

## 콘크리트 균열 분석 결과 시각화에 관한 연구

김수민<sup>0</sup>, 손정모\*, 김도수\*

<sup>0</sup>(주)이포존 기업부설연구소,

\* (주)이포존 기업부설연구소

e-mail: ksum\_3@epozen.com<sup>0</sup>, {jmsohn, dskim}@epozen.com\*

## A Study on Visualization of Concrete Crack Analysis Results

Su-Min Kim<sup>0</sup>, Jung-Mo Sohn\*, Do-Soo Kim\*

<sup>0</sup>Epozen's research institute,

\*Epozen's research institute

### ● 요약 ●

본 연구에서는 콘크리트의 균열을 추출하여 추출한 균열을 분석하고 시각화하여 나타내는 방법을 제안한다. 추출한 균열을 분석하여 길이, 넓이, 평균 폭 등의 주요 지표를 측정하여 균열 부위에 대한 세부 정보를 파악할 수 있게 하였다. 특히 균열 분석 과정에서 기존의 균열 중심부와 에지 간의 직선 최단 거리 계산을 통한 균열 폭 측정 방식이 아닌 내접원 탐색 방식을 적용하여 다각형의 균열에 대한 폭 측정 방식을 제안하고 있다. 또한 분석 결과를 Wavefront 3D OBJ 모델과 CAD 파일로 생성하였고, 이를 웹 브라우저를 통해 입체적으로 볼 수 있도록 시각화 하였다.

**키워드:** 딥러닝(Deep Learning), 균열 검출(Crack Detection), 시각화(Visualization)

### I. Introduction

콘크리트는 다양한 구조물의 재료로 사용되고 있으며 시간이 지남에 따라 다양한 원인으로 인해 균열이 발생한다. 콘크리트 균열은 구조적 결함, 내구성 저하 등의 치명적 사고 원인이 될 수 있다[1]. 따라서 균열의 초기 발견을 위해 주기적인 외관검사를 통한 관리가 필요하다. 현재 균열의 검사는 검사자가 육안으로 직접 관측하는 외관검사법을 사용한다. 하지만 육안으로 검측하는 방식에는 인적 오류의 발생, 검사 환경의 위험성 등 한계가 존재한다. 이에 최근 [1]과 같이 기존의 문제들을 개선하기 위해 신기술을 적용해 균열 분석 프로세스의 자동화에 대한 제안 연구들이 많이 이루어지고 있다.

이에 본 연구에서는 자동화 프로세스 중 시각화 방안으로 추출된 균열 형태 및 위치를 사용해 CAD파일 생성 및 Wavefront OBJ 모델을 생성해 웹 3D 렌더링을 이용한 방안을 제안한다. 본 연구는 [3]의 후속 연구로서 해당 연구의 결과를 기반으로 제안하고 있다.

### II. Preliminaries

#### 1. Related works

##### 1.1 Dilated U-Net 기반의 이미지 복원 기법

이미지 복원 기법은 이미지 속 균열을 정상 이미지로 복원시키도록 모델을 학습시켜 복원시킨 이미지와 입력 이미지 사이의 픽셀 값의 차를 통해 균열을 추출하는 방식이다. 이미지 복원 기법은 크게 학습 데이터 생성, 이미지 복원 학습, 균열 추출로 이루어진다.

학습 데이터 생성 과정은 이미지 복원 학습용 이미지 생성을 위해 균열이 없는 정상 이미지와 균열 이미지를 합성하는 과정이다. 균열 이미지로부터 균열 부분만 추출하기 위해 균열 라벨 이미지와 균열 이미지의 픽셀들을 곱한 후 나온 결과를 사용한다. 이후 정상 이미지와 무작위로 합성해 합성 균열 이미지를 생성한다. 이때문에 균열 데이터가 적고 정상 이미지를 충분히 확보 가능한 경우 활용도가 아주 높다[3].

이미지 복원 학습 모델은 Dilated U-Net 기반으로 구성된다. 균열 추출 과정은 이미지 복원 학습을 수행 후 학습된 모델을 가지고 진행된다. 해당 모델이 입력된 이미지로부터 이미지 복원이 끝났을 때 입력 이미지와의 각 픽셀 값의 차를 구해 균열 부분을 추출한다[3]. 이미지 복원 기법은 이미지 속 균열을 정상 이미지로 복원시키도록

모델을 학습시켜 복원시킨 이미지와 입력 이미지 사이의 픽셀 값의 차를 통해 균열을 추출하는 방식이다. 이미지 복원 기법은 크게 학습 데이터 생성, 이미지 복원 학습, 균열 추출로 이루어진다.

학습 데이터 생성 과정은 이미지 복원 학습용 이미지 생성을 위해 균열이 없는 정상 이미지와 균열 이미지를 합성하는 과정이다. 균열 이미지로부터 균열 부분만 추출하기 위해 균열 라벨 이미지와 균열 이미지의 픽셀들을 곱한 후 나온 결과를 사용한다. 이후 정상 이미지와 무작위로 합성해 합성 균열 이미지를 생성한다. 이때문에 균열 데이터가 적고 정상 이미지를 충분히 확보 가능한 경우 활용도가 아주 높다[3].

이미지 복원 학습 모델은 Dilated U-Net 기반으로 구성된다. 균열 추출 과정은 이미지 복원 학습을 수행 후 학습된 모델을 가지고 진행된다. 해당 모델이 입력된 이미지로부터 이미지 복원이 끝났을 때 입력 이미지와의 각 픽셀 값의 차를 구해 균열 부분을 추출한다[3].

### 1.2 균열 정보 시각화

추출된 균열은 사용자에게 보다 시각적으로 정보를 제공하고 웹 페이지에서 3D 렌더링을 위한 객체로 만들기 위해 Wavefront OBJ 파일 형식으로 저장한다. OBJ 형식은 1990 년경 Wavefront Technologies에서 개발되었으며, 1995 년 이전에 Wavefront에서 판매 한 Advanced Visualizer 소프트웨어 버전 3 설명서의 부록 B1를 통해 공개적으로 문서화 되었다. OBJ 형식은 하나 이상의 개체 표면에 대한 3D 지오메트리를 정의하기 위한 형식으로 다각형 메쉬를 사용하거나 자유형 곡선 및 표면을 사용하여 객체 표면에 대한 지오메트리 정의를 지원한다. 점, 선, 면, 곡선, 2D 곡선 및 표면 등의 요소들이 정점으로 구성된다[4]. 저장된 OBJ 파일은 사용자 접근이 쉬운 웹 브라우저를 통해 볼 수 있게 하였다.

대부분 웹 브라우저에서 3D 객체를 렌더링하는 데 WebGL(Web Graphics Library)을 사용한다. WebGL은 플러그인을 사용하지 않고 OpenGL ES2.0 기반 API를 이용하여 브라우저의 HTML canvas에 렌더링하여 3D 웹 콘텐츠를 제작할 수 있는 API이다[5]. 그러나 WebGL을 직접 사용하는 것은 상당히 높은 학습 곡선을 요구하므로 본 연구에서는 3D 렌더링을 좀 더 쉽게 구현하기 위해 Three.js를 사용하였다. Three.js는 웹 페이지에 3D 객체를 쉽게 렌더링하도록 도와주는 자바스크립트 3D 라이브러리로 WebGL을 한 단계 감싸 놓은 Wrapper 역할을 하는 라이브러리아이다[6].

또한 추출된 균열에 대한 정보 활용을 위해 CAD 파일을 제공한다. CAD 파일은 dxf확장자의 파일을 사용한다. Dxf 포맷은 AutoCAD와 다른 CAD소프트웨어 간 파일 교환을 위한 포맷이다. Dxf 확장자 파일의 생성을 위해 ezdxf 라이브러리를 사용한다. Ezdxf 라이브러리는 파이썬에서 dxf 파일 생성 및 수정을 위한 라이브러리아이다.

## III. The Proposed Scheme

이미지 속에 균열이 있는 경우에 대해 Dilated U-Net 기반의 이미지 복원 모델을 통해 데이터 내 균열 부분만 추출한다. 추출한 균열로부터 균열의 에지를 추출하고 세선화를 통해 중심선을 그린다. 또한 추출된 균열에 대해 길이, 넓이, 평균 폭 정보를 분석한다.

각 균열 정보들의 계산은 모델이 균열을 추출한 결과에 대해 임계치를 정해 0과 1로 변환 후 이루어진다. 본 연구에서는 임계치를 0.25로 진행했다. 균열의 넓이는 변환된 데이터에서 균열 부분인 1의 개수를 사용해 측정했다. 균열 길이는 균열 이미지가 일정 크기 이상인 경우, 주변 픽셀이 연속되는지를 판단해 연속하는 개수를 측정하는 방식을 사용했다. 최종적으로 계산된 넓이와 길이를 사용해 평균 폭을 측정할 수 있다.

분석한 모든 결과는 픽셀 단위로 계산이 되어있어 카메라의 정보를 사용해 실제 값으로 환산하는 과정이 필요하다. 논문 [8]에 따르면 픽셀 상의 길이는 다음의 수식을 통해 실제 길이로 변환한다.

$$W_r = \frac{D_w}{10P_c L_f} W_p$$

$W_r$ 은 계산된 실제 cm값,  $W_p$ 는 pixel상에서의 크기,  $D_w$ 은 카메라 센서와 표면까지의 거리,  $P_c$ 는 촬영에 사용된 카메라의 ppcm(pixel per centimeter)을 의미한다[8].

기존 균열의 폭 측정의 경우 길고 평행한 균열 형태를 기준으로 균열의 에지를 추출하고 세선화를 통해 구한 중심선 각 픽셀에서 에지 픽셀까지의 가장 가까운 두 점을 구해 거리를 계산하는 방법으로 폭을 측정하였다[8]. 하지만 Fig. 1과 같이 여러 갈래로 나뉜 균열 형태에서는 어떤 점을 양 끝점으로 두고 측정해야하는지 기준이 모호하다. 특히 측정의 기준이 되는 균열 위치에 따라 균열 폭이 다르게 측정되는 경우가 발생해 객관성이 떨어지게 된다.

따라서 본 연구에서는 중심선으로부터 같은 거리에 있을 수 있는 최대 내접원을 사용해 최대 폭을 구하는 방식을 제안한다. 이는 균열의 중심부로부터 양 엣지까지의 거리를 모든 방향으로 측정 가능하기 때문에 균열 위치에 따라 바뀌는 오차들이 발생하지 않아 최대 폭에 대한 객관적인 지표가 될 수 있다.

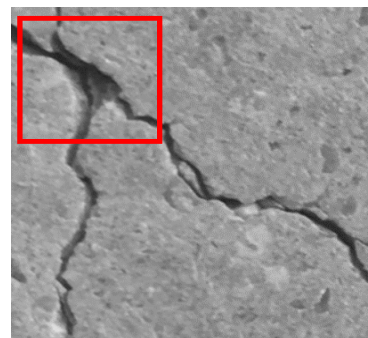


Fig. 1. 다각형 형태의 균열 사진

최대 균열 내접원을 계산하기 위해 탐색 범위는 세선점을 기준으로 하며 반지름의 최소 범위는 최소 평균 균열 폭부터 최대 추출된 균열의 모든 픽셀로 만든 정사각형의 내접원 반지름까지로 정한다. 내접원의 포함 여부는 Fig. 2와 같이 원으로 마스킹을 통해 판단한다. Fig. 2의 (가)는 원이 균열에 내접하지 않는 경우를, (나)는 원이 균열에 포함되는 경우를 보여준다. (a)는 예상 반지름 길이로 만든 원, (b)는 (a)의 크기로 지른 균열 이미지, (c)는 원으로 균열 이미지를 마스킹한 결과, (d)는 원과 마스킹 이미지를 빼 원의 포함 여부를 확인한 결과이다.

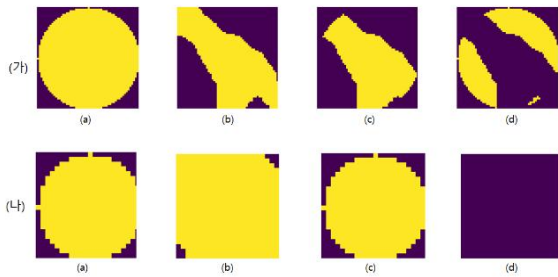


Fig. 2. 내접 여부 계산 과정

추출된 균열이미지로부터 균열의 에지와 세선화 작업을 통한 중심선, 그리고 최대 내접원 정보를 CAD파일과 OBJ 파일로 저장하였고 그 결과를 Fig. 3, Fig. 4와 같이 viewer 프로그램에서 확인하였다.

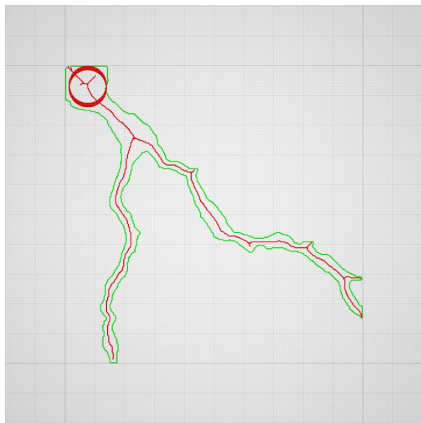


Fig. 3. OBJ 파일 확인 결과

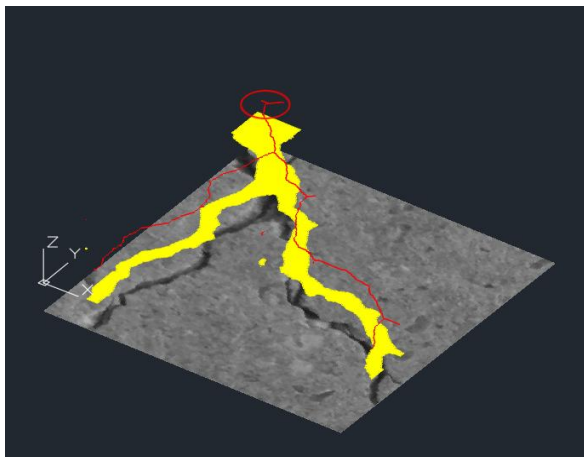


Fig. 4. CAD 파일 확인 결과

생성된 OBJ 파일과 함께 균열 이미지, 실제 균열 길이, 넓이, 평균 폭, 균열 내접원 반지름 길이의 계산 결과는 Fig. 5와 같이 웹 브라우저를 통해 시각적이고 직관적으로 확인 할 수 있다. OBJ 파일은 3D 렌더링을 통해 균열을 입체적으로 파악할 수 있도록 표시하였다.

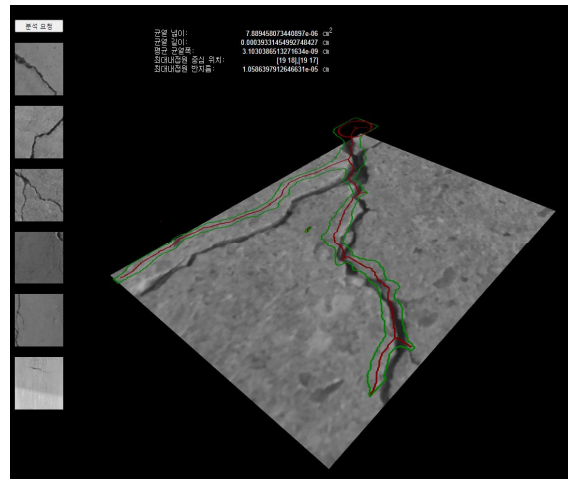


Fig. 5. 균열 분석 결과 웹 브라우저 화면

## IV. Conclusions

본 연구는 콘크리트의 균열을 추출하여 추출한 균열에 대한 정보를 분석하고 시각화 하는 방법을 제안했다. 균열 분석 과정에서 다각형 모양의 균열에 대한 최대 폭 정보를 분석하기 위해 내접원을 이용한 균열 폭 측정 방안을 제시하였다. 또한, 균열 정보는 CAD파일로 저장해 추후 활용도를 높였다. 또한 접근이 쉬운 웹 브라우저 상에서 3D 렌더링을 통해 나타내어 보다 시각적이고 입체적으로 볼 수 있도록 하였다.

## REFERENCES

- [1] Jung, Seo-Young, Yu, Jung-Ho. A Proposal of Automatic Condition Evaluation Process based on Unsupervised Learning. Proceedings of KICEM Annual Conference, pp.79-80, 2019.
- [2] "VggNet Architecture (VGG16)", tistory blog, last modified June 28.2019, accessed Feb 19.2020, <https://bskyvision.com/504>
- [3] Su-Min Kim, Jung-Mo Sohn, Do-Soo Kim. A method for concrete crack detection using U-Net based image inpainting technique. Journal of The Korea Society of Computer and Infromation. Vol.25 No. 10, 2020. DOI:10.9708/jksci.2020.25.10.035
- [4] "Wavefront OBJ File Format", Sustainability of Digital Formats: Planning for Library of Congress Collections, last modified Jan 23.2021, accessed Feb 26.2021, <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml>
- [5] "Getting started with WebGL", MDN Web Docs, last modified March 24.2019, accessed Jan 19.2021,

[https://developer.mozilla.org/ko/docs/Web/API/WebGL\\_API/Tutorial/Getting\\_started\\_with\\_WebGL](https://developer.mozilla.org/ko/docs/Web/API/WebGL_API/Tutorial/Getting_started_with_WebGL)

- [6] “Three.js library for using WebGL”, tistory blog, last modified Jan 2021, accessed Feb 2.2021, <https://darrengwon.tistory.com/397>
- [7] Çağlar Fırat Özgenel, Concrete Crack Segmentation Dataset, <https://data.mendeley.com/datasets/jwsn7tfbrp/1#file-52a39c5f-6914-4e26-88e7-ec81bfb938e>
- [8] Kim, Hyun Jun, Sim, Sung Han, Cho, Soo Jin. UAV-powered Concrete Crack Detection Technology. Magazine of the Korea Institute for Structural Maintenance and Inspection: Vol.19 Issue.1. Pages.4-7 .2015