

정확한 균열 데이터를 효율적으로 생성하는 벡터와 두께 기반의 데이터 증강

윤주영^o, 김종현^{*}

^o강남대학교 소프트웨어응용학부,

^{*}강남대학교 소프트웨어응용학부

e-mail: jonghyunkim@kangnam.ac.kr

A Vector and Thickness-Based Data Augmentation that Efficiently Generates Accurate Crack Data

Ju-Young Yun^o, Jong-Hyun Kim^{*}

^oSchool of Software Application, Kangnam University,

^{*}School of Software Application, Kangnam University

● 요약 ●

본 논문에서는 합성곱 신경망(Convolutional Neural Networks, CNN)과 탄성왜곡(Elastic Distortion) 기법을 통한 데이터 증강 기법을 활용하여 학습 데이터를 구축하는 프레임워크를 제안한다. 실제 균열 이미지는 정형화된 형태가 없고 복잡한 패턴을 지니고 있어 구하기 어려울 뿐만 아니라, 데이터를 확보할 때 위험한 상황에 노출될 우려가 있다. 이러한 데이터베이스 구축 문제점을 본 논문에서 제안하는 데이터 증강 기법을 통해 비용적, 시간적 측면에서 효율적으로 해결한다. 세부적으로는 DeepCrack의 데이터를 10배 이상 증가하여 실제 균열의 특징을 반영한 메타 데이터를 생성하여 U-net을 학습하였다. 성능을 검증하기 위해 균열 탐지 연구를 진행한 결과, IoU 정확도가 향상되었음을 확인하였다. 데이터를 증강하지 않았을 경우 잘못된 예측(FP)된 경우의 비율이 약 25%였으나, 데이터 증강을 통해 3%까지 감소하였음을 확인하였다.

키워드: 균열(Crack), 데이터 증강(Data augmentation), 합성곱 신경망(Convolutional Neural Network, CNN), 탄성왜곡(Elastic distortion), 균열 검출(Crack detection)

I. Introduction

최근 국내외에서 빈번히 발생하는 건축물 붕괴 사건을 CS(Computer Science) 관점에서 해결하고자 한다. 균열은 작업자가 현장에서 직접 눈으로 찾을 때 위험 사고에 노출될 우려가 있고 많은 시간과 비용이 소모되며 작업자의 시야, 날씨와 같은 외부 변수에 의해 영향을 받아 정확도에 한계가 있다. 본 논문에서는 데이터 세트 확장을 통해 인공지능의 성능을 향상하였으며, 이 과정에서 균열의 방향과 두께를 고려한 인공지능 기반 데이터 증강 기법을 활용하여 사용자의 직접적인 수집 없이도 균열 데이터를 대량으로 확보하였다.

이를 구현하기 위한 기여도는 다음과 같다.

- CNN과 탄성왜곡 기법을 통해 실제 균열 패턴과 형태를 반영한 새로운 균열 이미지를 생성하는 특징 전달 네트워크 개발
- 다양한 균열에서 U-NET을 통한 균열 감지

II. The Proposed Scheme

본 논문에서 제안하는 다양한 균열 데이터를 생성하는 인공지능 기반 기법은 균열 패턴을 추출하는 CNN과 균열 형태를 변형하는 탄성 변형 기법을 활용한다[1,2]. 균열은 정형화된 형태가 없어 실제 세상에서 나타날 모든 균열 패턴을 알 수 없으며 시점에 따라 형태가 달라지므로 이를 학습하여 실제 균열 데이터의 특징을 고려한 메타 데이터를 만든다. 이는 모델 학습 과정에서의 과적합(OverFitting) 문제점을 방지한다. 증강된 데이터 세트를 바탕으로 균열 감지 네트워크를 훈련하여 실제 균열 이미지로 테스트하였을 때 정확도가 향상되었음을 확인하였다. 이를 웹에 시각화하여 사용자가 직접 이미지를 입력한 뒤 예측 결과를 확인할 수 있도록 설계하였다. 본 연구에서는 DeepCrack의 데이터 세트를 증강하여 총 3,000장을 사용하였으며 [3], 모든 데이터는 각각의 mask와 함께 세트로 이뤄지며, train, test, validation 비율을 6:2:2로 나눠 사용한다. 이 데이터 세트는 얇은 균열부터 대형 균열까지 다양한 균열 형태를 반영한다.

1. Feature transfer

인공 신경망을 통해 균열 데이터의 패턴을 추출하는 방법을 소개한다. CNN은 저해상도 입력 이미지에서 패치별 특징을 추출하여 고해상도 패치에 대한 특징 벡터로 매핑하는 과정을 통해 이미지를 복원하는 기법이다. 우리는 다음의 이미지 전처리 과정을 통해 저해상도 입력 이미지를 3가지 중 가장 우수한 contour 입력 데이터로 결정하였다.

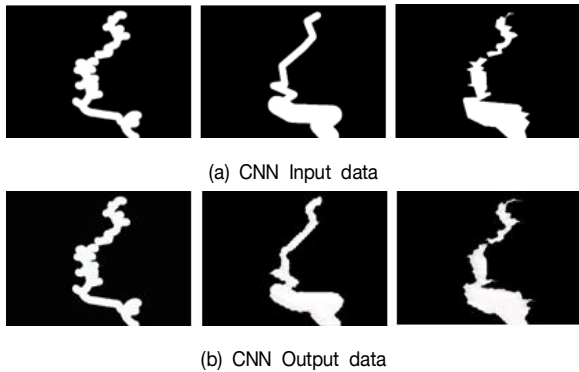


Fig. 1. Enhanced detail of crack with CNN.

(a)는 저해상도 입력 이미지이며 (b)는 합성곱 신경망 학습을 진행하여 실제 균열과 유사한 패턴을 얻은 특징전달(Feature transfer) 결과이다. 순서대로 Skeleton set, Corner set, Contour set이다. Skeleton set 이미지는 균열의 뼈대를 추출하여 너비를 조절한다는 점에서 개별 균열 두께가 아닌 정규화 되어버리는 문제점이 있다. Corner set 이미지는 에지 간 변화율을 계산하여 양방향 기울기 변화가 심한 곳을 코너로 간주하며 이를 직접 연결한 선분의 너비를 조절하므로 시간적으로 비효율적이다. 마지막으로 Contour 데이터 세트는 균열의 방향과 두께를 고려하여 자동으로 생성하므로 이를 입력 데이터로 결정하였다.

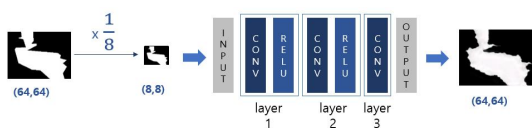


Fig. 2. Our neural network architecture.

Contour set의 저해상도 입력 이미지를 바탕으로 합성곱 신경망 학습을 진행하여 실제 균열과 유사한 패턴을 추출하였다.

2. Elastic distortion

탄성 왜곡 기법은 픽셀이 유사한 변위를 갖도록 표준편차 σ 와 강도 α 의 가중치를 조절하여 새로운 위치로 이동시켜 변형한다. 이때, 회전 각도와 아핀 변형(Affine transformation)의 가중치를 조절하여 균열 표면 형태를 함께 변형하였다. 이는 미시적-거시적 관점에서 다양한 방향성을 표현할 수 있다.

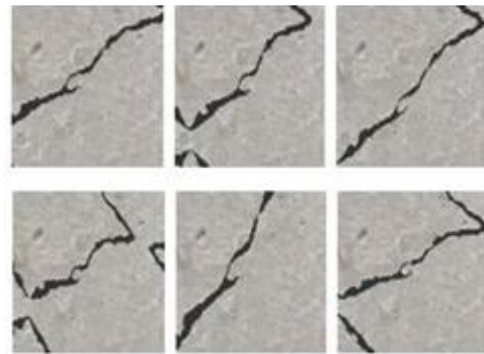


Fig. 3. variety crack data set with our method.

Fig 3. 에서는 CNN과 탄성 변형 기법을 통해 증강한 바이너리 이미지를 콘크리트 균열 배경 이미지와 합성을 하여 생성한 새로운 증강 균열 데이터이다.

3. Adaptive optimization

공간 분할 기법인 쿼드트리를 활용하면 연산이 필요한 데이터를 분할 연산하므로 메모리를 효율적으로 사용할 수 있다는 장점이 있다.

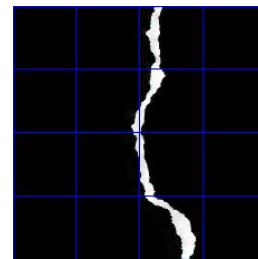


Fig. 4. Quadtree structures.

Fig 4.는 전체 4x4의 16개 영역에서 균열이 있는 영역인 6개의 영역만 사용하므로 37.5%로 이미지를 압축 사용하여 메모리 부족 문제를 피할 수 있다.

4. Crack detection test

선행 연구로 초기에 영상 처리(Image Processing)기술을 활용한 Yusuke Fujita et al, 2006 의 연구가 진행되었다[4]. 이는 불규칙한 노이즈를 제거하기 위해 픽셀이 평균을 활용한 이미지 스무딩 기법을 통해 가중치 값에 따라 균열을 검출한다. 특정 이미지를 감지하는데 효과적일 수 있겠으나 균열을 촬영한 이미지는 빛, 그림자, 거친 콘크리트 표면과 같은 요인에 영향을 받기 때문에 다양한 상황에 적용하기 어렵다는 한계점이 있다. 즉, 영상처리 기법을 통한 균열 감지 기법은 가중치나 계산 방식에 따라 이미지로부터 잘못된 특징을 추출할 수 있다는 문제가 있다.

최근에는 딥러닝을 활용하여 이미지의 있는 그대로의 정보를 학습하여 특징을 자동 추출하는 연구가 진행되었다. 본 논문에서는 다양한 상황에 적용하기 적합한 CNN을 기반으로 한 U-Net 모델을 활용하여

균열 검출 방법을 제안한다. 전처리를 진행한 균열과 마스크 데이터 세트를 U-Net을 통해 함께 학습하여 Test 단계에서 실제 균열 이미지를 입력으로 넣어 균열 영역을 감지하였다. U-Net의 Architecture는 그림 1. 과 같이 U자형 아키텍처로 구성된다[5].

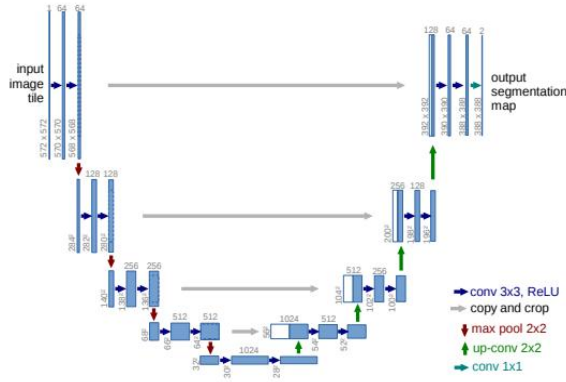


Fig. 5. U-net architecture

가중치가 학습되는 과정을 수식으로 표현하면 다음과 같다.

$$w(x) = w_c(x) + w_0 \cdot e^{-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}} \quad (1)$$

w_0, σ : Weight hyper-parameter

$d_1(x)$: 픽셀 x 로부터 가장 가까운 경계까지의 거리

$d_2(x)$: 픽셀 x 로부터 두 번째로 가까운 셀의 경계까지의 거리

5. Evaluation

평가 지표(Metric)로는 Mean IoU(Intersection over union)을 사용하였으며, 딥러닝 모델이 예측한 결과와 실제 값 사이의 교집합, 합집합 비율을 측정하여 모델을 평가한다.

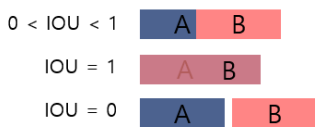


Fig. 6. IoU Metric

IoU는 값이 0.5 이상이면 제대로 검출(TP), 0.5 미만이면 잘못 검출(FP)되었다고 정의한다.

$$IoU = \frac{A \cap B}{A \cup B} \quad A: \text{target } B: \text{prediction}$$

$IoU \geq 0.5$: TP(True Positive)

$IoU < 0.5$: FP(False Positive)

데이터를 증강하지 않았을 경우 잘못 예측된 경우의 비율이 약 25%였으나, 10배 이상의 데이터 증강을 통해 3%까지 감소시킬 수 있었으며, IoU 정확도를 측정했을 때 0.58에서 0.93으로 향상하였다.

6. Web

위에서 제시한 데이터 증강을 통해 얻은 데이터들을 바탕으로 모델 훈련 결과를 저장하고 Python Flask프레임워크를 이용해 웹 (WEB)으로 구현하였다. Fig7. 는 과정을 표현한 것으로 균열 이미지를 바탕으로 훈련한 모델을 (.pkl) 파일로 저장한 뒤 Flask를 활용하여 웹에 예측 결과를 시각화한다. 페이지 기능으로는 자동으로 균열 감지 데모 영상이 재생되며, 사용자가 파일 선택을 통해 직접 균열 이미지를 입력할 수 있다. Predict 녹색 버튼을 실행하면 선택한 이미지를 바탕으로 균열이 감지된 영역을 바이너리 이미지로 시각화하여 함께 보여준다.

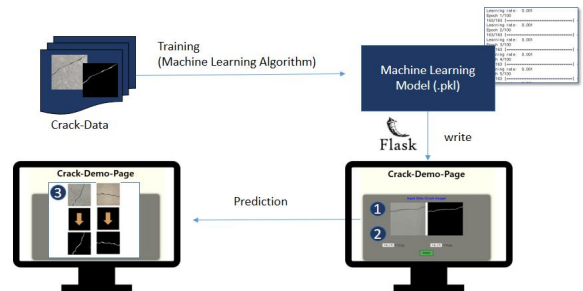


Fig. 7. Flask Web demo

III. Results

블러 효과를 적용한 흐린 균열, 유리 균열, 콘크리트 균열 등 다양한 환경에서 테스트하였다. 다음은 블러 효과를 적용한 균열 검출 결과이다.

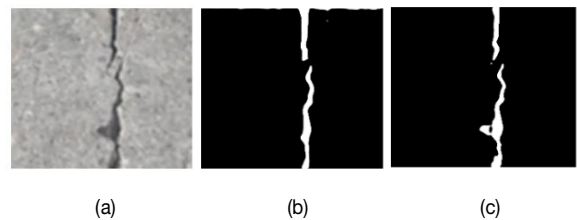


Fig. 8. Test Result.

Fig 8의 (a)는 Averaging Blur를 적용한 실제 균열 이미지이며, (b)는 증강하기 전 검출 결과, (c)는 데이터 증강 후 검출 결과이다. 데이터 증강한 모델로 훈련한 결과에서 (b) 상단에서 나타난 노이즈가 감소하고 정확도가 향상된 점을 확인하였다.

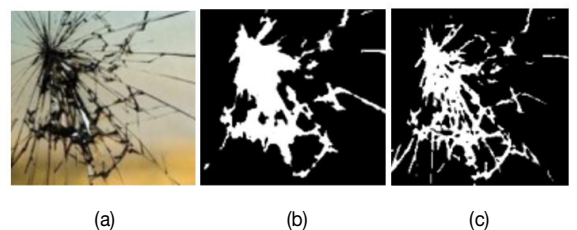


Fig. 9. Test Result.

Fig 9.는 깨진 유리에서 균열 검출 테스트를 진행한 결과이며, 데이터 증강하기 전의 결과(b)보다 증강 후 결과(c)에서 선명하게 검출해낸다는 점을 확인하였다.

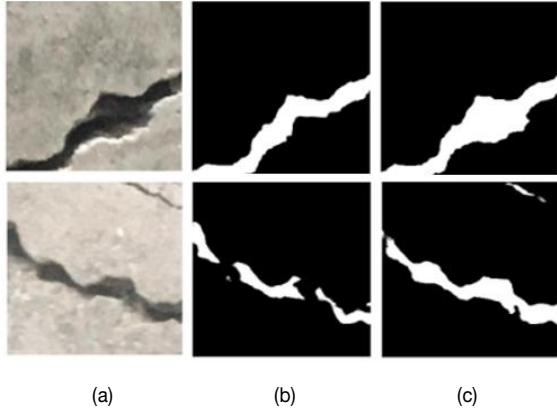


Fig. 10. Test Result.

Fig 10.은 콘크리트 균열에서 균열 검출 테스트를 진행한 결과이며, 데이터 증강하기 전의 결과(b)보다 증강 후 결과(c)에서 넓은 영역이 확실하게 검출되고 끊김 현상 없이 선명하게 감지해낸다는 점을 확인하였다.

IV. Conclusions

본 연구에서는 데이터 증강 기법을 통해 데이터를 확장하고 균열 감지 테스트를 진행하여 결과를 WEB으로 시각화함으로써 시스템 상용화를 도모하고자 하였다. 작업자의 수작업 없이 인공지능망을 활용한 데이터 증강을 통해 대량의 균열 데이터를 확보할 수 있으며 기존 육안 점검을 통한 인력 중심의 비효율적 요소들을 인공지능망을 통해 자동화하였다. 데이터 확장을 통해 균열 감지 성능이 향상하였음을 확인하였다. 이는 콘크리트 균열에 제한적으로 활용되는 것이 아닌 균열과 같은 가지 형태의 패턴에 다양하게 응용할 수 있다. 향후 연구에서는 다른 딥러닝 모델을 활용하여 모델 간의 성능을 비교 평가하여 최적의 모델을 탐색하고자 한다.

REFERENCES

- [1] Shorten, Connor, and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, vol. 6, pp. 1-48, 2019.
- [2] Wang, Jason, and Luis Perez. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, 2017.
- [3] Liu, Yahui, et al. "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation."

- Neurocomputing* 338 (2019): 139-153
- [4] Yusuke Fujita, Yoshihiro Mitani and Yoshihiko Hamamoto, "A Method for Crack Detection on a Concrete Structure", *International Conference on Pattern Recognition*, 2006
- [5] Olaf Ronneberger et al., "U-Net: Convolutional Networks for Biomedical Image Segmentation", *MICCAI* (2015): 234-241