연구

- 전처리에 따른 모델의 정확도 증가
 - 더욱 다양한 전처리 기법 적용
- VAE, GAN 등의 증강 기법 적용을 통한 모델의 정확도 개선 및 비교