

다양한 균열 데이터를 효율적으로 생성하는 인공신경망과 그 응용 사례

Artificial Neural Network to Efficiently Generate Various Crack Data and its Applications

강남대학교 소프트웨어응용학부 201902493 윤주영

1. 프로젝트 필요성

- 최근 국내외에서 빈번히 발생하는 건축물 붕괴 사건을 **CS(Computer Science) 관점에서** 해결하고자 한다.
- 작업자가 현장에서 직접 눈으로 찾는 것은 위험 사고에 노출될 우려가 있고 많은 시간과 비용이 소모되며 작업자의 시야, 날씨와 같은 외부 변수에 의해 영향을 받아 정확도에 한계가 있다.
- 본 작품에서는 다음과 같은 문제점을 균열의 방향과 두께를 고려한 **인공신경망 기반 데이터 증강 기법을 활용하여** 사용자의 직접적인 수집 없이도 균열 데이터를 대량으로 확보하고자 한다.

다양한 균열 데이터를 효율적으로 생성하는 인공신경망과 그 응용 사례

Artificial Neural Network to Efficiently Generate Various Crack Data and its Applications

강남대학교 소프트웨어응용학부 201902493 윤주영

2. 프로젝트 목표

- ▶ 작업자의 수작업 없이 Deep Learning을 활용한 데이터 증강을 통해 대량의 균열 데이터 확보한다.
- ▶ 기존 육안 점검을 통한 인력 중심의 비효율적 요소들을 인공신경망을 통해 자동화한다.
- ▶ 본 작품은 콘크리트 균열에 제한적으로 활용될 수 있는 것이 아니며,
그 외에도 **균열과 같은 가지 형태의 패턴에 다양하게 응용할** 수 있음을 사례를 통해 보여준다.

3. 핵심 기술 (1)

SR-CNN : 균열 패턴 추출
Elastic Distortion : 균열 형태 변형

1) SRCNN, Elastic Distortion - 데이터 셋 증강 (Data Augmentation)

1) 균열 패턴 추출하는 방법 (SRCNN)

저해상도 입력 이미지(Contour 선분)에서 패치별 특징을 추출하여 고해상도(실제 균열) 패치에 대한 특징 벡터로 매핑하는 과정을 통해 고해상도 이미지를 복원한다.

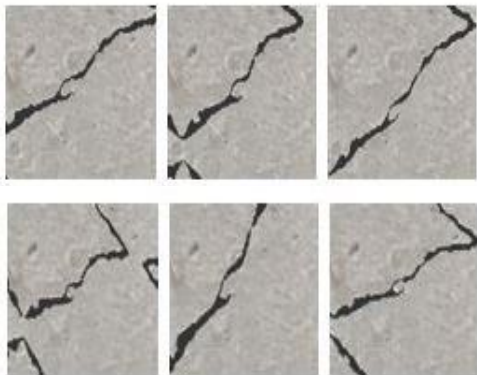


Input Data Preprocessing

Skeleton, corner-connection, contour

2) 균열 형태 변형하는 방법 (Elastic Distortion)

가까운 픽셀이 유사한 변위를 갖도록 고려하여 픽셀을 새로운 위치로 이동시켜 변형한다.



[데이터 증강 이유]

모델 학습과정에서 Overfitting(과적합) 문제점을 해결하기 위해 Data 증강이 필요하며, 시점에 따라 달라지는 형태를 학습하기 위해 데이터를 변형한다. 균열은 정형화된 형태가 없어 실제 세상에서 나타날 모든 균열 패턴을 알 수 없으므로, 실제 균열 데이터의 특징을 고려한 메타 데이터를 만들어 과적합 문제점을 방지한다.

3. 핵심 기술 (2)

[Input Data Preprocessing]

Skeleton

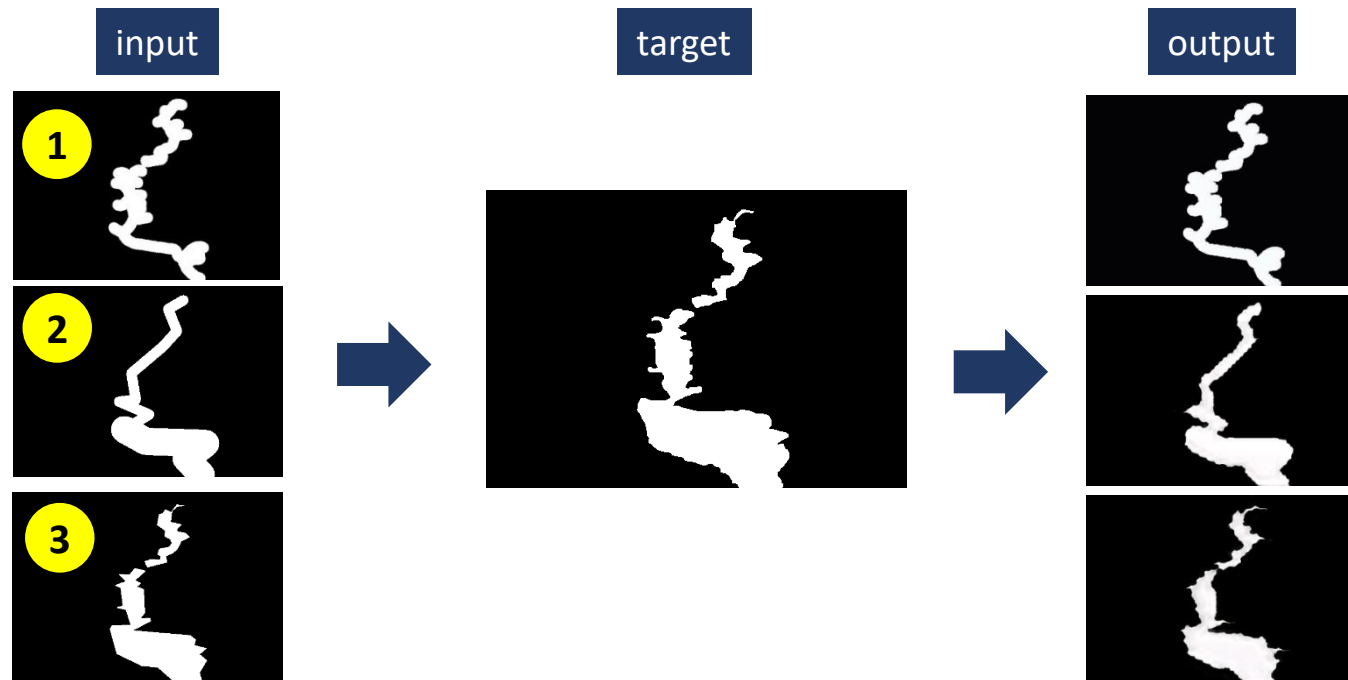
: 균열의 뼈대를 추출하여 너비 조절
문제점 | 정규화된 두께

Corner - connection

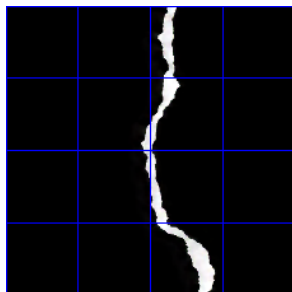
: 코너 점을 선분으로 연결, 너비 조절
문제점 | 시간적으로 비효율적

Contour -> Input Data

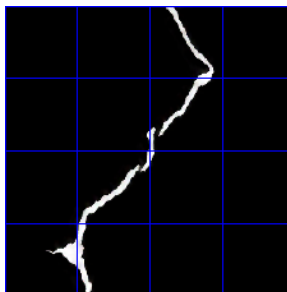
: 균열의 방향과 두께를 고려하여 자동으로
데이터 셋을 생성합니다.



+) **쿼드트리(Quad-Tree)** - 균열이 있는 영역만 활용하여 효율적인 메모리 사용



전체 : $4 \times 4 = 16$ 개
압축된 영역 : 총 6개
압축 비율 $\frac{6}{16} * 100 = 37.5\%$



전체 : $4 \times 4 = 16$ 개
압축된 영역 : 총 7개
압축 비율 $\rightarrow \frac{7}{16} * 100 = 43.8\%$

3. 핵심 기술 (3)

2) 균열 검출 테스트

■ 영상처리 기법을 통한 균열 감지

Yusuke Fujita et al., 2006;
불규칙한 노이즈를 제거하기 위해 픽셀의
평균을 활용한 이미지 스무딩 기법을 통해
가중치 값에 따라 균열을 검출

장점

영상 처리 기법은 특정 이미지를 감지하는데 효과적이다.

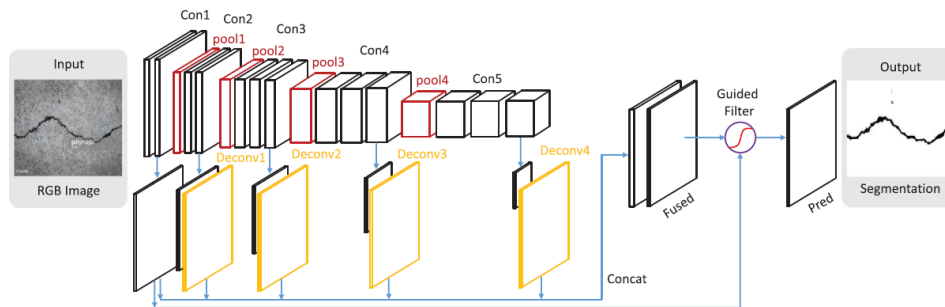
단점

균열을 촬영한 이미지는 빛, 그림자, 거친 콘크리트 표면과 같은 요
인에 영향을 받기 때문에 다양한 상황에 적용하기 어렵다.

■ 딥러닝을 통한 균열 감지

- 영상처리기법은 가중치나 계산 방식에 따라 이미지로부터 잘못된 특징을 추출할 수 있다는 문제가 있지만, 딥러닝은 이미
지 있는 그대로의 정보를 학습하여 특징이 자동으로 추출되므로 다양한 상황에 적용하기 적합하다.

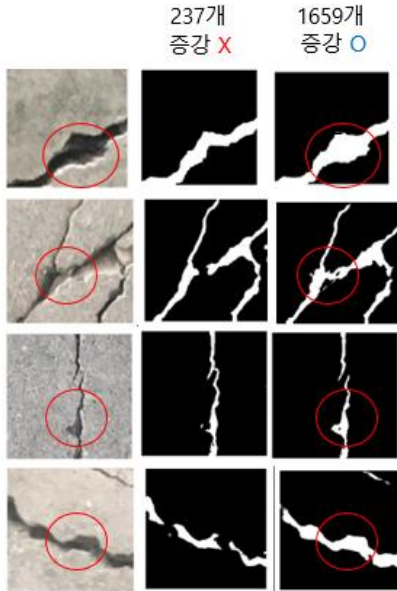
Y. Liu, J. Yao and X. Lu et al./Neurocomputing 338 (2019) 139–153



- 전처리를 진행한 균열과 마스크 데이터 세트를 학습한다.
- Test 단계에서 실제 균열 이미지를 입력으로 넣어 균열 영역을 감지한다.

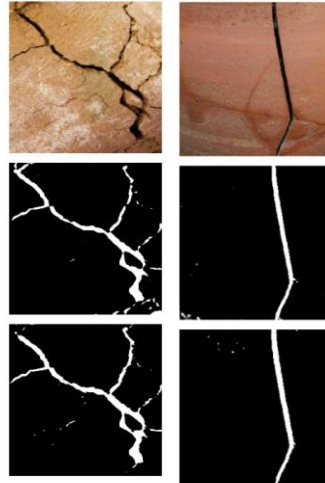
4. 결과

Test
1



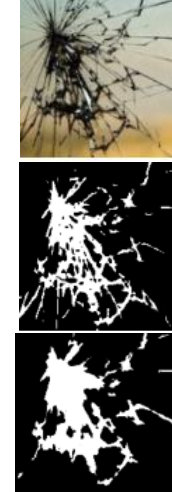
Test
2

도자기 균열
증강 데이터 (1659개)
기존 데이터 (237개)

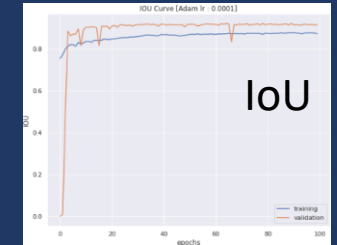
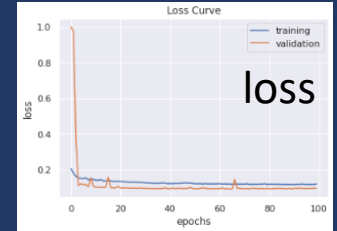


유리 균열

증강 데이터 (1659개)
기존 데이터 (237개)

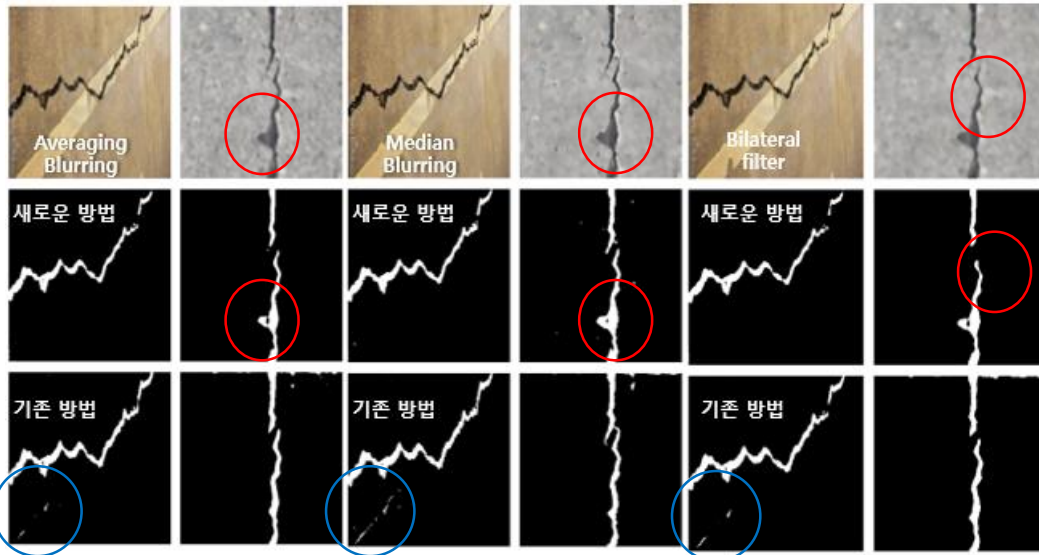


[1659개 증강 - Train Result]



Test
3

블러 효과 적용



증강 데이터 (1659개)

기존 데이터 (237개)

도자기의 균열, 유리의 균열, 콘크리트 균열, 흐린 균열 등 다양한 환경에서 테스트하였을 때, 전체적으로 6배 증강한 훈련한 새로운 방법에서 노이즈가 감소하였으며, 정확도가 향상된 점을 확인할 수 있다.

5. 평가 지표

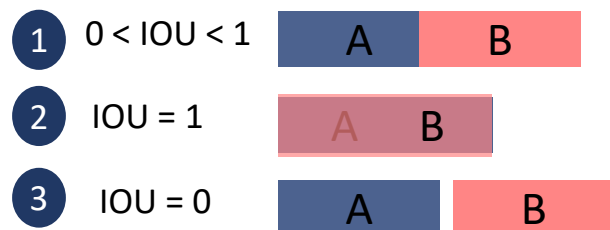
$$IOU = \frac{A \cap B}{A \cup B}$$

IoU ≥ 0.5 : 제대로 검출(TP), IoU < 0.5 : 잘못 검출(FP)

2D 이미지 분야에서 딥러닝 네트워크를 학습시킬 때 최종적인 목표는 Ground Truth 를 Prediction이 완벽하게 추종하는 것이다.

실제 값과 예측 값이 얼마나 겹치는 지를 따져 잘 예측했는지 성능을 평가하는 지표이다.

즉, 평가 지표인 IoU(Intersection over union)가 1이 되는 것을 목표로 한다.



Data	Val_IoU
기존 237개 균열	0.5803
증강 1659개 균열	0.9223

[데이터 증강 1659개 - 성능 평가]

	precision	recall	f1-score
False	1.00	1.00	1.00
True	0.97	0.95	0.96
accuracy			1.00

균열을 비균열로 감지한 확률: 5%,
비균열을 균열로 감지한 확률: 3%

6. 개발 환경

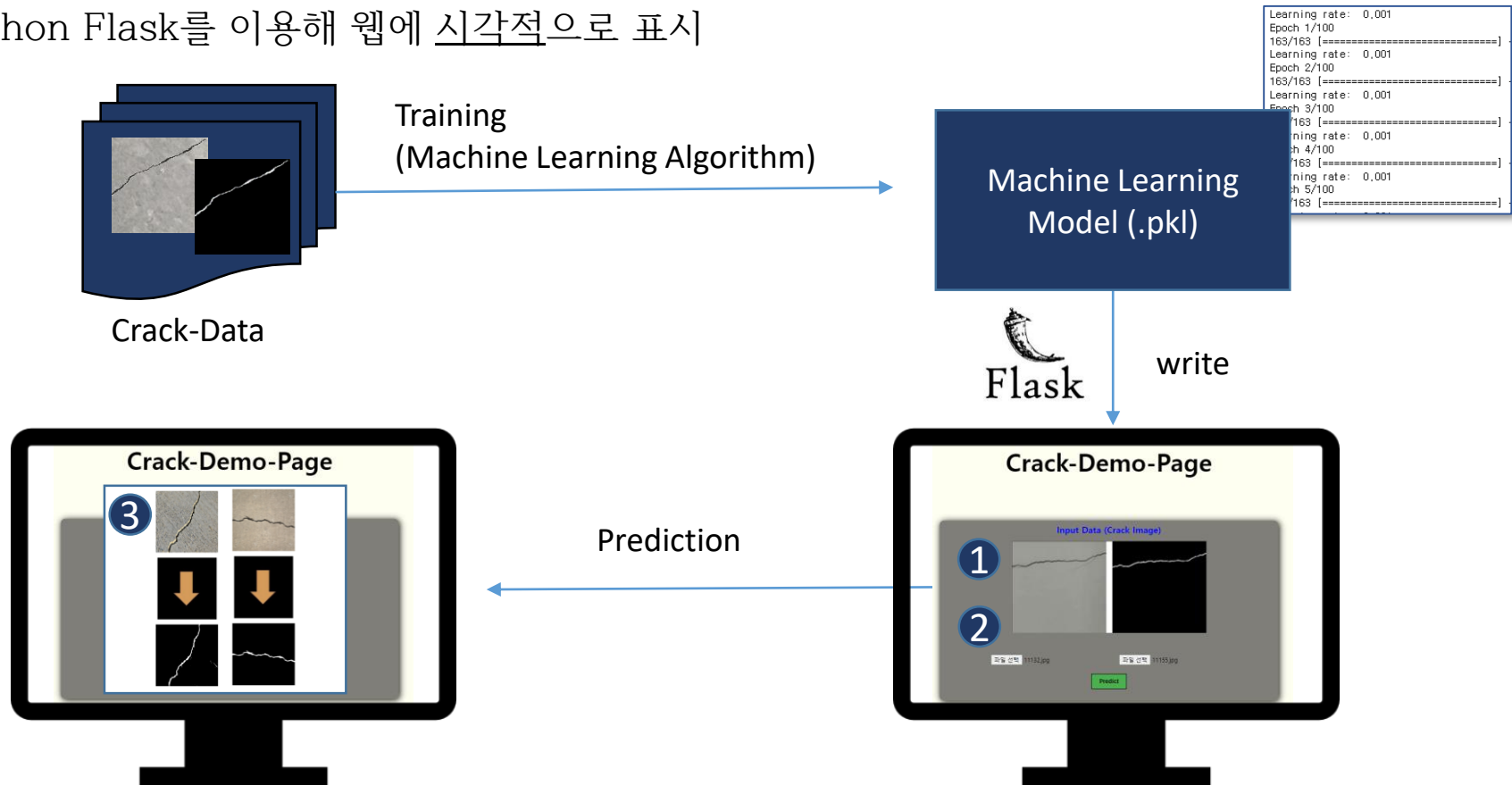
- 사용언어 : python
- 개발환경 : pycharm, anaconda
- 소프트웨어 라이브러리 : tensorflow, keras, numpy 등
- 구현 기술 : QuadTree, DeepLearning (EnhanceNet)
- 데이터 : DeepCrack의 237개 균열 이미지를 6배 증강하여 사용하였으며,

모든 데이터는 각각의 mask와 함께 세트로 이루어지며, train, test, validation 비율을 6:2:2로 나눠 사용한다.

이 데이터 세트는 얇은 균열부터 대형 균열까지 다양한 균열 형태를 반영한다.

7. Web Demo

- Python Flask를 이용해 웹에 시각적으로 표시



[페이지 기능]

- 1 자동으로 균열 감지 데모 영상 재생
- 2 파일 선택을 통해 이미지를 직접 입력한 뒤, Predict 버튼 실행
- 3 균열 이미지와 검출된 바이너리 균열 영역 이미지 세트로 출력