

## POC : 인공지능 기반 균열 탐지를 위한 데이터셋 구축

김지호<sup>o</sup>, 김경영\*, 김동주\*

<sup>o</sup>포항공과대학교 인공지능연구원,

\*포항공과대학교 인공지능연구원

e-mail: {kimjiho, gy\_kim, kkb0320}@postech.ac.kr<sup>o\*</sup>

## POC : Establishing Dataset for Artificial Intelligence-based Crack Detection

Ji-Ho Kim<sup>o</sup>, Gyeong-Yeong Kim\*, Dong-Ju Kim\*

<sup>o</sup>Postech Institute of Artificial Intelligence, Pohang University of Science and Technology,

\*Postech Institute of Artificial Intelligence, Pohang University of Science and Technology

### ● 요약 ●

건축물 안전 점검은 대부분 전문가의 현장 방문을 통한 육안검사다. 그중 균열 검사는 건물 위험도를 나타내는 중요한 지표로써 발생 위치, 진행성, 크기를 조사하는데, 최근 균열 조사 방식에 대해 객관성과 체계성을 보완할 딥러닝 개발이 활발하다. 그러나 균열 이미지는 외부 현장에 모양, 규모도 많은 종류라 도메인이 다양해야 하는데 대부분 제한된 환경과 실제적인 균열 검사와는 무관한 데이터로 구성되어 실효적이지 않다. 본 연구에서는 균열 조사에 적합하고 Wild 환경에 적용 가능한 POC 데이터셋을 소개한다. 기존 균열 공인 데이터셋 4종의 특징과 한계점을 분석을 토대로 고해상도 이미지로써 균열의 세부 특징을 담았고 균열 유사 환경과 조건들을 추가 촬영해 균열 검출에 강인하게 학습되도록 지향하였다. 정제 및 라벨링 작업을 거친 POC 데이터셋은 균열 검출모델인 YOLO-v5으로 성능을 실험하였고, mAP(mean Average Precision) 75.5%로 높은 검출률을 보였다. POC 데이터셋으로 더욱 도메인에 적응적(Domain-adapted)인 인공지능 모델을 개발하여 건물, 댐, 교량 등 각종 대형 건축물에 대한 안전하고 효과적인 안전 관리 도구로써 활용할 것을 기대한다.

**키워드:** 균열 검출(Crack Detection), 인공지능(Artificial Intelligence), 데이터셋 구축(Dataset Construction)

### 1. Introduction

건물 안전 점검은 건물에 발생한 물리적인 충격, 철근의 부식, 지진의 여파 등 다양한 요인에 따른 건물의 안전 여부를 점검하는 작업이다. 건물 안전 점검은 점검 자격을 취득한 전문가가 현장에서의 건축물에 대한 육안 검사를 통해 수행한다. 그 중 균열, 박리 등을 통해 건물의 위험도를 가장 직관적으로 확인할 수 있을 뿐 아니라, 실제적인 위험도의 지표가 될 경우가 많아 이에 대한 정확하고 체계적인 검사 방식이 필요하다[1].

최근 체계적인 균열 탐지를 통한 검사를 보완할 딥러닝 기반의 다양한 모델들과 데이터셋이 등장하고 있다[2,3,4,5]. 균열 탐지(Crack Detection)는 이미지 프로세싱을 통한 균열의 구획화(Image Segmentation)나 분류(Image Classification) 작업을 중심으로 세분화된다. 그러나 해당 작업으로 이뤄진 데이터셋을 기반으로 균열 탐지를 하고자 할 때, 결과가 나올 때의 이차적인 후 작업을 거치거나 직관적인 균열의 위치를 탐지하는 데에 어려운 한계가 있다[6]. 실제

육안 검사 방식과의 괴리가 존재해 건물 점검에 대한 효율성도 떨어진 다. 본 문제를 해결할 방식으로써 End-To-End로 검출할 수 있는 탐지용 건물 균열 데이터셋이 필요한데 이 또한 전무하였다.





본 연구에서는 균열 검출에 대해 실증적 연구에 활용할 수 있는 POC(Pohang Crack) 데이터셋 구축을 목표로 한다. 본 데이터셋 구축을 위해 (1) 기존 대표적인 균열 데이터셋의 종류와 특징, 한계점에 대해 살펴보고, (2) 포항시 내 노후 건축물군과 지진피해 지역의 건물을 조사하여 선정하였으며, (3) 데이터 촬영과 관련한 체계를 구축 후 촬영하였고, (4) 촬영된 데이터셋에 대한 정제와 라벨링 작업을 통해 구축하였다. 추가적으로 딥러닝 검출 모델로써 범용적으로 쓰이는 YOLO-v5 모델을 통해 다른 데이터셋과의 성능을 비교함으로써, 해당 데이터셋과 다른 유사 데이터셋의 품질을 분석하였다. 본 연구를 통한 데이터셋 적용을 통해 현장 적용에 용이한 딥러닝 모델을 만들 수 있을 것으로 기대한다.

## II. Preliminaries

### 1. Related works

균열 탐지와 관련한 공인 데이터셋을 탐색한 결과 크게 네 가지로 분류됨을 확인하였으며 각각의 데이터 관련 통계는 Table 1과 같다. METU 데이터셋은 중동 공과대학교(Middle East Technical University, METU)의 캠퍼스 내 빌딩 균열을 촬영한 데이터셋이다 [2]. METU 데이터셋은 균열 전체를 촬영한 원본 이미지를 작은 사이즈로 분할(Image Cropping)하고, 분할한 이미지들로 데이터셋을 구축하였다. 균열 분류를 목적으로 만들어진 데이터셋이기 때문에 전체 이미지를 조각 조각 잘라놓아 균열의 시작점과 끝점의 모양새, 전체적인 형태 등 균열의 특징을 알기에 어려운 부분이 많다.

Table 1. 균열 탐지 관련 공인데이터셋 정리

	METU[2]	SDNET[3]
대표 이미지		
이미지 크기	227 × 227	256 × 256
클래스 종류	균열/비균열	균열/비균열
데이터 개수	40,000	56,000
	CSD[4]	AIHub[5]
대표 이미지		
이미지 크기	448 × 448	2560 × 1440
클래스 종류	균열/비균열, 구획화	콘크리트 균열, 박리, 백태/누수, 철근 노출
데이터 개수	11,298	350,000

SDNET은 유타 주립대학(Utah State University) 캠퍼스 내의 벽, 도로 등에 대한 균열 데이터셋이다[3]. SDNET 데이터셋은 METU와 마찬가지로 균열 분류를 목적으로 만들어졌기 때문에 확대된 이미지를 중심으로 촬영되었으며, 벽 균열과 특성이 다른 도로 균열을 다수 포함하는 한계를 지닌다. 데이터 수가 많으나 대부분이 균열과 유사한 파인 자국, 훼손 등의 데이터로 구성되어 있고, Outlier의 종류가 많은 한계를 지닌다.

CSD(Crack Segmentation Dataset)는 12개의 균열 구획화 데이터셋들을 모아 새롭게 구성한 데이터셋이다[4]. CSD 데이터셋은 균열을 촬영한 이미지를 모두 448 × 448로 통일하여 해상도를 낮추었다. 이에 따라 이미지의 세부 정보가 손실되어 이미지 내에 존재하는 미세한 균열은 보이지 않을 가능성이 크다.

AIHub 데이터셋은 AIHub 내 제공하는 건물 균열 탐지 이미지 데이터셋이다[5]. AIHub의 데이터셋은 드론을 활용하여 콘크리트 면의 균열을 직접 촬영하였으며, 대부분 2560 × 1440의 고해상도로 활용 가능성이 높고 데이터 규모가 크다. 하지만 해당 데이터셋의 공식 촬영 가이드 내용 중에는 1) 해를 바라보고 찍지 않을 것, 2) 그림자가 없도록 찍을 것 등 촬영 시 제약 조건이 있다. 이는 빛의 변화를 통제할 수 없는 환경적 도메인에서 데이터를 수집하였다고 볼

수 있으며 다양한 도메인에 적용하기에 어렵다.

본 연구에서는 이러한 한계점을 보완할 수 있는 데이터 수집을 목표로 한다. 고해상도로 촬영하여 균열의 세부 특징을 추출할 수 있고, 균열로 오인하기 쉬운 그림자, 장애물 등을 포함한 데이터를 포함하여 실제 환경에서의 오탐지를 줄일 수 있도록 도울 데이터셋을 구성하였다.

## III. The Proposed Scheme

균열 데이터셋 구축을 위해 (1) 데이터 촬영, (2) 데이터 정제 및 라벨링, (3) 검출 모델 적용 및 분석으로 나누어 소개하고자 한다.

### 1. 균열 데이터셋 촬영

데이터 촬영은 포항 지역의 건물 벽면 균열을 촬영하여 POC 데이터셋이라는 이름으로 구축하였다. 데이터셋 수집은 2017년 포항 지진으로 인해 피해를 받은 건축물과 노후 건축물을 대상으로 총 7개 장소 인근에서 촬영하였다(Fig. 1). 건축물 안전 검사의 간소화를 염두하여 스마트폰 기본 기능의 카메라로 촬영을 하였고 기존 공인 데이터셋에서 유사성이 높은(수평/수직으로 약간 옮긴 후 촬영 경우 등) 구성이 많은 것에 비해 POC의 데이터셋은 최대한 유사성이 없도록 하여 총 4,643장의 균열 이미지로 구성하였다. 실제 건축물 안전 점검을 수행할 때는 균열 폭의 구체적인 수치를 측정하거나 균열이 발생한 위치와 추가 균열 발생 위험을 조사하기 때문에 이러한 현장 점검의 특성을 고려하여 균열의 세부 특징을 잘 담아야 할 수 있는 4000 × 3000의 고해상도로 데이터를 수집하였다.



Fig. 1. 포항 지역 내 균열 데이터 수집 위치

균열 탐지 기술 고도화의 일환으로써 균열과 유사하여 모델 오검출이 나타날 위험이 있는 이미지가면서 실제 현장에서도 자주 접할 수 있는 데이터를 촬영하였다(Fig. 2). 첫 번째 유형은 면의 디자인이 균열과 유사한 경우로, 균열로 오검출하는지를 확인할 수 있는 용도로 활용된다. 두 번째는 벽돌 간의 틈새가 짙어 균열로 보일 수 있는 경우이다. 세 번째 유형은 나무나 철망과 같은 큰 구조물에 의해 그림자가 짙게 진 유형으로, 그림자와 균열의 차이를 구별할 수 있는지

의 여부를 판단한다. 네 번째는 인쇄물 혹은 도장 형태로 글자나 숫자가 함께 촬영된 경우이다. 위와 같은 특이 데이터를 포함하여 학습한 모델로 다양한 균열 유사 형태에 대한 오검출을 방지하는데 도움이 될 것으로 기대한다.

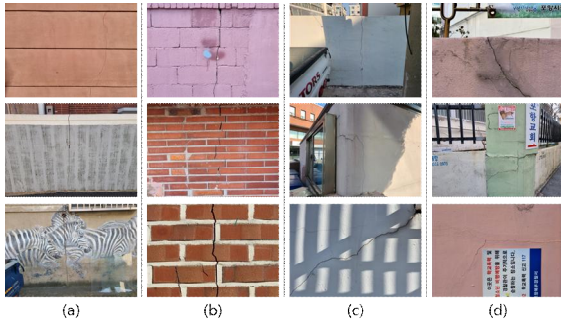


Fig. 2. POC 특이 데이터 유형별 대표 이미지 (a) 균열 유사 벽면 디자인 (b) 벽돌 틈새 (c) 그림자 (d) 글자 문구 포함

## 2. 균열 데이터셋 정제 및 라벨링

검출용 딥러닝 모델을 위한 라벨링 작업으로써 이미지별 균열의 좌표 위치를 매칭하였다. 기존 균열 검출과 관련한 자동화된 검출 라벨링 프로세스가 비효율적이었던 점을 고려하여[6], 수집한 데이터 4,643장에 대해 수동적으로 라벨링하였고, 라벨링 규칙 중 일부는 Table 2와 같다. Fig. 3의 라벨링 예시와도 같이, 모든 경계박스(Bounding Box)와 균열 사이에 남는 공간이 거의 없도록 박스를 생성하여 정확한 라벨 생성을 첫 번째 기준으로 삼았다.

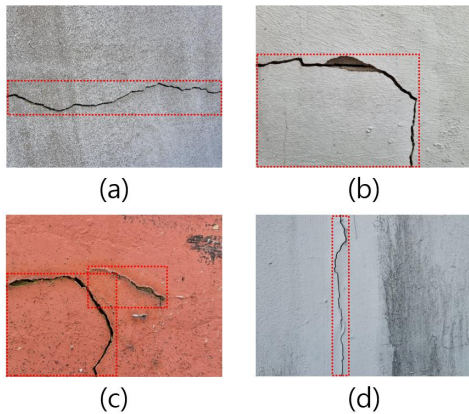


Fig. 3. POC 데이터셋 및 라벨링 예시

Table 2. 라벨링 기준

경우	취급
1 두꺼운 구조물에 의해 균열이 가린 경우	별개의 균열로 취급
2 두 개 균열이 가까운 위치에 있는 경우	별개의 균열로 취급
3 페인트가 불거져 균열처럼 보이는 경우	균열이 아닌 것으로 취급
4 균열 내부가 보수된 경우	균열이 아닌 것으로 취급
5 페인트가 갈라진 경우	균열이 아닌 것으로 취급

## 3. 균열 검출 모델 적용 및 분석

해당 데이터셋에 대한 검출용 딥러닝 모델에서의 성능을 확인하였다. 검출 모델의 연구 방식은 기존 균열 검출 연구에서 적용되었던 방식과 성능 지표를 차용하였으며[6], 검출 모델로 범용적이고 높은 성능을 보이는 Yolo-v5모델[7]을 통해 검출 모델을 학습 및 평가하였다. 기존 데이터셋과는 달리 POC의 데이터규모가 상대적으로 작아 Scratch 형식의 학습에는 미적합(Underfitting)한 것을 확인하였고 상기 공인 데이터셋으로 학습한 모델에서 미세 조정(Fine-Tuning)하여 학습하였다. 미세 조정된 모델을 통해 최종적으로 정밀도는 0.758, 재현율은 0.733, mAP은 0.755가 나오는 것을 확인하였으며 평가한 데이터의 예시는 Fig. 4와 같다.

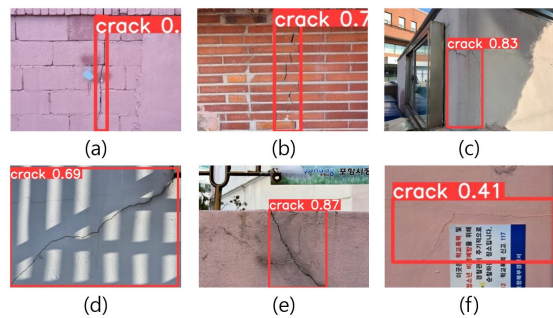


Fig. 4. POC 데이터셋 기반 모델 구동 결과 샘플

기존 다른 데이터셋을 통해 학습된 결과가 최소 0.88 이상의 mAP였던 점을 고려하면 값이 0.13 이상 낮아진 성능으로 확인하였다. POC 데이터셋은 특성상 각 데이터마다의 유사성이 다른 데이터셋에 비해 작고 검출 난이도가 어려운 것에 비해, 학습된 공인 데이터셋의 경우 대부분 확대된 상태에서의 균열로 검출 난이도가 쉽기 때문에 나타난 현상으로 분석한다. 다른 공인 데이터셋과 달리 특이 상황에 대한 데이터의 구성이 존재한다는 점에서도 오검출이 특히 발생하는 원인으로 추정된다. 특이 데이터에 대한 검출 모델의 구동 시 균열의 윤곽이 어느 정도 있을 때 잘 검출해내는 경우도 있지만, 많은 경우 균열을 잘 검출해내지 못하는 점을 확인하였으며, 이는 오검출을 방지하기 위한 추가적인 데이터의 학습을 통해 해결할 수 있을 것으로 분석된다.

## IV. Conclusions

본 연구에서는 기존 균열 데이터셋의 한계를 파악하고, 해당 한계를 해결할 방안으로써 포항시 내의 건물 균열을 통한 인공지능 학습용 데이터셋 POC를 구축하였다. POC는 기존 데이터셋에 비해 각 데이터마다의 편차가 크게 구성된 고화질 데이터로, 실제 현장 상황과 높은 유사성을 띠고 균열과 유사한 이미지에 대한 대처를 위해 다양한 조건에서 접근했다는 점에서 의의가 있다. 다양한 상황을 대처하려는 것에 비해 데이터 규모가 상대적으로 작아 딥러닝 기반 검출 모델에 있어서 목표 대비 낮은 성능을 보였다. 그러나 기존 도메인 적용 이슈 문제에 대한 해결책으로써 가능성을 확인하였고, 데이터 추가 수집 및 학습 전략 개선과 데이터 증강기법을 적용하여 실제 현장에서

활용할 수 있는 균열 검출의 도구로써 보완될 수 있길 기대한다.

## ACKNOWLEDGEMENT

본 논문은 행정안전부 지역맞춤형 재난안전 연구개발 사업의 지원을 받아 수행되었으며(20015427), 2022년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2022R1A6A1A03052954)

## REFERENCES

- [1] Kim, Ah-Ram, et al. "Crack Detection of Concrete Structure Using Deep Learning and Image Processing Method in Geotechnical Engineering." *Journal of the Korean Geotechnical Society*, vol. 34, no. 12, the Korean Geotechnical Society, Dec. 2018, pp. 145-154, doi:10.7843/KGS.2018.34.12.145.
- [2] Özgenel, Ç. F., and A. Gönenç Sorguç. "Performance comparison of pretrained convolutional neural networks on crack detection in buildings." *Isarc. proceedings of the international symposium on automation and robotics in construction*. Vol. 35. IAARC Publications, 2018.
- [3] Dorafshan, Sattar, Robert J. Thomas, and Marc Maguire. "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks." *Data in brief* 21 (2018): 1664-1668.
- [4] Middha, L., Crack segmentation dataset, <https://www.kaggle.com/lakshaymiddha/crack-segmentation-dataset> (November 5, 2021)
- [5] Building Crack detection Image dataset(AIHub), <https://aihub.or.kr/aidata/34141> (November 23, 2021)
- [6] Kim, Gyeong-Yeong, Ho-Ryeong Lee, and Dong-Ju Kim. "Robust Detection Deep Learning Model in the Various Exterior Wall Cracks." *Proceedings of the Korean Society of Computer Information Conference*. Korean Society of Computer Information, 2021.
- [7] Yolov5: <https://github.com/ultralytics/yolov5>