

콘크리트 라이닝 균열 분할 딥러닝 모델 평가 방법

함상우¹ · 배수현² · 이임평³ · 이규필⁴ · 김동규^{5*}

¹정회원, 서울시립대학교 대학원 공간정보공학과 박사과정

²비회원, 서울시립대학교 대학원 공간정보공학과 석사과정

³비회원, 서울시립대학교 공간정보공학과 정교수

⁴정회원, 한국건설기술연구원 지반연구본부 수석연구원

⁵정회원, 한국건설기술연구원 지반연구본부 연구위원

An evaluation methodology for cement concrete lining crack segmentation deep learning model

Sangwoo Ham¹ · Soohyeon Bae² · Impyeong Lee³ · Gyu-Phil Lee⁴ · Donggyou Kim^{5*}

¹Ph.D. Candidate, Dept. of Geoinformatics, University of Seoul

²Master Student, Dept. of Geoinformatics, University of Seoul

³Professor, Dept. of Geoinformatics, University of Seoul

⁴Senior Researcher, Dept. of Geotechnical Engineering Research, Korea Institute of Civil Engineering and Building Technology

⁵Research Fellow, Dept. of Geotechnical Engineering Research, Korea Institute of Civil Engineering and Building Technology

*Corresponding Author : Donggyou Kim, dgkim2004@kict.re.kr



OPEN ACCESS

Journal of Korean Tunnelling and
Underground Space Association
24(6)513-524(2022)
<https://doi.org/10.9711/KTAJ.2022.24.6.513>

eISSN: 2287-4747

pISSN: 2233-8292

Received September 26, 2022

Revised October 18, 2022

Accepted October 18, 2022



This is an Open Access article
distributed under the terms of the
Creative Commons Attribution

Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2022, Korean Tunnelling and Underground
Space Association

Abstract

Recently, detecting damages of civil infrastructures from digital images using deep learning technology became a very popular research topic. In order to adapt those methodologies to the field, it is essential to explain robustness of deep learning models. Our research points out that the existing pixel-based deep learning model evaluation metrics are not sufficient for detecting cracks since cracks have linear appearance, and proposes a new evaluation methodology to explain crack segmentation deep learning model more rationally. Specifically, we design, implement and validate a methodology to generate tolerance buffer alongside skeletonized ground truth data and prediction results to consider overall similarity of topology of the ground truth and the prediction rather than pixel-wise accuracy. We could overcome over-estimation or under-estimation problem of crack segmentation model evaluation through using our methodology, and we expect that our methodology can explain crack segmentation deep learning models better.

Keywords: Deep learning, Semantic segmentation, Performance evaluation

초 록

터널을 비롯한 여러 가지 기반시설물에 발생한 콘크리트 균열을 영상과 딥러닝 기반으로 자동 탐지하는 연구가 최근 활발히 이루어지고 있다. 이러한 연구성과를 실제 현장에 적용하려면 딥러닝 모델의 신뢰성을 설명할 수 있어야 한다. 본 연구에서는 선형성이 강한 균열의 기하적인 특성을 고려했을 때 화소 기반으로 계산하는 기존 평가지표가 충분치 않다는 점을 지적하며, 균열 분할 딥러닝 모델의 성능을 더 합리적으로 설명할 수 있는 다른 평가지표를 제시하고 비교 분석한다. 먼저 선형 객체의 유사성을 측정할 수 평가방법을 제시한다. 구체적으로는 기준 데이터에 허용 버퍼(tolerance buffer)를 부여하여 평가하는 방법을 설계, 구현, 검증한다. 실험 결과 본 연구에서 제안하는 방법은 균열 분할 딥러닝 모델 평가시 기존 대비 과대평가 또는 과소평가 문제를 해결할 수 있었으며, 화소 기반 성능 평가 지표에 비해 균열 분할 딥러닝 모델의 성능을 더 잘 설명할 것으로 기대한다.

주요어: 딥러닝, 의미론적 분할, 성능평가

1. 서 론

딥러닝 모델의 성능 평가는 모델 개발 과정에서 필수적인 단계다. 학습에 사용하지 않은 시험용 데이터셋을 이용하여 여러 가지 지표를 계산하고 이를 통해 모델 성능을 정량적으로 표현하여야 모델의 취약점과 개선점을 도출할 수 있기 때문이다. 컴퓨터 비전 분야 딥러닝 모델의 성능 평가 지표에는 재현율(recall), 정밀도(precision), f1-score, mAP (mean average precision), mIoU (mean intersection over union), dice coefficient 등이 있으며, 각 지표는 용도에 맞게 활용하여야 한다. 예를 들어 주어진 영상으로부터 화소 단위로 분류를 수행하는 의미론적 분할(semantic segmentation) 모델의 성능을 평가할 때는 주로 mIoU (mean intersection over union)를 사용한다 (Garcia-Garcia et al., 2017).

의미론적 분할 딥러닝 모델은 영상 기반으로 터널 등 기반시설에 발생한 콘크리트 균열을 탐지하는 연구에 많이 활용된다(Hsieh and Tsai, 2020). 의미론적 분할 모델은 균열의 형상을 화소 단위로 분류하여 표현하므로 균열의 형상과 위치를 곧바로 알 수 있어 유용하기 때문이다. Kim et al. (2018)은 객체 분류 모델과 의미론적 분할 모델을 조합하여 균열을 추출하였으며, Liu et al. (2019)은 콘크리트 표면 영상에서 균열을 분할하는 딥러닝 모델인 DeepCrack을 제안하였고, Paik et al. (2021)은 의미론적 분할 모델을 균열 탐지를 위해 사용하였다. 또한, 국내에서는 AI Hub에서는 건물 균열 탐지 이미지를 공개하여 연구자, 개발자들이 균열 탐지 딥러닝 모델을 학습시킬 수 있도록 데이터를 공개하였으며(Park, 2022), 해외에서는 COCO-Bridge 데이터셋(Bianchi et al., 2021), Crack Segmentation Dataset (Middha, 2020), CODEBRIM 데이터셋(Mundt et al., 2019) 등이 공개되었다. 특히, 터널의 콘크리트 라이닝에 발생한 균열은 빅아이(Big Eye)와 같은 라인카메라 기반 영상취득 시스템과 딥러닝을 결합하여 효율적으로 파악할 수 있다(Ham et al., 2021).

그러나 균열을 분할하는 모델을 평가할 때 앞서 제시한 보편적인 평가지표를 적합한가에 대해서는 의문이 제기

된다. 재현율, 정밀도, f1-score, mIoU 등은 화소 단위로 true positive, false positive, false negative의 개수를 파악하여 계산하는데, 이러한 특징을 놓고 볼 때 화소 단위 성능 평가는 폭이 좁고 길이가 길어 선형성이 강한 균열의 특성을 제대로 반영하지 못할 수 있어 새로운 평가 방법론이 필요하다. 예를 들어, Tsai and Chatterjee (2017)는 균열 탐지 알고리즘 평가를 위한 새로운 방법으로 향상된 하우스도프 거리(enhanced Hausdorff distance, EHD)를 제시하였으며, 기준 데이터로부터 상대적인 거리에 따라 상대적인 패널티를 부과하도록 하였다.

구체적으로는 Fig. 1과 같은 상황을 생각해볼 수 있다. 기준 데이터(검은 선)에 비해 1화소 차이나는 지점에서 균열을 탐지한 모델 A (파란 선)와 10화소 차이나는 지점에서 균열을 탐지한 모델 B (빨간 선)가 있다고 했을 때, 화소 기반 성능 평가 수행 시 해당 화소를 같은 가중치로 false positive 취급하므로 성능이 같은 것으로 처리한다. 모델 A와 같은 경우는 실제로 잘못 탐지한 것일 수도 있지만, 전체적인 균열 추세에서 조금만 벗어나서 비교적 작은 문제일 수 있으며, 이러한 정도의 오탐지는 라벨링 오류에서 기인했을 가능성도 높기 때문이다.



Fig. 1. Over-estimation and under-estimation problem of pixel-based performance evaluation (Black line depicts ground truth. Blue line and red line depicts the model A and the model B. In this case, the evaluation metric will be almost identical, so the performance of the model A is over-estimated, and vice versa.)

따라서 본 연구는 균열의 특성을 고려한 새로운 평가 지표의 필요성을 위와 같이 제기하고, 스켈레톤화(skeletonization) 및 허용 버퍼(tolerance buffer)에 기반한 딥러닝 모델 평가 방법을 Fig. 2와 같이 설계하고 구현한다. 먼저 주어진 영상으로부터 딥러닝 모델을 이용하여 균열을 분할한다. 기준 데이터와 탐지 결과로부터 스켈레톤화(Zhang and Suen, 1984)를 수행하여 기준 데이터와 탐지 결과에 나타난 균열의 위상(topology)을 추출하고, 기준

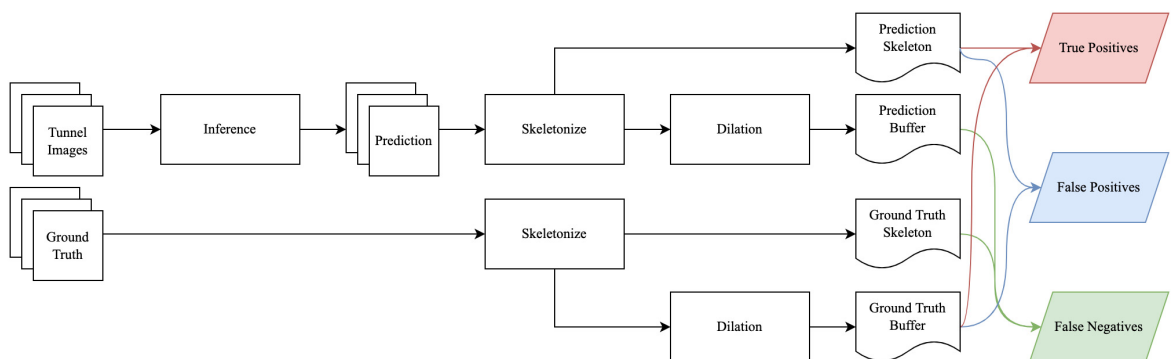


Fig. 2. Research overview

데이터 및 탐지 결과에 팽창(dilation) 연산을 적용하여 일정 거리 범위 내에서는 기준 데이터와 탐지 결과가 차이가 나더라도 true positive 처리하도록 한다. 이를 통해 전체적인 균열 추세에서 크게 벗어나지 않는 화소와 그렇지 않은 화소를 분리할 수 있으며 이를 통해 과대평가나 과소평가 문제를 해소할 수 있다.

2. 실험 방법

2.1 데이터셋 구성 및 딥러닝 모델 학습

본 연구에서는 캐글에서 공개한 Crack segmentation dataset (Middha, 2020; 이하 캐글 데이터셋)과 터널 콘크리트 라이닝 영상 취득 장비인 빅아이(Big Eye)로 구축한 영상을 DeepLabv3+ (Chen et al., 2018)에 학습시킨다. 빅아이는 2.5 t급 트럭에 라인 카메라, 조명, 영상처리 장치를 장착한 터널 스캐닝용 특수 차량으로, 전용 소프트웨어를 이용하여 터널의 전개 영상을 획득할 수 있고, 전개 영상에 균열의 위치를 Fig. 3과 같이 폴리라인(polyline) 형태로 표현할 수 있다.

Ham et al. (2021)은 빅아이 데이터를 딥러닝용으로 가공하는 방법을 제시하였으며, 캐글 데이터셋을 전부 학습시키는 것보다 캐글 데이터셋 중 도로 포장 영상 등을 제외하고 터널 콘크리트 라이닝과 유사한 영상만 이용하여 딥러닝 모델에 대한 사전학습을 수행하였을 때 빅아이 영상자료에서 추론 성능이 좋았다는 점을 확인하였다. 따라서 본 연구에서는 캐글 데이터셋 및 딥러닝용으로 가공된 빅아이 데이터셋을 함께 학습시킨 후, 빅아이 데이터셋에서 본 연구에서 제시하는 평가 방법을 적용해본다.

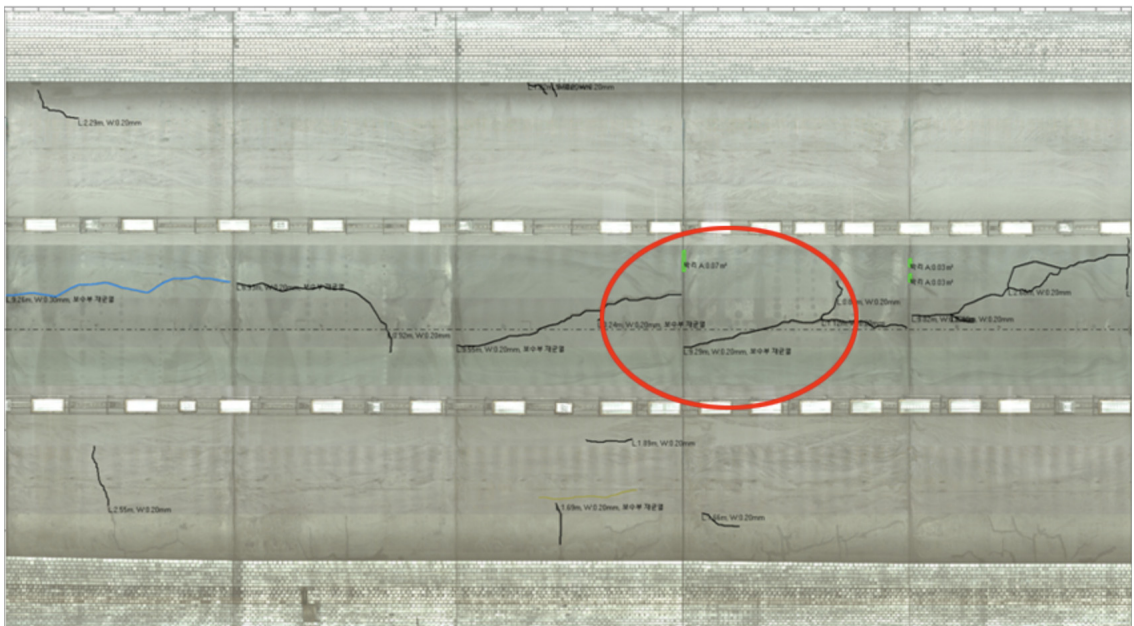


Fig. 3. An example of Big Eye tunnel concrete lining image with crack polylines (OurBrain, 2022)

2.2 스켈레톤화 및 허용 버퍼 생성

스켈레톤화는 형태학적 연산(morphological operations)의 일종으로 이진 영상을 폭이 1화소인 형태로 표현하는 것을 말하며, 허용 버퍼는 스켈레톤화 후 팽창 연산을 통해 생성한다. 스켈레톤화와 팽창 연산의 예시는 Fig. 4에 표현되어있다. 기준 데이터와 탐지 결과에 스켈레톤화를 적용하여 균열의 위상을 추출한다. 허용 버퍼는 형태학적 연산의 일종인 팽창 연산으로 생성한다. 기준 데이터와 탐지 결과 모두에 허용 버퍼를 생성하고 저장한다.

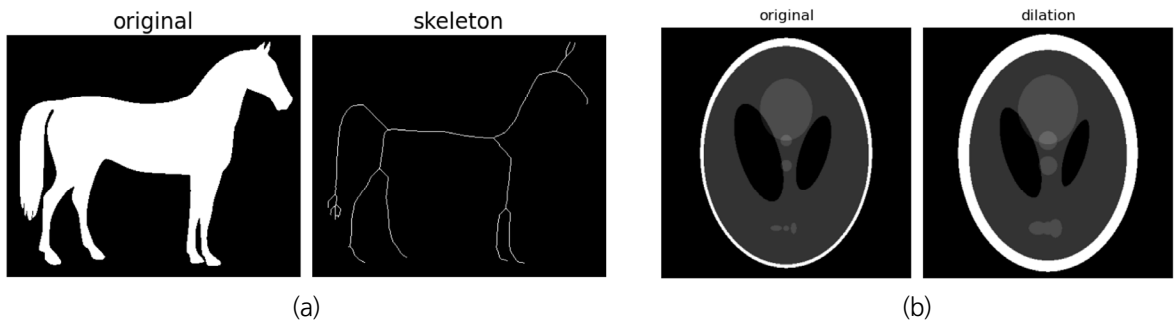


Fig. 4. Skeletonization and dilation (Scikit-image contributors, 2022)

2.3 재현율, 정밀도, f1-score 계산

재현율, 정밀도, f1-score를 계산하기 위해서는 먼저 Fig. 5와 같이 true positive (TP), false negative (FN), false positive (FP)에 해당하는 화소의 개수를 계산하여야 한다. TP와 FP의 경우, Positive로 즉 균열로 탐지된 픽셀들이 실제 균열에 해당하는 지 여부를 확인한다. 이때, 실제 균열을 의미하는 기준 데이터와 비교하게 되는데, 기준 데이터에 허용 버퍼를 적용한다. 기준 데이터와 완전히 일치하지 않더라도 일정 범위안에 있으면 TP로, 그렇지

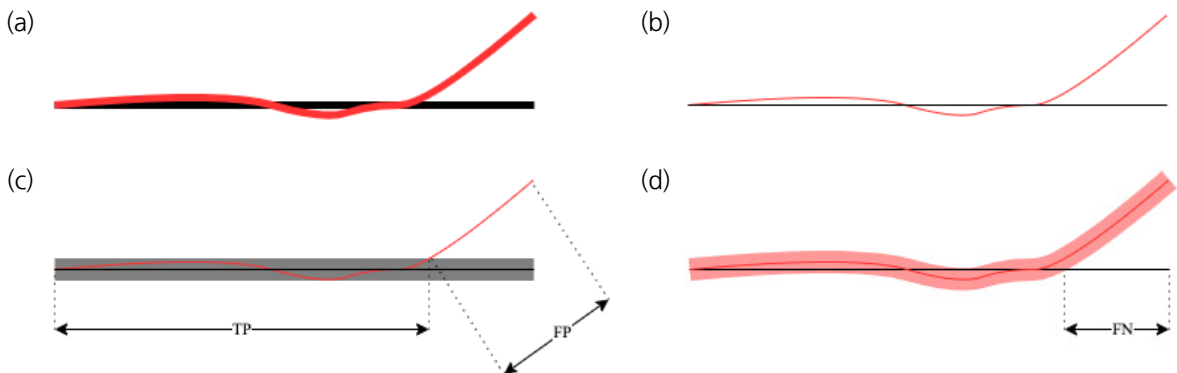


Fig. 5. Calculation of TP, FP and FN: (a) Black line and red line represents ground truth and prediction, respectively, (b) Depicts skeletonized ground truth and prediction, (c) Depicts calculation of TP and FP using the buffered ground truth, (d) Depicts calculation of FN using the buffered prediction

않으면 FP로 분류한다. FN의 경우, 실제 균열인 픽셀들 중에서 Negative로, 즉 균열로 탐지되지 않은 픽셀들을 점검한다. 이때, 실제 균열인 픽셀들을 균열로 탐지된 데이터에 허용 버퍼를 적용하여 비교한다. 탐지된 데이터에 일정 범위 밖으로 벗어나는 경우 FN으로 분류한다. FP와 FN를 계산할 때 각각 기준 데이터에 생성한 허용 버퍼, 탐지 결과에 생성한 허용 버퍼를 이용하는데, 이는 버퍼의 폭으로 인해 FP 또는 FN이 과다하게 계산되지 않도록 하기 위함이다. 그리고 계산한 TP, FP, FN을 토대로 재현율, 정밀도, f1-score를 계산한 후, 제안하는 방법의 유효성을 확인하기 위해 스켈레톤화 및 허용 버퍼를 적용하지 않고 계산한 재현율, 정밀도, f1-score와 비교한다.

3. 실험 결과

3.1 딥러닝 모델 학습

본 연구에서 제안하는 기법을 적용하기 위해 딥러닝 모델을 학습시킨다. 딥러닝 모델은 DeepLabv3+를 사용하며, 특징 추출을 위한 백본(backbone)은 Resnet50을 사용하고, 학습을 위한 소프트웨어로는 여러 가지 의미론적 분할 모델을 통합적으로 제공하는 API인 MMSegmentation을 이용한다(MMSegmentation Contributors, 2020). 데이터셋은 캐글에서 제공하는 Crack Segmentation Dataset 가운데 터널의 질감과 관련성이 높은 자료와, 빅아이 시스템으로 촬영한 라인 스캐닝 자료에 균열 영역을 라벨링한 자료를 혼합하여 구성한다. 그 결과 학습 데이터셋으로는 512 by 512 화소 크기의 영상 패치 3,767매, 시험 데이터셋으로는 80매를 구축할 수 있었다. 미니 배치 크기는 4, 최적화 알고리즘은 Adam, 학습률은 0.0001이었으며, 총 100 에포크 학습하였다.

3.2 스켈레톤화 및 허용 버퍼 기반 평가 결과

시험 데이터 80장에 대하여 본 연구에서 제안하는 방법을 적용하여 모델 성능 평가를 수행하였다. 스켈레톤화를 적용하지 않은 실험과 스켈레톤화를 적용한 실험을 수행하여 스켈레톤화의 영향을 살펴보았다. 허용 버퍼의 반경은 10화소로, 빅아이 장비의 공간해상도가 1 mm/px.인 점을 고려할 때 약 1 cm 내외가 된다. 스켈레톤화 및 허용 버퍼 기반 평가 결과 시험 데이터에 대한 재현율, 정밀도, f1-score는 아래 Table 1과 같았다. 스켈레톤화를 적용하지 않은 경우 재현율은 28.89%p 상승하였으며, 정밀도는 49.40%p 상승하였고, f1-score는 32.03%p 상승하였다. 스켈레톤화를 적용한 경우 재현율은 23.18%p 상승하였으며, 정밀도는 47.56%p 상승하였고, f1-score는 32.03%p 상승하였다.

Table 1. Recall, precision and f1-score of original data and our methodology

	Recall (original, %)	Precision (original, %)	f1-score (original, %)
Original	33.95	47.68	39.39
Ours w/o skeletonization	62.84	97.08	76.30
Ours w/ skeletonization	57.13	95.24	71.42

위 실험 결과를 놓고 볼 때 전반적으로는 제안하는 방법을 적용했을 때 평가 지표가 상승하는 것을 확인할 수 있다. 스켈레톤을 적용한 경우가 그렇지 않은 경우에 비해 상승폭이 약간 작았는데, 이는 균열의 폭을 스켈레톤화를 통해 1화소로 줄이는 과정에서 균열의 폭이 고려되지 않았기 때문으로 예상된다.

본 연구에서 제안하는 스켈레톤화 및 허용 버퍼 기반 평가 방법을 적용했을 때와 그렇지 않았을 때의 양상을 비교하기 위해 X축을 기존 평가 방법에서의 재현율 또는 정밀도로 하고 Y축을 제안하는 방법에서의 재현율 또는 정밀도로 하는 산점도를 그린 후 $y=x$ 를 표현하는 직선을 표시하였다. 스켈레톤화를 적용하지 않았을 때의 결과는 Fig. 6에 표현되어있으며, 스켈레톤화를 적용했을 때의 결과는 Fig. 7에 표현되어있다. 각 산점도에서 $y=x$ 로부터 가장 먼 3개의 점은 본 연구에서 제시하는 평가 방법을 적용했을 때 기본 방법에 비해 평가 결과가 많이 달라진 것을 의미한다. 따라서 각 데이터 포인트는 $y=x$ 에서 멀수록 붉게, 가까울수록 파랗게 표현하였다. 그리고, 각 산점도에서 $y=x$ 로부터 가장 먼 3개의 데이터 포인트의 재현율, 정밀도, f1-score를 Table 2에 나타내었으며, 해당 데이터 포인트의 기준 데이터와 탐지 결과를 중첩하여 시각화한 자료를 Fig. 8과 Fig. 9에 표현하였다. Fig. 8과 Fig. 9에서 TP는 붉은색, FP는 초록색, FN은 파란색으로 표현했다. 이를 통해 본 연구에서 제안하는 방법이 모델 성능의 저평가/고평가 문제를 충분히 해결할 수 있는지 확인해보고자 하며, 목적을 달성하기 위해 제시된 자료로부터 재현율 및 정밀도에 대해 스켈레톤화 적용 여부에 따른 평가 결과를 정량적 측면과 정성적 측면에서 분석한다.

첫째, 스켈레톤화를 적용하지 않고 제안하는 방법을 적용했을 때 재현율의 변화를 살펴본다. Fig. 6(a)에서 확인할 수 있듯이 거의 모든 데이터 포인트가 $y=x$ 의 상부에 위치하여 재현율이 전반적으로 높아진 것을 알 수 있다. 재현율이 가장 높아진 세 가지 데이터인 Fig. 8(a)~8(c)를 살펴보면 실제 균열과 인접한 부분의 탐지 결과가 기존에는 FP 처리되었지만 제안하는 방법을 적용했을 때 TP 처리되었음을 확인할 수 있다.

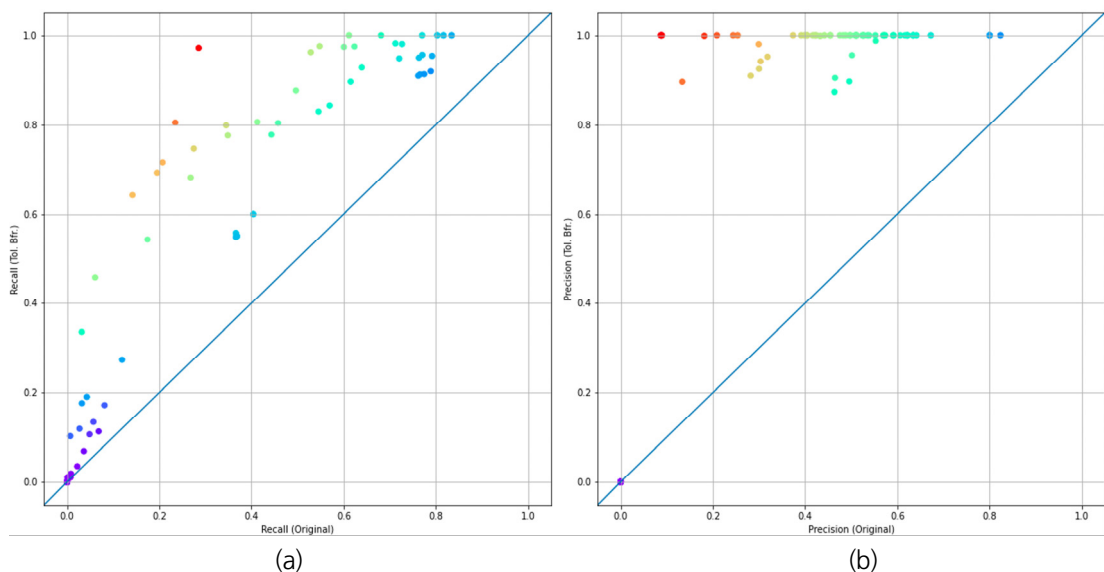


Fig. 6. Scatterplot of recall and precision (ours, w/o skeletonization)

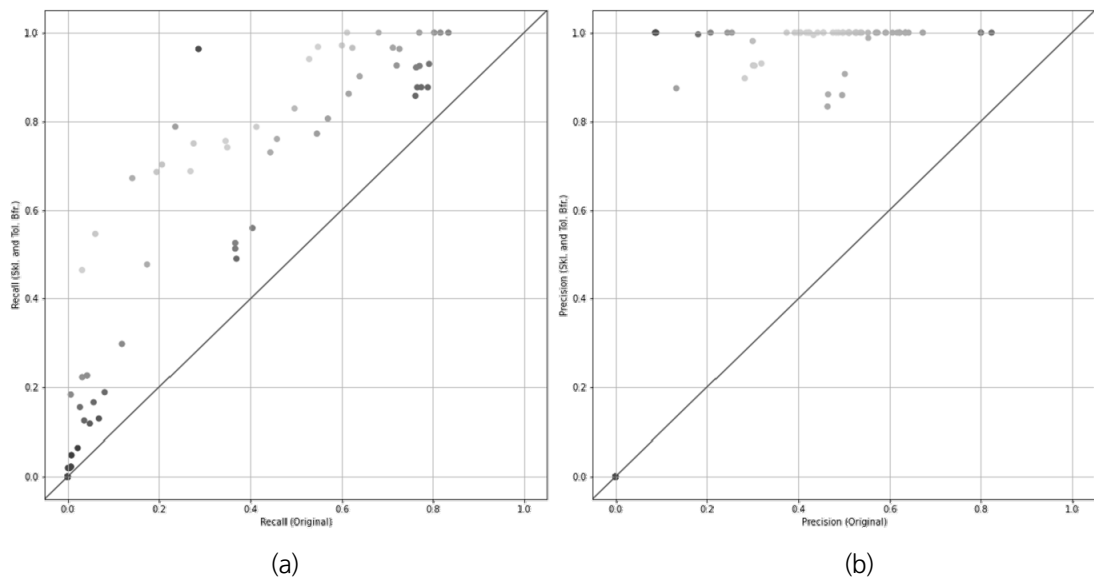


Fig. 7. Scatterplot of recall and precision (ours, w/ skeletonization)

Table 2. Recall, precision, f1-score of top 3 farthest data points from $y = x$

# of Figure	Recall (original, %)	Precision (original, %)	f1-score (original, %)	Recall (ours, %)	Precision (ours, %)	f1-score (ours, %)
Fig. 8(a)	28.55	37.38	32.37	97.22	100.00	98.59
Fig. 8(b)	23.46	30.01	26.33	71.59	95.24	81.74
Fig. 8(c)	20.72	31.84	25.10	64.19	98.06	77.59
Fig. 8(d)	28.55	37.38	32.37	0.97	100.00	1.91
Fig. 8(e)	23.46	30.01	26.33	10.22	100.00	18.55
Fig. 8(f)	20.72	31.84	25.10	45.81	99.86	62.81
Fig. 9(a)	28.55	37.38	32.37	96.31	100.00	98.12
Fig. 9(b)	23.46	30.01	26.33	78.78	92.61	85.14
Fig. 9(c)	14.19	29.93	19.26	67.17	98.11	79.74
Fig. 9(d)	0.07	8.70	0.16	1.97	100.00	3.86
Fig. 9(e)	0.69	8.93	1.28	18.36	100.00	31.02
Fig. 9(f)	6.08	18.13	9.11	54.60	99.62	70.54

둘째, 스켈레톤화를 적용하지 않고 제안하는 방법을 적용했을 때 정밀도의 변화를 살펴본다. Fig. 6(b)에서 확인할 수 있듯이 거의 모든 데이터 포인트가 $y = x$ 의 상부에 위치하여 정밀도가 전반적으로 높아진 것을 알 수 있다. 정밀도가 가장 높아진 세 가지 데이터인 Fig. 8(d)~8(f)를 살펴보면 실제 균열과 인접한 부분의 탐지 결과가 기존에는 FP 처리되었지만 제안하는 방법을 적용했을 때 TP 처리되었음을 확인할 수 있다. 그러나 세 가지 데이터

모두 재현율이 현저히 낮아 탐지 결과 자체는 좋지 못하다고 할 수 있다. 이러한 예를 살펴봤을 때 균열탐지 결과를 분석하려면 반드시 재현율과 정밀도를 동시에 확인해야 함을 알 수 있다.

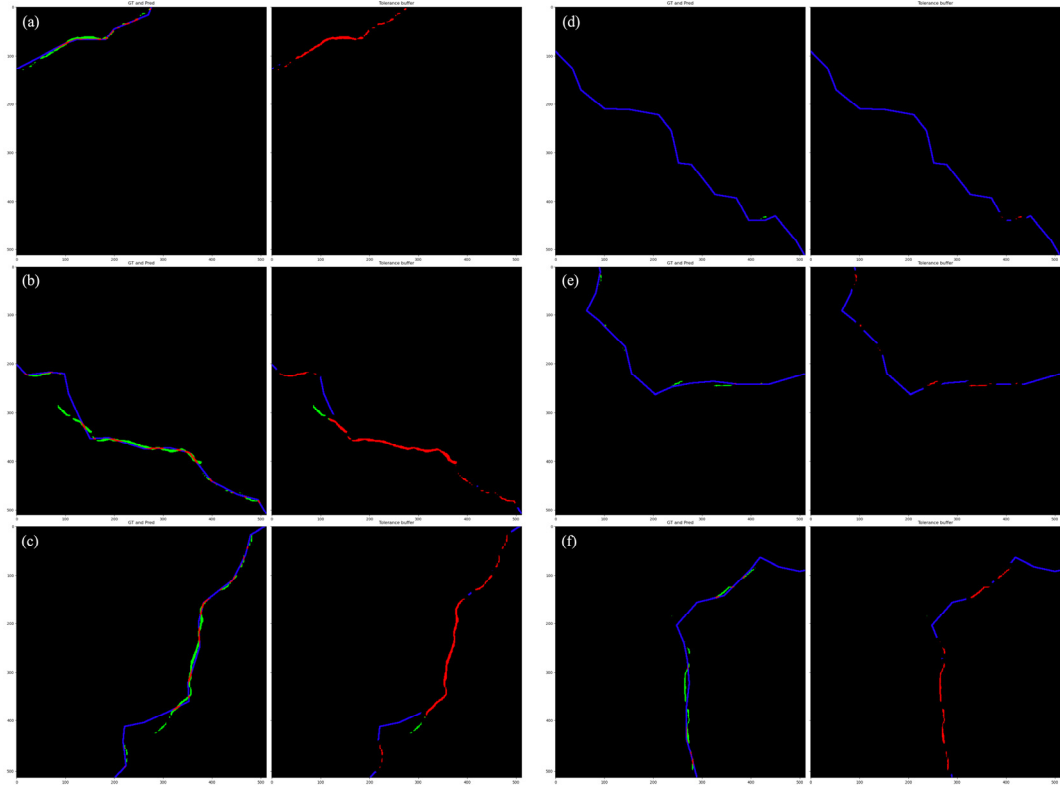


Fig. 8. Evaluation results from top-3 farthest points from the scatterplot (ours, w/o skeletonization): (a)~(c) Depicts 3 farthest points from the recall scatterplot, (d)~(f) Depicts 3 farthest points from the precision scatterplot

셋째, 스켈레톤화를 적용하고 제안하는 방법을 적용했을 때 재현율의 변화를 살펴본다. Fig. 7(a)에서 확인할 수 있듯이 거의 모든 데이터 포인트가 $y = x$ 의 상부에 위치하여 재현율이 전반적으로 높아진 것을 알 수 있다. 재현율이 가장 높아진 세 가지 데이터인 Fig. 9(a)~9(c)를 살펴보면 실제 균열과 인접한 부분의 탐지 결과가 기존에는 FP 처리되었지만 제안하는 방법을 적용했을 때 TP 처리되었음을 확인할 수 있다.

넷째, 스켈레톤화를 적용하고 제안하는 방법을 적용했을 때 정밀도의 변화를 살펴본다. Fig. 7(b)에서 확인할 수 있듯이 거의 모든 데이터 포인트가 $y = x$ 의 상부에 위치하여 정밀도가 전반적으로 높아진 것을 알 수 있다. 정밀도가 가장 높아진 세 가지 데이터인 Fig. 9(d)~9(f)를 살펴보면 실제 균열과 인접한 부분의 탐지 결과가 기존에는 FP 처리되었지만 제안하는 방법을 적용했을 때 TP 처리되었음을 확인할 수 있다. 그러나 세 가지 데이터 모두 재현율이 현저히 낮아 탐지 결과 자체는 좋지 못하다고 할 수 있다. 둘째 경우와 마찬가지로, 이러한 예를 살펴봤을 때 균열탐지 결과를 분석하려면 반드시 재현율과 정밀도를 동시에 확인해야 함을 알 수 있다.

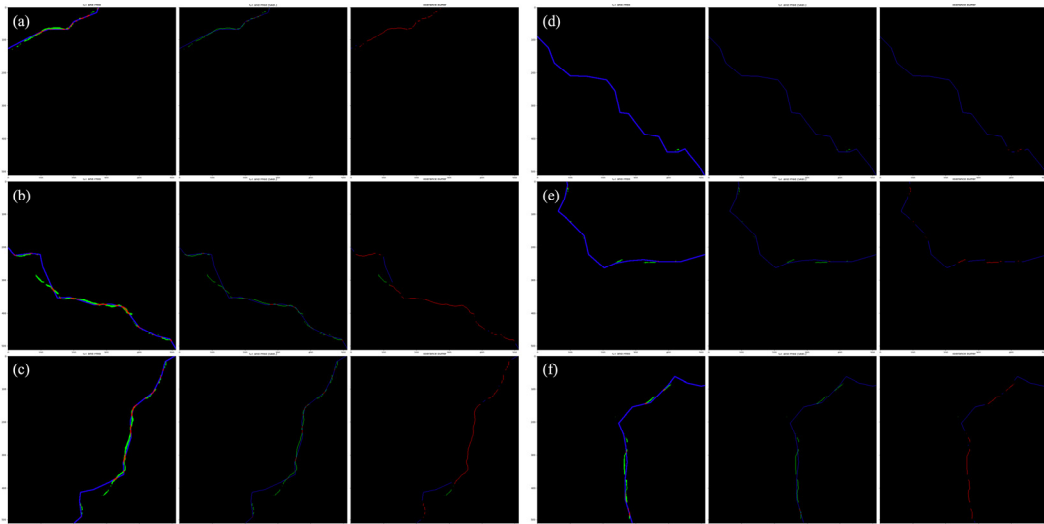


Fig. 9. Evaluation results from top-3 farthest points from the scatterplot (ours, w/ skeletonization): (a)~(c) Depicts 3 farthest points from the recall scatterplot, (d)~(f) Depicts 3 farthest points from the precision scatterplot

Fig. 10은 Fig. 8과 Fig. 9에서 기존 화소 기반 방법과 본 연구에서 제시하는 방법을 적용하였을 때 계산한 재현율, 정밀도, f1-score를 나타낸 막대 그래프이다.

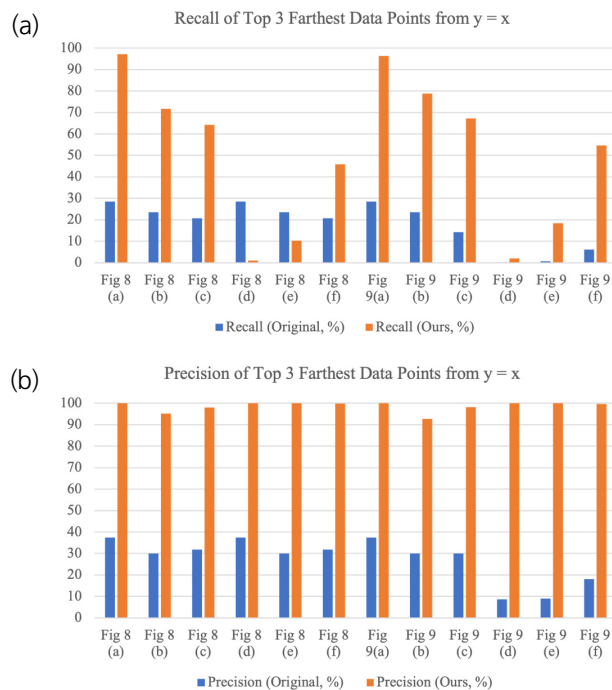


Fig. 10. Recall, precision, f1-score of top 3 farthest data points from $y = x$

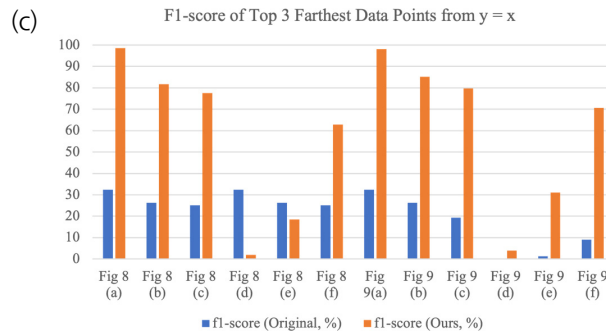


Fig. 10. Recall, precision, f1-score of top 3 farthest data points from $y = x$ (continued)

4. 결론

본 연구에서는 터널 라이닝 영상으로부터 균열을 분할하는 딥러닝 모델의 성능을 평가하는 새로운 방법을 제안하였다. 영상분할 모델이 주로 다루는 거리 레벨에서의 객체(차량, 보행자, 건물, 도로 등)와 달리 균열은 폭이 매우 얇은 특성이 있어 기존 모델 평가 방법을 그대로 사용한다면 모델의 성능이 과소평가되거나 과대평가될 우려가 크다. 따라서 화소 단위로 균열 분할 모델의 성능을 평가하기보다는 전체적인 위상의 일치도를 확인하기 위해 스켈레톤화 및 허용 버퍼 기반의 평가 방법을 설계, 구현, 시험해보았다. 이러한 방법을 통해 기존 평가 방법으로는 모델의 성능이 과소평가되는 문제를 해소할 수 있었으며, 향후 위성영상에 나타난 도로 및 철도와 같은 선형 객체와 같이 폭이 매우 얇은 객체에 대한 딥러닝 모델 성능 평가에 본 연구가 제시하는 방법을 활용할 수 있을 것으로 전망한다.

감사의 글

본 논문은 한국건설기술연구원 주요사업으로 지원을 받아 수행된 연구(인공지능을 활용한 대심도 지하 대공간의 스마트 복합 솔루션 개발)로 이에 감사합니다.

저자 기여도

함상우는 실험 설계, 딥러닝 모델 구현, 전처리 프로세스 설계, 원고 작성을 수행하였음. 배수현은 실험 수행, 빅아이 데이터에 대한 전처리 프로세스 적용, 선행연구 조사를 수행하였음. 이임평은 연구 개념 설계, 요구사항 정의, 원고 검토를 수행하였음. 김동규와 이규필은 연구 목적 및 내용 정립, 기존 터널 영상취득시스템의 영상 수집 및 원고 검토를 수행하였다.

References

1. Bianchi, E., Abbott, A.L., Tokekar, P., Hebdon, M. (2021), "COCO-Bridge: structural detail data set for bridge inspections", *Journal of Computing in Civil Engineering*, Vol. 35, No. 3, 04021003.
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018), "Encoder-decoder with atrous separable convolution for semantic image segmentation", *arXiv:1802.02611*, pp. 1-23.
3. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S.O., Villena-Martinez, V., Garcia-Rodriguez, J. (2017), "A review on deep learning techniques applied to semantic segmentation", *arXiv:1704.06857*, pp. 1-22.
4. Ham, S., Bae, S., Kim, H., Lee, I., Lee, G.P., Kim, D. (2021), "Training a semantic segmentation model for cracks in the concrete lining of tunnel", *Journal of Korean Tunnelling and Underground Space Association*, Vol. 23, No. 6, pp. 549-558.
5. Hsieh, Y.A., Tsai, Y.J. (2020), "Machine learning for crack detection: review and model performance comparison", *Journal of Computing in Civil Engineering*, Vol. 34, No. 5, 04020038.
6. Kim, A.R., Kim, D., Byun, Y.S., Lee, S.W. (2018), "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, pp. 145-154.
7. Liu, Y., Yao, J., Lu, X., Xie, R., Li, L. (2019), "DeepCrack: A deep hierarchical feature learning architecture for crack segmentation", *Neurocomputing*, Vol. 338, pp. 139-153.
8. Middha, L. (2020), "Crack segmentation dataset", <https://www.kaggle.com/lakshaymiddha/crack-segmentation-dataset> (September 20, 2022).
9. MMSegmentation Contributors (2020), "OpenMMLab semantic segmentation toolbox and benchmark", <https://github.com/open-mmlab/mms Segmentation> (September 20, 2022).
10. Mundt, M., Majumder, S., Murali, S., Panetsos, P., Ramesh, V. (2019), "Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, pp. 11196-11205.
11. OurBrain (2022), "Big Eye System", OurBrain, <http://ourbrain.co.kr/> (September 20, 2022).
12. Paik, S.H., Choi, D.Y., Kim, Y.K., Jung, S.W., Kim, D.N. (2021), "Implementation of the drones with deep-learning crack detection analysis for inspection of bridge", *The Journal of Korean Institute of Information Technology*, Vol. 19, No. 3, pp. 45-52.
13. Park, G. (2022), "Building crack detection images", AI Hub, <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realml&dataSetSn=162> (September 20, 2022).
14. Scikit-image contributors (2022), "Morphological filtering", Scikit-image, https://scikit-image.org/docs/stable/auto_examples/applications/plot_morphology.html (September 20, 2022).
15. Tsai, Y.C., Chatterjee, A. (2017), "Comprehensive, quantitative crack detection algorithm performance evaluation system", *Journal of Computing in Civil Engineering*, Vol. 31, No. 5, 04017047.
16. Zhang, T.Y., Suen, C.Y. (1984), "A fast parallel algorithm for thinning digital patterns", *Communications of the ACM*, Vol. 27, No. 3, pp.236-239.