

# 건축물 점검을 위한 딥러닝 기반의 도메인 적응적 균열 검출 시스템

김 지 호\*, 김 경 영\*, 김 동 주<sup>o</sup>

## Domain Adaptive Deep Learning-Based Crack Detection for Building Inspection

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

### 요 약

건축물 안전 점검은 점검자가 육안으로 건축물을 조사하는 방식이며, 특히 건물의 위험을 직관적으로 나타내는 균열에 대한 점검이 핵심적이다. 최근에는 이를 보조할 수 있도록 균열 점검의 자동화를 위한 인공지능 기반의 연구가 활발히 이루어지고 있다. 균열은 벽면의 종류와 발생 위치 등에 따라 그 형태가 매우 달라지기 때문에 학습 데이터의 폭이 넓어야 하나, 대부분 공인된 데이터셋들은 제한적인 환경과 낮은 해상도로 구성되어 있어 실제 현장에 적용하기엔 실효성이 부족하다. 본 연구에서는 건물 내 균열 점검 현장에 적합한 검출 시스템을 개발하였다. 이를 위해 균열 조사 현장과 같은 이미지 데이터셋 POC를 11,466장을 구축하였고, 데이터의 특성과 분포를 고려한 증화 표집 방법을 통해 구축 데이터셋의 높은 다양성을 보장하였다. 또한, 딥러닝 기반 객체 검출 모델인 YOLO 계열의 모델들로 균열 검출 성능을 비교하였고, 최종적으로 YOLO-Cr 모델을 개발하여 mAP(mean Average Precision) 91.5%라는 높은 균열 검출 성능을 보였다. 더 나아가, YOLO-Cr을 토대로 실제 현장점검에서 실시간 스캐닝과 UAV를 활용하는 검출 시스템을 설계함으로써 기존 육안 검사의 개선 방법을 제시한다. 본 연구를 통해 다양한 구조물의 안전 점검 자동화에 기여하고 점검의 객관성 및 효율성이 증대될 수 있길 기대한다.

**키워드** : 균열 탐지, 인공지능, YOLO, 데이터셋 구축, 균열 점검 시스템

**Key Words** : Crack Detection, Artificial Intelligence, YOLO, Dataset Construction, Crack Inspection System

### ABSTRACT

In building safety inspection, the inspection of cracks plays a significant role in the process since it intuitively indicates the dangerousness of the facility. To aid the automation of crack inspection, development of an artificial intelligence-based model has been ongoing. However, the development of a robust crack detection model faltered due to the limitations of existing crack datasets. Depending on the type and location

※ 본 연구는 행정안전부 지역맞춤형 재난안전 연구개발 사업의 지원을 받아 수행된 연구입니다(20015427).

※ 본 연구는 2023년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업입니다(No.2022R1A6A1A03052954).

※ 본 연구는 과학기술정보통신부의 재원으로 한국지능정보사회진흥원의 지원을 받아 구축된 “건물 균열 탐지 이미지”를 활용하여 수행된 연구입니다. 본 연구에 활용된 데이터는 AI 허브(aihub.or.kr)에서 다운로드 받으실 수 있습니다.

• First Author : POSTECH Institute of Artificial Intelligence, Pohang University of Science and Technology, kimjiho@postech.ac.kr, 정회원

◦ Corresponding Author : POSTECH Institute of Artificial Intelligence, Pohang University of Science and Technology, kkb0320@postech.ac.kr, 정회원

\* POSTECH Institute of Artificial Intelligence, Pohang University of Science and Technology, gy\_kim@postech.ac.kr, 정회원

논문번호 : 202212-288-B-RN, Received December 1, 2022; Revised January 30, 2023; Accepted February 19, 2023

of the crack on the wall, cracks vary greatly requiring a wide variety of domains in the training dataset. In this paper, we propose a domain-general crack detection model specializing in building safety inspection environments along with introducing POC(Pohang Crack), a crack dataset consisting of 11,466 images manually collected and annotated, ensuring diversity of characteristics and distribution of data through stratified sampling method. By comparing the detection performance of deep learning-based YOLO models, YOLO-Cr was developed, showing the highest mAP(mean Average Precision) of 0.915. Moreover, we introduce a detection system utilizing real-time scanning and UAV to aid the visual inspection. It is expected that this study will contribute to the automation of safety inspection of various structures and increase the objectivity and efficiency of inspection.

## I. 서 론

건축물 정기 점검은 건축물의 수명을 단축하는 결함을 감지하고 적절한 유지보수를 통해 건축물이 안전토록 관리하기 위해 실시된다. 2021년 국토교통부 건축물 현황 발표에 의하면, 전국에 30년 이상이 된 건축물이 전체 건축물의 39.6%에 달하는 만큼 건축물의 안전 여부를 주기적으로 파악하고 검토할 필요가 있다. 건축물 정기 점검은 공인된 전문가에 의해 시행되는 건축물 육안 점검으로, 점검자가 직접 방문하여 건물 전반의 안전 요소들(균열, 누수, 박리, 박락, 층분리, 백태, 철근 노출 등)을 조사한다<sup>[1]</sup>. 그중 균열은 건축물이 위험한 정도를 표시하는 가장 대표적인 항목으로써 균열의 발생 여부와 진행성에 대한 체계적이고 철저한 검사가 필수적이다. 하지만 현재 균열에 대한 점검 방식은 반복적인 건물 점검 업무에 인력 소모가 크고, 작업자 간 업무 역량 차이나 피로도 등에 따라 판단이 달라질 수 있다는 한계가 있다. 이에 따라 점검의 객관성을 높이고 자동적, 효율적으로 건물 외벽을 점검할 수 있는 수단에 관한 관심이 늘어나는 추세다.

균열 인식의 자동화를 위한 연구는 크게 영상 처리 기법(Image Processing)과 딥러닝 기법(Deep Learning)으로 나눌 수 있다. 영상 처리 기법은 경계 추출(Edge Detection) 알고리즘을 통한 균열 영역 검출 방법<sup>[2]</sup>과 이미지 히스토그램과 이진화 알고리즘을 통한 균열/비균열 구분 방법<sup>[3]</sup>이 있다. 깊이 센서나 적외선 카메라를 통한 촬영으로 균열의 위치를 찾는 방법도 존재한다<sup>[4]</sup>. 영상 처리 기법은 데이터 기반의 학습 방식이 아니므로 학습 데이터에 따라 성능이 영향을 받지 않지만, 영상 내 환경 변화에 취약하고 성능을 일반화하기에 까다롭다는 단점이 있다. 딥러닝 기반의 균열 인식 기법은 데이터 일반화 및 최적화가 이루어지도록 학습하여 영상 처리 기법의 단점을 해결하고자 했다. 딥

러닝 기법의 개발로써 공인 균열 데이터셋 구축과 딥러닝 모델 개발 또한 활발하게 진행되어왔다. 그러나 공인 데이터셋의 수집 환경 도메인이 협소하여 실제 현장점검 시 적용성이 떨어지고, 균열 검출 라벨링에 대한 정의가 일관적이지 않아 검출에 대해 신뢰하기 어려웠다. 개발한 모델을 균열 점검 시스템에 실질적으로 활용하기 위해서는 상기한 문제들을 중점으로 해결하는 것이 중요하다.

본 연구는 균열 점검 시스템에 적합한 딥러닝 기반의 균열 검출 모델을 개발하는 것을 목표로 한다. 해당 목표를 구현하고자 1) 균열 공인 데이터셋의 한계를 보완하는 데이터셋 POC(Pohang Crack)를 구축하였고, 2) 점검 현장에서의 균열에 강인하게 검출할 수 있는 YOLO-Cr 모델을 개발하였고, 3) YOLO-Cr 모델을 기반으로 현장점검에 적용할 수 있는 시스템을 설계하였다. POC는 포항시에서 직접 촬영하여 수집한 데이터셋과 균열 검출 데이터셋인 AIHub<sup>[9]</sup> 데이터를 포괄하여 총 11,466장을 구축하였다. POC는 실제 점검 현장에서 나타날 수 있는 다양한 균열 종류를 촬영하여 현장 적용성과 협소한 도메인 문제, 데이터 품질 문제를 해결하고자 하였다. 객관적이고 체계적인 라벨링 규칙을 정의하고 생성한 레이블에 대해 여러 번 검수함으로써 신뢰할 수 있는 레이블 데이터를 생성하였다. YOLO-Cr은 YOLOv5와 YOLOv7의 각 세부 모델들을 비교 분석한 결과, mAP 91.5%로 가장 높은 성능을 보여준 YOLOv7-E6E 기반의 모델로 선정하였다. 기존 검출 모델들에서<sup>[8]</sup> 발견된 미검출, 오검출, 중복 검출 사례들이 획기적으로 줄어드는 점을 통해 YOLO-Cr의 뛰어나고 안정적인 검출 성능을 확인하였다. 균열 검출 시스템은 사람이 직접 균열 점검을 수행하는 경우와 무인 비행 시스템을 통해 균열을 확인하는 경우로 나누어 상황별로 모델을 활용하여 현장점검의 효율을 극대화하는 방안을 제시한다. 본 연구를 통해 균열 검사의 자동화가 적용되어 기존 점

검 방식에서의 큰 인력 소모의 문제와 검사의 객관성 부족 문제가 해결될 수 있길 기대한다.

## II. 본 론



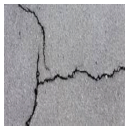

본론의 구성은 POC 데이터셋 구축, 균열 검출 모델 개발, 균열 점검 시스템 설계 순으로 진행된다. POC 데이터셋 구축은 공인 데이터셋에 대한 분석과 데이터셋 수집 체계를 설명하고, 균열 종류에 따른 분할 및 라벨링을 순서로 진행한다. 균열 검출 모델 개발은 선정한 인공지능 모델과 객체 검출 문제의 평가 지표를 소개하고 검출 모델별 성능과 데이터셋 구성에 따른 모델의 성능 차이를 분석하였다. 균열 점검 시스템은 두 가지 시나리오로 구별하여 균열 검출 모델의 예측 결과를 바탕으로 건물 안전도를 계산하고 활용하는 방안을 설계하였다. 실제 현장에서 관찰되는 균열의 다양한 요소를 고려하고, 균열 점검 시에 적용할 시스템을 설계했다는 점에서 이전 연구와 달리 실질적인 현장 활용 가능성에 초점을 맞췄다.

### 2.1 POC 데이터셋 구축

#### 2.1.1 공인 데이터셋 분석

딥러닝 기반의 균열에 관한 연구는 원본 이미지를 패치 단위로 잘라 균열/비균열 여부를 구분하는 방식<sup>15,6)</sup>(분류 모델), 픽셀 단위로 균열 여부를 판단하여 균열 영역을 확정하는 방식<sup>7)</sup>(구획화 모델), 이미지 내에 균열을 경계 박스(Bounding Box) 형태로 표출하는 방식<sup>8)</sup>(검출 모델)으로 나뉜다. 각 모델 개발을 위해 균열 관련 공개 데이터셋들이 존재함을 확인하였고, 인용 횟수나 적용 가능 범위에 따라 분석할만한 데이터셋은 크게 네 가지로 추려볼 수 있다(표 1). METU(Middle East Technical University)[5] 데이터셋은 균열, 비균열에 대한 분류 모델용 데이터셋으로써 균열과 비균열이 1:1로 균일하게 분포되어있고, 균열 자체가 명확하게 드러난다는 장점이 있다. 그러나 중복된 이미지의 존재, 콘크리트 균열 이미지 전체를 잘게 잘라 구성된 도메인 환경이 한정적이며, 매우 선명한 균열에 국한된다. SDNET2018<sup>6)</sup>은 METU와 같이 전체 이미지를 잘라 구성되어 있지만 잘린 이미지를 붙이면 원본 균열 이미지가 어떻게 구성되는지를 확인할 수 있고, METU보다 다양한 도메인을 추구했다는 장점이 있다. 그러나 도로 균열을 포함하고, 극히 미세한 균열들로 인해 균열 점검으로 활용하기에 적절한 데이터라고 보기 어렵다. CrackSeg9k<sup>7)</sup>은 균

표 1. 공개 데이터셋 요약  
Table 1. Open dataset summary

	(a) METU[5]	(b) SDNET2018[6]
Example		
Image size	227×227	256×256
Class	Crack, Non-Crack	Crack, Non-Crack
Domain	Building	Bridge, Walls, Pavements
Purpose	Classification	Classification
	(c) CrackSeg9k[7]	(d) AIHub[9]
Example		
Image size	400×400	500×332 ~ 2560×1440
Class	Crack, Non-Crack	Concrete Crack, Spalling, Efflorescence, Exposure, Steel Defect, Paint Damage
Domain	Walls, Building, Pavements	Dam, Bridge, Walls, Building, Plant
Purpose	Segmentation	Object Detection

열 구획화 모델 개발을 목적으로 구축된 10개의 데이터셋들을 모아 새롭게 구성된 데이터셋으로써 400×400 크기의 균열 이미지와 균열 영역을 추출한 이진화 마스크 데이터가 짝을 이뤄 구성된다. 앞선 두 데이터셋에 비해 환경이 다양하고, 균열의 전체적인 특징을 파악하기에 용이하다. 하지만 대부분 포장도로 균열에 집중되어 있어, 일반적인 건축물의 벽면 균열과는 다른 양상도 많았다. 국내 인공지능 학습용 데이터셋 허브인 AIHub 내 건물 균열 탐지 이미지 데이터셋<sup>9)</sup>은 드론을 활용하여 촬영한 건축물 결합 데이터셋으로, 콘크리트 균열, 바리, 백태/누수, 철근 노출, 강재 손상, 도장 손상 총 6개 종류와 다양한 크기의 데이터로 구성된 것이 특징이다. 그러나 이미지에 그림자가 포함되지 않도록 촬영하여 빛의 변화를 통제된 환경으로 구성되어 있었고, 이미지만으로는 균열이 전혀 분간되지 않는 경우들도 있었다.

균열 공인 데이터셋을 전반적으로 분석한 결과, 대부분의 데이터셋이 건물 벽면 발생 균열이 아니거나 한정적인 도메인에 적용된 점을 확인하였다. 이에 따라 본 연구에서는 높은 다양성과 범용성을 만족하는 데이터 촬영 기준을 선정하고, 선정된 기준에 따라 균열 이미지를 수집 후, 균열 종류별로 데이터 분포를 확인하였다. 분포를 토대로 각 균열 종류가 학습/검증/평가 데이터셋에 균일한 비율로 나뉘도록 하여 점검 현장에 특화된 데이터셋을 구축하였다. 추가로 균열 점검에 적합한 라벨링 방식을 정의하고 라벨링 작업을 거쳤다.

2.1.2 균열 데이터셋 수집

본 연구에서는 현장점검 환경에 가까운 균열 데이터셋 수집을 위해 포항 지역의 건물 벽면 균열을 촬영하였다. 포항시 내 지진 피해 건물과 노후 건축물이 위치한 곳을 대상으로 촬영 장소를 선정하였다. 표 2의 기준에 따른 데이터셋 수집 방법을 통해 총 4,447장의 데이터를 수집 및 정제하였다. 1,2) 균열의 형태가 분명하게 식별 가능한 날씨와 시간대에 촬영을 진행하였고, 3) 촬영 기기는 스마트폰에 탑재된 기본 카메라 사양을 활용하였다. 4,5) 촬영 시에는 바닥과 수직으로 카메라를 들고 흔들림 없이, 균열에 초점을 맞춰 촬영하였다. 6) 촬영한 결함의 종류는 콘크리트나 벽돌 균열로, 7) 빌라, 아파트 등의 건축물 외벽을 촬영한다. 8) 촬영 시 차량 번호, 사람의 얼굴 등 민감 정보는 식별 불가능하도록 하였다. 9) 공인 데이터셋에서는 같은 균열에 대해 촬영 위치를 근소한 차이로 두어 촬영하거나 원본 이미지를 패치로 나눴기 때문에 데이터 간 유사도가 90% 이상이지만, 본 촬영 데이터셋에서는 이미지 데이터 간 유사도를 50% 미만으로 촬영을 진행하여 이미지 구성이 최대한 다양하게 하였다. 10) 데이터는 모두 4000 × 3000 이상의 고해상도로, 데이터 형식은 JPG로 통일하였다. 11) 실제 건물 벽면에는 그림자, 균열 유사 패턴, 전선 등 다양한 장애물이 존재하는데, 이처럼 넓은 데이터 도메인에서 이미지를 수집하여 실제 균열 검출 모듈로써 활용하였을 때 부족함이 없도록 설정하였다.

실제 현장점검을 할 때 쉽게 접할 수 있지만, 현재 이에 유연하게 대처할 수 있는 검출 모델이 전무한 점을 고려하여 데이터를 수집하였다. 다른 공인 데이터셋에 비해 촬영 환경에 있어 특이점을 갖는 데이터의 유형은 크게 네 가지로, 그림 1에서 대표적인 예시 이미지를 확인할 수 있다. 각 유형은 (a) 벽면의 디자인이 균열과 유사한 유형, (b) 벽돌 간 틈새가 균열과 유

표 2. 데이터셋 수집 방법  
Table 2. Dataset acquisition methods

No.	Item	Details
1	Time zone	Before sunset
2	Weather	Sunny Recommended(No rain/snow)
3	Device type	Smartphone basic camera
4	Method	Close to the crack and hold the camera perpendicular to the floor
5	Data quality	Focused on crack and shoot without shaking
6	Defect type	Concrete wall crack
7	Facility type	Exterior walls of buildings (apartments, villas, and fences)
8	Sensitive information	Unidentifiable car plate, human face, location
9	Similarity	Less than 50% between images
10	File format	JPG, 4000 × 3000 ≤ size
11	Domain	Shoot to include various environments (shadows, crack-like patterns, electric wires, objects other than walls, etc.)

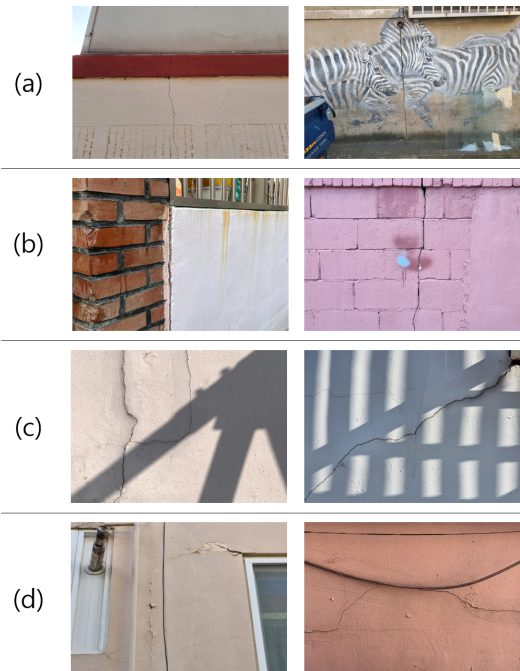


그림 1. 포항시 균열 데이터 예시  
Fig. 1. Example of crack data in Pohang

사한 유형, (c) 주변 건물이나 철망과 같은 구조물에 의한 짙은 그림자가 있는 유형, (d) 전선이나 식물 등

이 균열과 유사하게 존재하는 유형으로 나뉜다.

균열 검출 모델이 균열의 형태와 특성에 대해 충분히 학습할 수 있도록 직접 수집한 데이터셋과 더불어 AIHub 건물 균열 탐지 이미지 데이터셋도 추가로 활용하였다. 상기한 데이터셋 수집 방법과 기준에 부합하는 데이터로 한정하였고, 콘크리트 균열에 해당하는 데이터 중에서 7,019장을 선정하였다. 촬영을 통해 본 연구에서 수집한 데이터 4,447장과 AIHub 데이터 7,019장을 합하여 총 11,466장의 데이터셋을 구축하였다.

2.1.3 데이터셋 분할 및 라벨링

본 연구에서는 다양한 종류의 균열, 작고 가는 균열 등에 강인한 검출 모델 학습을 위해 균열의 형태와 특성에 따라 세부적인 분류 기준을 제안하고, 이를 바탕으로 체계적인 데이터셋 분할작업을 수행하였다. 균열 데이터셋의 하위 집단은 균열 종류 분류 모델 개발에서 흔히 사용하는 분류<sup>10)</sup>를 포함하여 다음의 10개 항목으로 나누었다(표 3): 1) 일반형(Normal), 2) 가지형(Branch), 3) 종방향(Longitudinal), 4) 횡방향(Transverse), 5) 거북등(Alligator) 6) 분산형(Scatter), 7) 박락(Spalling), 8) 단층(Fault), 9) 유사 균열(Similar), 10) 기타(장애물, 보수 흔적 등).

공인 데이터셋의 경우, METU와 SDNET2018은 균열이 횡/종방향의 여부만 파악되고 그 외의 균열 종류는 파악할 수 없는데 이는 원본 이미지를 작은 패치 형태로 잘라서 데이터셋을 구성하였기 때문이다. 또한, 데이터가 모두 유사한 형태이므로 데이터 구성이 단순하다. CrackSeg9k, AIHub는 분류항목 1~4와 같

은 현장에 다수 발생하는 유형의 균열을 포함하고 있었다. 하지만 5~7 유형의 데이터 비율은 훨씬 낮고, 8~10 유형은 전혀 없었는데, 벽의 종류와 촬영 환경의 범위가 넓지 않은 것을 반증한다. POC는 그림 2에서 확인할 수 있듯이 균열 검사 현장에서 발견 가능한 여러 종류의 균열을 포함하여 높은 데이터 다양성을 추구하였다. 특히 오검출을 유도할 수 있는 단층형, 유사 균열형, 기타 장애물을 포함하는 데이터를 실제 점검 시에 나타나는 비중에 맞게 확보함으로써 검출 모델의 변별력을 높이고자 했다. 수집한 데이터는 정기 안전 점검<sup>11)</sup>의 대상이 되는 3종 시설물 및 지진 피해 건축물을 방문하여 수집한 것으로 실제 시설물 점검이 수행되는 환경과 동일하다.

구축한 데이터셋은 학습/검증/평가 데이터셋으로 분할하였고, 본 연구에서는 모든 균열 유형에 대해 검출 성능이 뛰어난 모델을 개발하기 위해 층화 표집(Stratified sampling)을 활용하여 데이터셋 분할을 진행하였다. 분할 기법으로 가장 많이 사용되는 단순 무선 표집(Simple random sampling)은 모집단의 모든 데이터가 표본으로 추출될 확률이 동일하나, 층화 표집은 모집단을 속성에 따라 구분한 각 계층에서 단순 무선 표집을 수행함으로써 각 계층의 데이터를 모두 포함할 수 있도록 한다<sup>12)</sup>. 이는 단순 무선 표집이 계층별 데이터를 고루 포함하지 못하는 한계를 보완하며, 균열은 형태와 촬영 환경에 따라 10개 계층으로 분류될 수 있으므로 각 계층에 대한 공정한 표집을 수

표 3. 데이터셋 계층  
Table 3. Dataset subgroups

No.	Subgroup	Description
1	Normal	Various cracks of common shape
2	Branch	Small cracks attached to large crack
3	Longitudinal	Vertically split crack
4	Transverse	Horizontally split crack
5	Alligator	The whole wall is cracked
6	Scatter	Cracks are disconnected, but the direction of crack is connected
7	Spalling	Cracks containing parts that have peeled off the surface of the wall
8	Fault	Cracks in the layered wall
9	Similar	Crack-like wires, door gaps, etc
10	etc	Obstacles obscuring crack

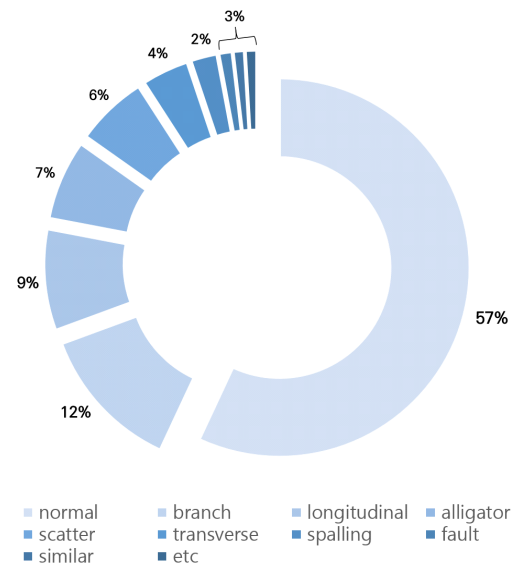


그림 2. 데이터셋 계층별 분포  
Fig. 2. Distribution by dataset subgroup

행하는 방식을 채택하였다. 구축한 POC 데이터셋 총 11,466장은 학습 7,647장, 검증 1,910장, 평가 1,909장으로 약 4:1:1의 비율로 분할하였다.

POC 데이터셋은 다양한 환경에서 발생한 균열을 포함하고 있는 만큼, 균열의 형태를 분명히 파악하여 검출할 수 있도록 정확하고 신뢰도 높은 라벨링이 필수적이다. 이는 라벨링에 따라 모델이 학습하는 균열의 특징, 크기, 개수 등이 크게 영향을 받기 때문이다. 본 연구에서는 선행 연구<sup>8)</sup>에서 선정한 라벨링 규칙을 기반으로 라벨링을 수행하되, 추가로 확인한 부분에 대해 가이드를 보완하여 작성함으로써 작업자의 판단에 있어 발생할 수 있는 모호함 및 레이블의 비일관성을 최소화하였다. 기존 가이드에는 균열 외 결함에 대한 설명이나 기준이 부재하였으나, 그림 3(a), 그림 3(b)와 같이 균열은 흔히 박리/박락과 같은 기타 결함과 공존하는 비율이 높다. 기타 결함이 균열과 겹쳐 발생하거나 유사한 형태로 나타나는 경우 작업자마다 균열 영역에 대한 인식이 달랐기 때문에 목표로 하는 결함인 콘크리트 균열 외 박리/박락, 누수, 철근 노출 등의 결함은 포함되지 않도록 규칙을 작성하였다. 건축물의 벽면에 연결되어 있지 않은 두 개 이상의 균열이 존재할 때는 각 균열의 가까운 정도와 갈라진 방향에 따라 레이블 박스를 합칠지, 분리할지 방식을 달리 한다. 서로 다른 각 균열이 직접적으로 연결되어 있지 않더라도 간격이 매우 가깝고 진행 방향이 유사한 경우 벽면 내부적으로 연결되었다고 판단하여 하나의 균열로 간주한다(그림 3(c)). 또한, 균열과 비슷하게 보이지만 건축 구조상 발생한 틈이나 단층, 그림자 등의 경우에는 그림 3(d)와 같이 레이블을 부여하지 않

는 것을 원칙으로 한다. 가이드에 맞게 생성한 레이블 데이터에 대해서는 라벨링 작업자 간 교차 검증을 통해 세 차례 검수를 진행함으로써 작업자의 실수나 일관성이 떨어지는 문제를 방지하였다. 딥러닝 모델 학습을 위한 레이블 데이터는 라벨링 프로그램인 LabelImg를 통해 생성하였으며, 이미지 내 균열은 직사각형의 박스 형태로 표시하였다.

## 2.2 YOLO-Cr 모델 개발

### 2.2.1 균열 검출 모델 후보군 선정

객체 검출 모델은 문제 해결 방법론에 따라 크게 One-Stage Detector와 Two-Stage Detector 두 가지로 분류할 수 있다. Two-Stage Detector는 물체가 있을 만한 영역을 찾아내는 과정과 객체를 식별하는 과정을 순차적으로 수행하고, One-Stage Detector는 두 과정을 동시에 수행한다는 점에서 차이가 있다. 본 연구에서는 균열 검출에 있어 One-Stage Detector인 YOLO(You Only Look Once)가 Two-Stage Detector(Feature Pyramid Network, Faster R-CNN) 대비 높은 성능을 보인 기존 선행 연구<sup>8,12)</sup>를 토대로 YOLO 계열의 모델을 중심으로 모델 후보군을 선정하였다. YOLO는 합성곱 신경망(Convolutional Neural Network)을 기반으로 구현된 객체 검출 모델로, 높은 검출 성능뿐만 아니라 빠른 수행 속도를 바탕으로 객체를 실시간으로 검출할 수 있는 장점이 있다. YOLO 모델은 v1<sup>[13]</sup>부터 v7<sup>[14]</sup>까지 성능을 개선하여 발표하고 있으며, 가장 최신 모델인 YOLOv7과 지속적인 업데이트와 안정성이 특징인 YOLOv5<sup>[15]</sup> 모델을 집중적으로 비교하였다.

기본적으로 두 네트워크는 모두 YOLOv4<sup>[16]</sup>의 구조를 기반으로 하여 Backbone, Neck, Head로 구성된다. Backbone은 입력 이미지의 특징을 추출하는 역할, Neck은 추출된 특징들을 결합하여 Head로 전달하는 역할, Head는 Bounding Box의 좌표와 클래스를 예측하는 역할이다.

YOLOv5의 Backbone 네트워크는 CBS(Conv layer + Batch Normalization + SiLU) 모듈과 C3 모듈로 구성되어 있고, C3 모듈(그림 4)은 연산량을 크게 줄이면서도 뛰어난 특징 추출 기능을 가지는 Cross Stage Partial Network(CSPNet)<sup>[17]</sup> 기법에 기반하였다. Neck에서는 Path Aggregation Network(PAN) 구조를 통해 저수준 및 고수준의 정보를 결합함으로써 다양한 크기와 비율로 존재하는 객체의 검출을 돕는다. 모델의 Head는 YOLOv3<sup>[18]</sup>와 유사하지만 보다

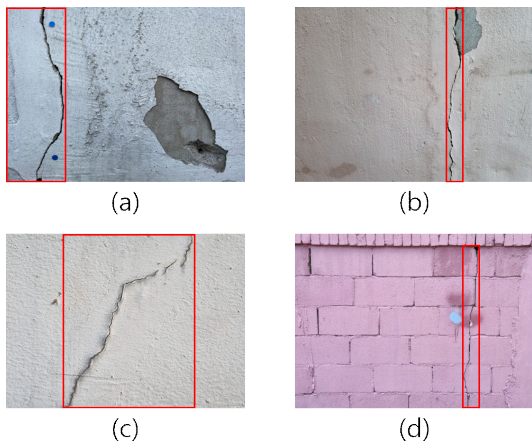


그림 3. 데이터 라벨링 예시  
Fig. 3. Examples of data labeling

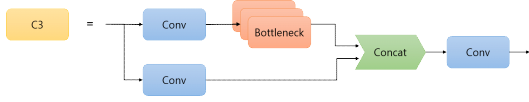


그림 4. C3 구조[15]  
Fig. 4. C3 structure[15]

많은 Anchor Box를 사용하도록 구현했다. YOLOv5는 모델의 크기 순서대로 S, N, M, L, X 구조로 나뉘고, 각 구조는 업데이트를 통해 S6, N6, M6, L6, X6로 개선되었다. 본 연구에서는 균열 검출 성능에 집중하여 모델의 복잡도가 있지만 높은 성능인 L6와 X6 모델을 채택하였다.

YOLOv7의 backbone 네트워크는 CBS 모듈, ELAN(Efficient Layer Aggregation Networks) 모듈, E-ELAN(extended-ELAN) 모듈로 구성된다(그림 5). E-ELAN은 C3 모듈과 같이 CSPNet의 아이디어와 Grouped Convolution<sup>[19]</sup>을 사용해 효율적인 연산이 가능하고 그룹마다 독립적인 특징을 학습할 수 있는 효과를 얻는다. 또한, Concatenation-based Model Scaling<sup>[14]</sup>을 사용하여 연산 과정 중에 모델의 축소/확대 시 늘어나는 차원을 다루기가 어려워지는 문제를 해결하였다. Neck에서는 YOLOv5와 같은 PAN 구조를 사용하며, head에서는 YOLOR<sup>[20]</sup>의 Head 구조를 따른다. YOLOv7은 적용된 기법과 사용하는 모듈의 종류에 따라 기본모델, X, W6, E6, D6, E6E로 나뉘며, 본 연구에서는 상기한 이유로 D6, E6E 모델을 채택하였다.

채택한 네트워크 구조 중 균열을 검출해내는 성능이 가장 뛰어난 네트워크 구조를 찾자 하였으며, 이는 YOLO-Cr(Crack)이라고 칭한다.

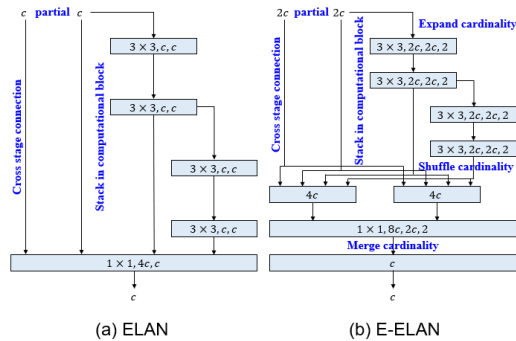


그림 5. ELAN, E-ELAN 구조[14]  
Fig. 5. ELAN, E-ELAN structure[14]

### 2.2.2 균열 검출 모델 성능 지표

본 논문에서는 균열 검출의 성능 지표로 Mean Average Precision(mAP)을 사용하였다. mAP를 산출하기 위해서는 정밀도(Precision, (1))와 재현율(Recall, (2))의 값이 필요하며, 정밀도는 모델이 옳다고 검출한 것 중 실제로 정답인 것의 비율, 재현율은 실제 정답인 것 중 모델이 옳다고 검출한 것의 비율이다.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

모델이 맞게 검출하였는지에 대한 여부는 정답 박스(Ground Truth)와 예측 박스(Prediction) 간 겹치는 정도인 Intersection over Union(IoU)을 기반으로 판단한다(그림 6). IoU에 대한 임계값을 0.5로 가정했을 때, 예측 박스와 정답 박스 간 IoU 값이 0.5 이상이면 해당 예측은 옳은 검출이라고 판단하며(TP, True Positive), 예측 박스는 있지만 어떤 정답 박스와도 IoU 값이 0.5를 넘기지 못한다면 오검출인 FP(False Positive)로 판단한다. 정답 박스를 찾지 못했을 때는 미검출이므로 TN(True Negative)이라고 판단하며, FN(False Negative)은 정답 박스도 없고, 예측 박스도 없는 경우이기 때문에 고려하지 않는다.

IoU의 임계값에 따라 정밀도와 재현율의 값이 달라지는데, Precision-Recall 곡선(PR 곡선)은 x축을 재현율, y축을 정밀도로 하여 그래프로 그린 것이며, PR 곡선 아래 면적을 구해 Average Precision(AP) 값을 얻을 수 있다. 모델의 성능을 판단하는 데 있어 정밀도와 재현율을 별개로 하여 고려하기보다 두 값을 종합적으로 평가할 수 있어 적합한 성능 지표라고 할 수 있다. 마지막으로, 검출하고자 하는 클래스마다 AP를 구하고 이들의 평균을 구하여 mAP 값을 산출한다.

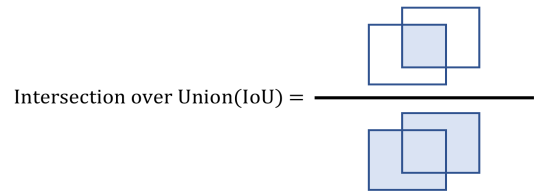


그림 6. 영역 간 겹침  
Fig. 6. Intersection over Union

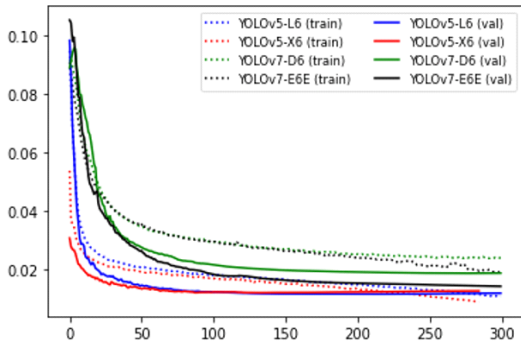
2.2.3 모델별 검출 성능 비교 분석 및 선정

POC를 기준으로 검출 모델의 성능을 비교하고자 YOLOv5-L6, YOLOv5-X6, YOLOv7-D6, YOLOv7-E6E 네 개 모델에 대한 학습을 진행하였다. 학습 최적화를 위해 각 모델의 일부 초매개변수들을 다르게 설정하였으나, 입력 이미지의 크기, 학습 횟수, 증강 기법, 성능 지표 등은 모두 동일하게 적용하였다. 학습 시에는 COCO 데이터셋으로 사전 학습된 가중치 행렬(Weight)를 활용하여 전이 학습(Transfer Learning)하였고, 입력 이미지는 모두 640 × 640 크기로 조정 후, 좌우 반전, 색 공간 값 변화, Mosaic, Mixup 등의 데이터 증강 기법을 적용하였다. 모델 학습을 위한 딥러닝 프레임워크로 Pytorch를 사용하였고, 학습한 각 모델의 학습 추이는 x축을 학습 횟수(Epoch)로 하여 그림 7과 같다. 학습 횟수는 학습용 데이터가 전부 한 번씩 모델을 통과한 횟수를 의미하며, 학습을 여러 번 반복하여 가중치를 더 나은 방향으로 수정하였다. 학습 과정에서 검증 데이터에 대한 mAP 성능이 개선되지 않는다면 학습을 조기 종료하

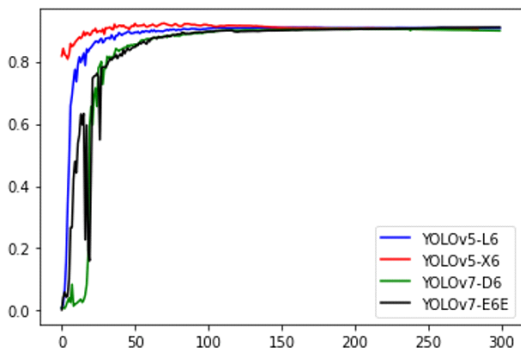
도록 구현하여 과적합을 방지하였다.

학습이 진행됨에 따라 YOLOv5-L6, YOLOv7-D6 두 개 모델의 검증 데이터의 손실 값과 mAP 값이 수렴하는 것을 통해 모델이 최적화되었다고 판단할 수 있다. YOLOv5-X6 모델은 학습 데이터의 손실이 꾸준히 감소하지만, 검증 데이터의 손실은 증가하고 mAP가 감소하므로 과적합 모델로 판단하여 300회의 학습 이전에 조기 종료되었다. YOLOv7-E6E 모델은 학습/검증 데이터셋에 대한 손실 값은 지속적으로 감소하고 mAP 값은 수렴하지 않아 학습을 추가 진행하였다. 모델마다 수렴이 되는 학습 횟수의 기준은 달랐으나, 데이터에 대해 충분히 학습된 것을 확인할 수 있다.

평가용 데이터셋에 대한 각 모델의 정밀도, 재현율, mAP 성능은 표 4와 같다. 전반적인 검출 모델의 성능 차이가 큰 폭으로 나타나진 않았다. 이는 YOLOv5와 YOLOv7은 네트워크의 뼈대 구조가 기본적으로 유사하고 YOLOv5 자체도 이미 다양한 검출들에 강인한 검증된 모델이기 때문이다. 특히 YOLOv5-X6는 훨씬 작은 Epoch에서도 YOLOv7-E6E와 유사한 mAP 성능을 보여주었다는 점이 고무적이다. YOLOv7-D6 모델은 가장 낮은 성능을 보여주었는데, 이는 YOLOv7에서 새롭게 제안한 핵심 모듈인 E-ELAN 모듈을 미사용해서 데이터를 학습하는 능력에 있어 YOLOv7-E6E 모델과 차이가 발생한 것으로 추정된다. mAP를 기준으로 평가할 때 YOLOv7-E6E가 가장 높고, YOLOv5-X6, YOLOv5-L6, YOLOv7-D6 순으로 그 뒤를 이었다. 특별히 YOLOv7-E6E 모델은 다른 모델에 비해 학습 속도가 느리고 모델의 성능의 안정화를 위해 더 많은 학습 횟수가 필요했지만, 결과적으로 가장 높은 정확도를 보였다. 공통적으로 300 Epoch을 학습한 모델보다 600 Epoch을 학습했을 때 개선된 점을 통해 확인할 수 있다. 모델 내 네트워크의 복잡도가 높지만, 그만큼 오검출의 위험이 있는 다



(a) Train / Validation Box Loss



(b) Validation mAP

그림 7. 모델별 학습 추이  
Fig. 7. Learning progress for each model

표 4. 딥러닝 모델 성능 평가

Table 4. Deep learning model performance evaluation

	Precision	Recall	mAP.5
YOLOv5-L6	0.88	0.86	0.901
YOLOv5-X6	0.891	<b>0.869</b>	0.913
YOLOv7-D6	0.86	0.843	0.892
YOLOv7-E6E	0.891	0.84	0.906
<b>YOLO-Cr</b> (YOLOv7-E6E, 600)	<b>0.914</b>	0.84	<b>0.915</b>



양한 경우들을 학습하고, 표상할 수 있었던 것으로 추측한다. mAP값과 재현율은 다른 모델들과 근소한 차이를 보이지만, 정밀도에서 압도적으로 높은 점을 통해 확인할 수 있다. YOLOv7-E6E에서의 핵심 모듈이었던 E-ELAN 모듈이 기존에는 잡지 못할 까다로운 검출 내용들에 대해 강인한 점을 확인하였으므로, 성능이 가장 뛰어난 YOLOv7-E6E 모델을 YOLO-Cr로 선정하였다. YOLO-Cr의 구체적인 학습 초매개변수(Hyper-parameter)는 기본적으로 설정되어있는 값에서 다양하게 변경하며 성능을 비교해보았고, 최종적으로 초기 학습률(Initial learning rate) 0.01, 모멘텀(momentum) 0.937, 가중치 감쇠율(weight decay) 0.0005를 적용하였다.

YOLO-Cr의 성능에 대해 복합적으로 평가할 수 있도록 검출 박스를 정성적으로도 평가하였다. YOLO-Cr 모델을 활용한 균열 검출 예시를 그림 8에 나타냈으며, 사진 내 파란 선은 정답 경계 박스, 빨간 선은 모델이 검출한 경계 박스다. 예측 박스와 정답 박스를 직접 비교한 결과, 주변 배경이 함께 촬영됨에도 균열을 정확하게 검출하고, 전선이나 차량, 그림자 등 장애물이 분포하는 중에도 유연하게 검출하였다. 균열로 혼동될 수 있는 경우에도 잘 구별해낸 것은 다양한 도메인의 데이터들이 POC 데이터셋 내 특이 데이터 확보와 강화 표집 방식을 통해 학습되어 반영된 것으로 분석된다. 평가 데이터셋의 오류 샘플들에 대해서도 사례별로 통계를 도출하는 방식으로 구체적인 정성 평가를 수행했다. 1909개의 평가 데이터셋 중 잘못된 검출 사례는 총 217개로, 오류 유형은 오검출(False Detection), 미검출(Missed Detection), 중복 검출(Duplicated Detection), 넓은 검출(Wide Detection)

표 5. YOLO-Cr 모델 정성 평가  
Table 5. YOLO-Cr model qualitative evaluation

Error-case	Number	Example
False Detection	35 (1.8%)	
Missed Detection	102 (5.3%)	
Duplicated Detection	57 (3.0%)	
Wide Detection	23 (1.2%)	

로 나누었다. 각 사례의 개수와 발생 비율, 예시 이미지는 표 5와 같다.

오검출은 누수로 인한 곰팡이, 박리로 손실된 표면, 흘러내린 페인트 자국, 보수 흔적, 매우 짙은 그림자, 담쟁이 종류의 얇은 식물, 스크래치 등에 대해서 균열

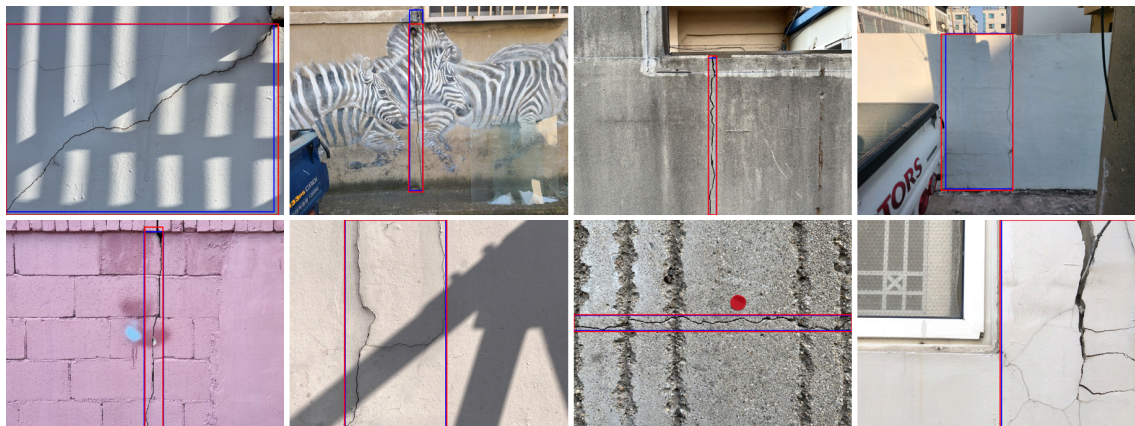


그림 8. 균열 검출 예시, 파랑: 기준 라벨링, 빨강: 모델 검출 결과  
Fig. 8. Crack detection example, Blue: Ground Truth, Red: Prediction

로 검출한 경우들이 있었다. 그러나 상기한 경우가 포함 된 사진은 평가 데이터셋 중 약 24% 정도였는데, 해당 경우의 데이터에 대해서 8%만 오검출이 나타났다. 직접 벽면을 확인하지 않는 이상 균열 구분이 어려운 데이터를 고려하면 모델의 고도화로 개선될 수 있는 검출은 35건 중 약 19건으로 집계하였다. 유사 사례의 데이터를 늘려 추가 학습하면 모델의 오검출 사례가 감소할 것으로 보인다. 미검출은 전체 평가 데이터셋에서 5.3%로 가장 큰 비율을 차지하였는데, 대다수 끈은 직선 형태의 가로 혹은 세로 균열이었다. 건물의 구조상 틈이나 경계면을 다수 포함하고 있으므로 균열임에도 끈은 형태의 틈으로 인식해 미검출한 것으로 확인된다. 또한, 두께 차이가 극심한 상황에서 얇은 균열을 스크래치로 판단해 미검출한 경우, 단층 유형에서 서로 다른 층의 균열 중 상대적으로 크기가 작은 균열을 미검출하는 경우들이 있었다. 미검출된 균열 중 위험한 수준의 균열이 있는지 파악하고자 위험 수준을 표 6과 같이 세 개 단계로 나누었고<sup>[21]</sup>, 미검출 사례 102개에 대해 분석한 결과 심각 단계의 균열은 전체의 3.9%로 작은 부분을 차지하였다. 중복 검출은 주로 큰 균열의 곁에 작은 균열이 있을 때, 큰 균열만 검출한 박스, 작은 균열도 포함해 검출한 박스를 모두 생성하여 하나의 균열에 두 개의 박스를 생성하는 경우가 존재하였다. 넓은 검출은 균열의 근처가 빈 벽면임에도 박스를 확장 검출한 경우로, 총 23건 발생하였다. 이는 학습을 통해 모델이 최적화한 경계 박스의 크기 및 비율이 존재하는 모든 균열에 대해 표상할 수 없으므로 나타난 문제로 보인다. 모델이 균열을 적합한 비율과 크기의 박스로 검출하기 위해 더 많은 종류의 Anchor box를 적용한다면 이러한 사례들은 개선될 수 있다.

정상 검출 사례 및 오류 사례에 대한 정성 평가를 통해 모델의 검출 사례를 확인한 결과, 복잡한 환경에서도 균열을 검출하는 능력이 뛰어나고 여백이 없도록 경계 박스를 그려 균열 영역을 분명히 확인할 수

있었다. 그러나 균열과 유사한 객체들과 위험도가 매우 낮은 극미세 균열 등이 함께 나타나고 있어 검출에 어려운 부분이 있었고, 모든 종류의 균열에 대해 강인하게 검출하기에는 학습 데이터 중 부족한 일부 데이터 계층이 존재했다. 미검출, 중복 검출의 사례가 존재했으나, 위험한 균열에 대해서는 명확히 검출하였으므로 상기한 정성 평가 분석에 따르면 본 모델의 활용성에 대해 긍정적으로 판단할 수 있다.

### 2.3 학습 데이터셋 구성에 따른 균열 검출 성능 비교

POC 데이터셋을 점검 시스템에 활용함에 있어 실효성을 확인하고자 두 가지 실험을 진행하였다. 첫 번째는 YOLOv7-E6E를 기반으로 AIHub 데이터셋만 학습한 모델과 POC 데이터셋을 학습한 모델에 대한 검출 성능을 비교하였다. 표 7을 통해 각 모델의 정밀도, 재현율, mAP값을 도출하였다. POC를 학습했을 때 AIHub로만 학습한 결과보다 mAP 성능이 5.5% 높게 나온 것을 확인할 수 있다. 그림 9와 같이 각 모델의 검출 결과를 살펴본 결과, 두 모델 모두 일반적인 균열에 대해서는 잘 검출하였지만, 그림자, 전선 같은 유사 균열 사례에서는 AIHub 데이터로만 학습한 모델이 오검출하였다. 이는 AIHub 데이터에는 없는 데이터로써 관련한 학습이 반영되지 않아 균열로 오인식한 문제로 분석되며, POC를 통해 해당 문제에 강인하게 학습된 것을 확인하였다.

두 번째 실험에서는 POC 데이터셋과 기존 데이터셋(METU, SDNET2018, CrackSeg9k)의 유사성과 차이를 검출 모델을 통해 간접적으로 확인하였다. 이를 위해 기존 공인 데이터셋을 학습한 모델과 POC를 학습한 모델에 대해 검출 성능을 비교하였다. 상기한 라벨링 규칙에 따라 세 데이터셋의 레이블 데이터를 검수하였고, 각 모델의 POC 평가 데이터셋에 대한 검출 성능과 YOLO-Cr 모델의 각 데이터셋에 대한 검출 성능은 표 8과 같다. METU와 SDNET2018을 학습한 모델은 POC에 대한 mAP가 각각 0.396, 0.201로, 현장에서 촬영한 크기가 크고 복잡한 균열을 검출하기에 역부족임을 확인하였다. 이는 두 모델 모두 균열

표 6. 미검출 균열의 위험 수준 분석  
Table 6. Danger level analysis for missed crack

Level	Definition	Number
Minor	Very small or thin crack	60 (58.8%)
Moderate	Larger than minor crack and not much severe crack	38 (37.3%)
Severe	Very large and dangerous crack and hence immediate remedial measures are required	4 (3.9%)

표 7. 학습 데이터셋에 따른 모델 성능 차이  
Table 7. Differences in model performance by train dataset

	Precision	Recall	mAP.5
AIHub	0.855	0.792	0.860
POC	0.914	0.840	<b>0.915</b>

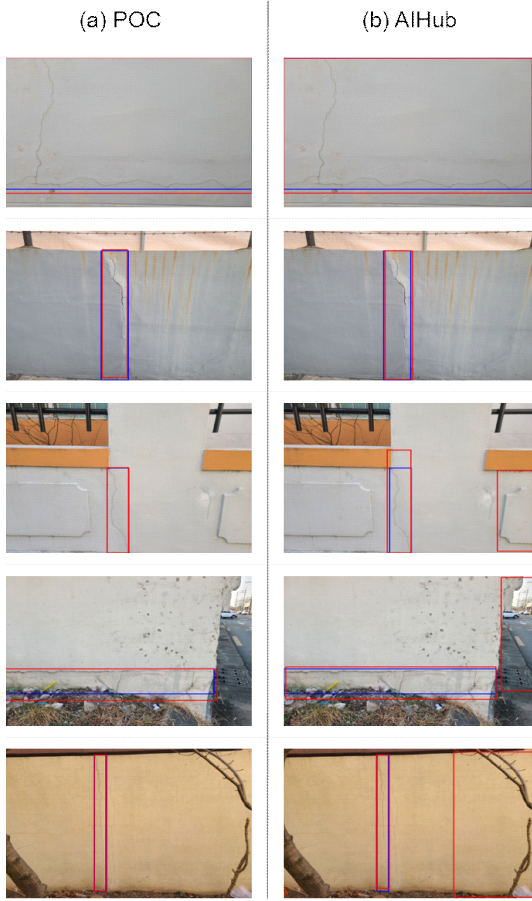


그림 9. 균열 검출 비교  
Fig. 9. Crack detection comparison

이미지를 잘게 분할한 데이터셋을 기반으로 학습하였기 때문에 균열 자체가 이미지에서 차지하는 비율이 달라 생긴 문제로 본다. 특별히 검출된 결과를 확인해 본 결과 균열에 근접하게 촬영하여 두꺼운 형태의 균열에 대해서만 검출이 된 점으로 확인할 수 있다. CrackSeg9k 데이터셋으로 학습한 모델은 mAP가 0.76이었는데, 더욱 다양한 환경에서 데이터 수집이 이루어졌기 때문에 균열의 형태나 맥락을 잘 학습한 것으로 판단되었다.

YOLO-Cr 모델로 METU, SDNET, CrackSeg9k 평가용 데이터셋에 대해 평가한 결과, YOLO-Cr 모델은 도메인이 전혀 다른 데이터셋에 대해서도 준수한 검출 성능을 보여주었다. SDNET2018 데이터셋의 검출 성능이 가장 낮았는데, 정밀도와 재현율 값의 차이도 가장 큰 것을 확인할 수 있다. 이는 SDNET2018 데이터가 유독 균열이 얇고 깊이가 얕은 데이터가 많아 미검출한 것으로 확인했다.

표 8. 모델 성능 교차 검증  
Table 8. Cross-validation of model performance

Train data	Test data	Precision	Recall	mAP.5
METU	POC	0.518	0.427	0.396
CrackSeg9k		0.760	0.740	0.760
SDNET2018		0.344	0.256	0.201
POC	METU	0.903	0.638	0.768
	CrackSeg9k	0.930	0.700	0.809
	SDNET2018	0.821	0.503	0.615

두 실험을 통해 도메인에 따라 mAP 값이 0.4 이상 차이가 발생할 만큼 균열 데이터의 도메인이 넓은 것을 확인하였으며, 넓은 도메인 간극을 POC와 YOLO-Cr을 기반으로 최소화할 수 있었다. 특히, POC 데이터셋이 복잡한 균열 점검 상황에 대처할 수 있는 인공지능 개발에 있어 최적화된 데이터셋임을 검증하였다.

#### 2.4 현장 균열 점검 시스템 설계

기존 현장 균열 점검에서는 점검자가 육안으로 균열을 확인하고, 확인된 균열에 따라 균열의 크기, 종류, 두께, 위치 등에 대한 정보를 균열 자나 센서 등을 통해 심층적으로 측정한다. 이때 현장점검에서의 육안 검사는 건물의 규모가 커 균열을 다 파악하지 못하고 지나칠 수 있고, 균열을 인지하더라도 접근할 수 없는 위치라 측정이 불가능한 문제가 발생할 수 있다. 상기한 두 가지 문제에 대해 YOLO-Cr을 활용하여 해결할 수 있는 방법을 소개하고자 하며, 이에 대한 개요도는 그림 10과 같다. 본 시스템에서는 균열에 대한 발견과 균열 개수, 크기, 위치를 알릴 수 있는 기능을

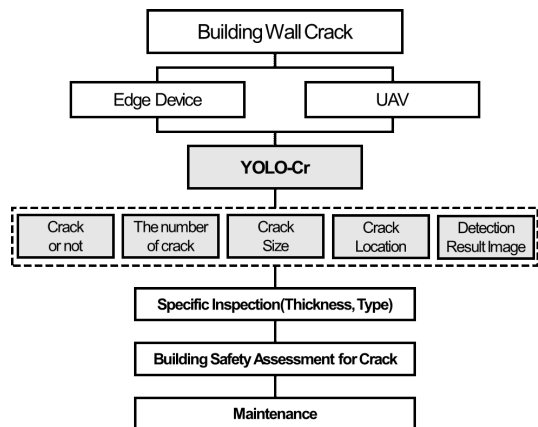


그림 10. 균열 점검 시스템  
Fig. 10. Crack inspection system

통해 심층적인 측정 이전의 균열 인식에 초점을 둔다.

자동화된 균열 점검 방식으로 두 가지 시나리오로 나누어 고려하였으며, 먼저 동영상을 기반으로 한 스케닝이다. 이는 휴대전화의 애플리케이션과 같이 에지 디바이스(Edge Device)에서 검출 모델이 구동할 수 있다는 전제로 한다. 영상 내 프레임마다 균열 검출을 수행하며, 연속적인 프레임 내에서 중복적으로 검출이 되는 경우 균열로 간주한다. 균열이 검출되면 촬영 당시의 사진, 검출 결과(Bounding Box), 균열의 개수 등을 저장하여 점검자가 쉽게 확인할 수 있도록 한다. 점검자는 모든 현장 사진에 대해 직접 확인할 필요 없이, 점검이 필요한 사진에 대해 실제로 균열인지 확인하고 그에 따라 점검자는 균열의 종류나 심각성을 판단하여 심층 작업으로 연계하면 된다. 본 개발 모델은 이미 벽면과 가깝게 촬영한 데이터뿐만 아니라 멀리서 촬영한 데이터에 대해서도 다루지기 때문에 점검 초반에 건물을 스캔하면서 작업자가 놓칠 수 있는 균열에 대해 신호를 줄 수 있을 것으로 기대한다.

두 번째는 드론과 같은 무인 항공기(Unmanned Aerial Vehicle, UAV) 등을 활용하여 결함을 검출하는 방법이다. 본 검출 모델을 활용하면 높은 건물이나 구조물 등 사람이 접근하기 까다로운 위치에 있는 균열 상태를 점검하고 안전도를 평가하는 시스템에 활용할 수 있다. 무인 항공기를 활용해 건축물의 외벽 이미지를 촬영하고 균열 검출 모델에 입력하면, 해당 모델은 균열의 위치 좌표, 벽면 내 균열의 개수, 균열의 크기 정보, 균열이 포함된 사진을 포출한다. 위와 마찬가지로 자동으로 벽면에 대해 균열을 검출하고 위험 정보를 전달하는 방식을 토대로 점검자는 균열을 심층 분석하여 위험도를 판단할 수 있을 것이다.

추가로 주기적으로 균열의 진행성을 파악하기 위해 본 검출 모델을 활용할 수 있다. 동일한 위치에서 촬영되었음을 가정할 때, 경계 박스의 크기 변화를 관찰하여 진행률을 파악하고 이에 따른 균열의 위험성을 판단할 수 있다. 최종적으로 점검자가 파악한 균열 관련한 지표들을 활용하여 건물 안전 평가 점수를 산출하고 유지관리 및 보수가 더욱 빠르게 이루어지도록 시스템을 구축할 수 있다.

### III. 결 론

본 연구에서는 육안 점검 형식의 균열 검사를 대체할 수 있는 균열 검출 모델과 시스템을 제안했다. 현장점검 상황에 유사한 환경으로써 POC 데이터셋을 구축하여 기존 균열 관련 연구에서의 도메인 부적응

문제를 해결하였고, 증화 표집 방법과 독자적인 라벨링 방법을 적용함으로써 검출 모델이 검출하기 까다로운 수 있는 데이터에 대해서도 대처할 수 있었다. 딥러닝 기반의 YOLO-Cr 모델은 POC 데이터셋에 대해 높은 정밀도와 함께 뛰어난 검출 성능을 보였고, 실제 점검에 적용하기에도 적합할 정도로 안정적이었다.

본 연구를 통해 기존의 균열 검사 방식과 균열 검출 모델의 한계를 해결하였으나, 추가로 고도화할 수 있는 방향성은 다음과 같다. POC 데이터셋의 경우 데이터 규모에 대한 고찰이 필요한데, 추가적인 데이터를 통해 성능을 높일 수 있는 여지를 확인할 수 있다. 또한, 미검출된 균열이 존재하여 비슷한 패턴의 데이터에 대해 적대적 생성 신경망(Generative Adversarial Network)과 같은 생성 알고리즘을 기반으로 데이터를 증강하여 문제를 보완할 수 있다. 균열과 같이 점검하게 되는 박리, 박락, 누수, 철근 노출 등도 종합적으로 점검할 수 있도록 보강할 필요성이 있으며 도로, 교량, 터널 등 다양한 곳에서의 균열을 포함하여 진행해볼 수 있다. 열화상 카메라, 깊이 카메라와 같은 다른 촬영 기법을 종합하여 일반 컬러 이미지만으로 알 수 없는 균열에 대한 세부 정보를 얻을 수 있을 것이다. 시스템에 대한 설계 수준으로 YOLO-Cr을 통한 적용을 고려하였으나, 이에 대한 구체적인 실증으로 관련 모델과 시스템의 실효성을 검증할 필요가 있을 것이다.

노후화 건물이 늘어날수록 건물 안전 점검, 특히 전문 점검자가 직접 확인해야 하는 균열 점검에 대한 효과적인 시스템의 수요는 늘어날 것이다. 본 연구에서 개발한 YOLO-Cr 모델과 시스템을 기반으로 균열 점검의 자동화가 적용되어 현재의 점검 방식을 보완할 수 있길 기대한다.

### References

- [1] Ministry of Land, *Infrastructure and Transport, Guidelines for Safety and Maintenance of Facilities*(2021), Retrieved Nov. 6, 2022, from <http://www.assi.or.kr/sub/law/equipment/detailcheck.asp?TS=4>
- [2] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *J. Comput. in Civil Eng.*, vol. 17, no. 4, pp. 255-263, Oct. 2003. ([https://doi.org/10.1061/\(ASCE\)0887-3801\(2003\)17:4\(255\)](https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)))

- [3] N. D. Hoang, "Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding," *Advances in Civil Eng.*, vol. 2018, Article ID 3924120, 10 pages, Apr. 2018.  
(<https://doi.org/10.1155/2018/3924120>)
- [4] B. Oswald-Tranta, "Induction thermography for surface crack detection and depth determination," *Appl. Sci.*, vol. 8, no. 2, pp. 257, Feb. 2018.  
(<https://doi.org/10.3390/app8020257>)
- [5] Ç. F., Özgenel and A. G. Sorguç, "Performance comparison of pretrained convolutional neural networks on crack detection in buildings," in *Proc. Int. Symp. Automat. and Robotics in Construction*, pp. 1-8, Berlin, Germany, Jul. 2018.  
(<https://doi.org/10.22260/ISARC2018/0094>)
- [5] M. Maguire, S. Dorafshan, and R. J. Thomas, *SDNET2018: A concrete crack image dataset for machine learning applications*(2018), Retrieved Nov. 6, 2022, from [https://digitalcommons.usu.edu/all\\_datasets/48/](https://digitalcommons.usu.edu/all_datasets/48/)  
(<https://doi.org/10.15142/T3TD19>)
- [7] S. Kulkarni, S. Singh, D. Balakrishnan, S. Sharma, S. Devunuri, and S. C. R. Korlapati, *Crackseg9k: A Collection of Crack Segmentation Datasets*(2022), Retrieved Nov. 6, 2022, from <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EGIEBY>  
(<https://doi.org/10.7910/DVN/EGIEBY>)
- [8] G. Y. Kim, D. H. Kim, and D. J. Kim, "Deep learning-based exterior wall crack detection method for building safety diagnosis," *J. Korean Inst. Plant Eng.*, vol. 26, no. 4, pp. 31-43, Dec. 2021.
- [9] AIHub, *Building Crack Detection Image* (2022), Retrieved Nov. 6, 2022, from <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=162>
- [10] J. W. Ha, K. W. Park, and M. S. Kim, "A development of road crack detection system using deep learning-based segmentation and object detection," *The J. Soc. e-Busin. Stud.*, vol. 26, no. 1, pp. 93-106, Feb. 2021.  
(<https://doi.org/10.7838/jsebs.2021.26.1.093>)
- [11] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 85-101, Jun. 2020.  
(<https://doi.org/10.26599/BDMA.2019.9020015>)
- [12] G. Y. Kim, H. R. Lee, and D. J. Kim, "Robust detection deep learning model in the various exterior wall cracks," in *Proc. Korean Soc. Comput. Inf. Conf.*, Jeju Island, Korea, Jul. 2021.
- [13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. CVPR*, pp. 779-788, Las Vegas, NV, USA, Jun. 2016.  
(<https://doi.org/10.1109/cvpr.2016.91>)
- [14] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, Jul. 2022.  
(<https://doi.org/10.48550/arXiv.2207.02696>)
- [15] G. Jocher, A. Chaurasia, A. Stoken, et al., "ultralytics/yolov5: v6.2 - YOLOv5 classification models, apple m1, reproducibility, clearml and deci.ai integrations," *Zenodo*, Aug. 2022.  
(<https://doi.org/10.5281/zenodo.7002879>)
- [16] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, Apr. 2020.  
(<https://doi.org/10.48550/arXiv.2004.10934>)
- [17] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. CVPR Wkshps.*, pp. 390-391, Seattle, WA, USA, Jun. 2020.

- (<https://doi.org/10.1109/cvprw50498.2020.00203>)
- [18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, Apr. 2018.  
(<https://doi.org/10.48550/arXiv.1804.02767>)
- [19] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. CVPR*, pp. 1492-1500, Honolulu, HI, USA, Jul. 2017.  
(<https://doi.org/10.1109/cvpr.2017.634>)
- [20] C. Y. Wang, I. H. Yeh, and H. Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," *arXiv preprint arXiv:2105.04206*, May 2021.  
(<https://doi.org/10.48550/arXiv.2105.04206>)
- [21] S. N. Sitara, "Review and analysis of crack detection and classification techniques based on crack types," *Int. J. Appl. Eng. Res.*, vol. 13, no. 8, pp. 6056-6062, 2018.  
(<https://dx.doi.org/10.37622/IJAER/13.8.2018.6056-6062>)

**김 지 호 (Ji-Ho Kim)**



2021년 2월: 경북대학교 심리학과/심리정보융합전공 학사  
2021년 9월~현재: 포항공과대학교 인공지능연구원 연구원  
<관심분야> Computer Vision, Image processing, Deep learning

[ORCID:0009-0005-0132-9769]

**김 경 영 (Gyeong-Yeong Kim)**



2015년 2월: KAIST 수리과학과 학사  
2018년 2월: KAIST 건설및환경공학과 석사  
2020년 3월~현재: 포항공과대학교 인공지능연구원 팀장  
<관심분야> Computer Vision, Face Recognition, Deep learning

[ORCID:0000-0001-6891-4130]

**김 동 주 (Dong-Ju Kim)**



1998년 2월: 충북대학교 전과 공학과 학사  
2000년 2월: 충북대학교 전과 통신학과 석사  
2010년 8월: 성균관대학교 전기전자 및 컴퓨터공학과 박사

2016년 9월~현재: 포항공과대학교 인공지능연구원 연구부장/연구부교수  
<관심분야> Computer Vision, Image processing, Deep learning

[ORCID:0009-0009-6950-4200]