



AI 균열 진단 (0.1mm) 및 길이 측정 수행

TEAM 22조 CRACKER
이현민, 김승빈, 유대순, 한세진



01

INTRODUCTION

1

- 1.1 팀원 소개
- 1.2 프로젝트 주제
- 1.3 배경 및 동기
- 1.4 프로젝트 목표
- 1.5 프로젝트 사용스택



02

데이터

2

- 2.1 데이터셋 소개
- 2.2 EDA



03

프로젝트 내용

3

- 3.1 Augmentation
- 3.2 Modeling
- 3.3 균열 사이즈 측정
- 3.4 추가적인 시도 사항
- 3.5 Model serving
- 3.6 시연 영상



04

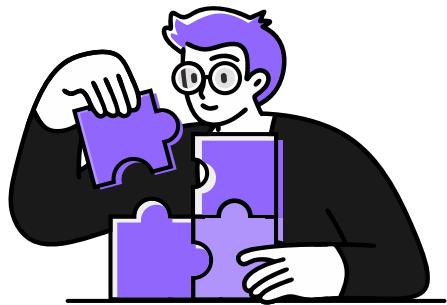
느낀점

4

- 4.1 어려웠던 점
- 4.2 개선 사항
- 4.3 느낀 점

CRACKER 팀원 소개

이현민
팀장



데이터 전처리
EDA
BaseLine 구축
HRNetV2 학습
Model Serving
발표 준비 및 결론 도출

김승빈
팀원



데이터 전처리
EDA
하이퍼파라미터 튜닝
균열 폭, 면적 측정 구현
결론 도출 및 PPT 제작

유대순
팀원



시장 조사 및 인사이트 도출
EDA
DeepLabV3+학습
하이퍼파라미터 튜닝
발표 준비 및 결론 도출

한세진
팀원



EDA
Unet++ 학습
하이퍼파라미터튜닝
균열 폭, 면적 시각화
발표 준비 및 결론 도출

프로젝트 주제 소개

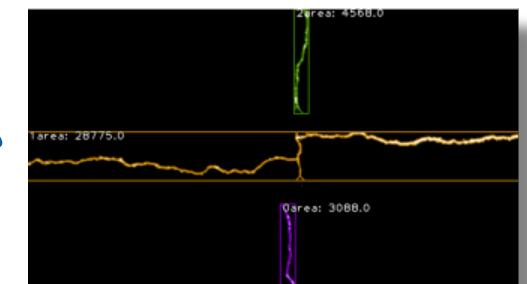
드론이 촬영한 영상을 분석하여 0.1mm 수준의 균열을 진단하고,
픽셀 단위로 면적 및 폭을 측정하는 모델을 만든다.



원본 영상



균열 진단



각 균열의 픽셀수 측정



균열의 최대 폭 측정

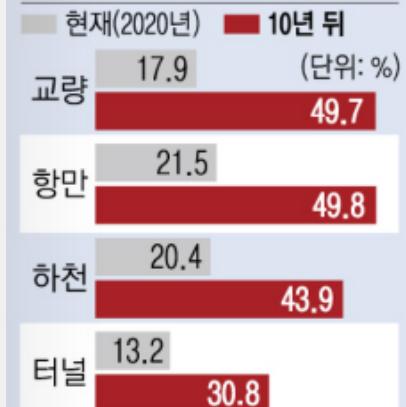
프로젝트 기획 의도 및 동기



점점 노후화되는 국내 인프라 시설

국내 시설물 노후화율*

*지어진 지 30년 이상 된 시설물 비율



〈자료: 국토교통부〉

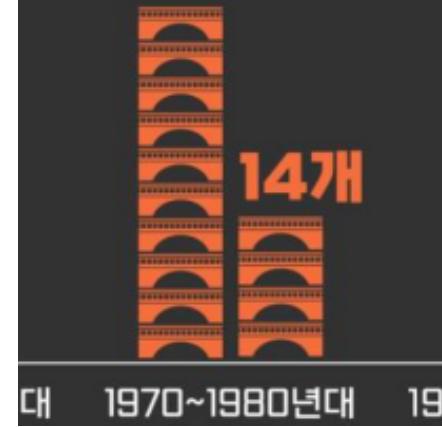
출처: <https://www.seoul.co.kr/news/newsView.php?id=20220210008009>

국내 주요 인프라 시설: 16만 381개

30년 이상 된 시설물: 2만 7997개

한강다리 건설시기

출처: 서울연구원



출처: http://tbs.seoul.kr/news/newsView.do?seq_800=20469504&typ_800=7

한강다리 건설시기

1900~1960년대: 5개

1970~1980년대: 14개

1990~2000년대: 9개

2020년 11월 기준

국내 주요 인프라 시설 5개 중 1개가 노후 시설로,
10년 뒤 전체의 **44.2%**가 30년을 넘은 노후 시설물이 된다.

프로젝트 기획 의도 및 동기



인력 사용으로 인한 안전문제

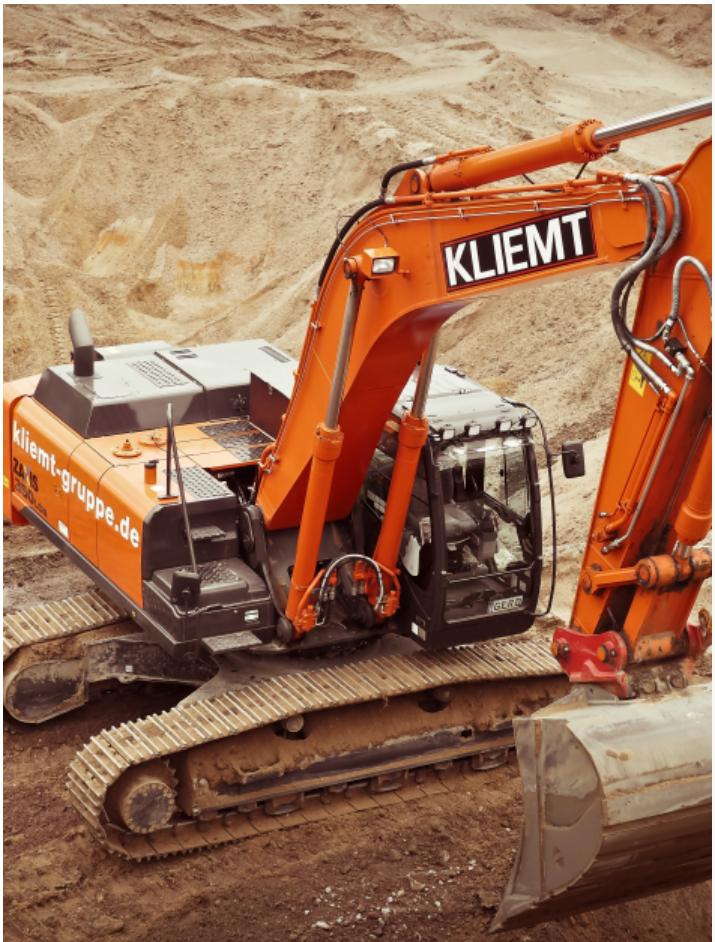
고용노동부 2020년 산업재해 업종별 사고사망 통계

(단위: 명, %, %p)

구분	계	떨어짐		부딪힘	물체에 맞음		끼임	깔림, 뒤집힘		기타			
		비중	비중		비중	비중		비중	비중				
2020년	87	66	75.9	7	8.0	3	3.4	4	4.6	1	1.1	6	6.9
2019년	54	46	85.2	5	9.3	1	1.9	1	1.9	0	0.0	1	1.9
증 감	33	20	-9.3	2	-1.3	2	1.5	3	2.7	1	1.1	5	5.0

“보수공사의 사고사망자 대부분은 ‘떨어짐’으로 인해 발생했으며, 전년대비 ‘보수공사 중 떨어짐 사망사고’가 크게 증가했다.”

프로젝트 기획 의도 및 동기



중장비 사용, 인력 고용에 의한
고가 비용

굴절차 대여 비용 없음

굴절차 1회 대여에 약 100만원 가량의
높은 비용이 발생하는데,
드론을 이용한 균열 진단은 드론 초기 구매 비용만 소요됨

투입 인력 대폭 감소

교량 안전 진단 시 평균 11명의 인력이 필요해
10개 교량을 기준으로 평균 **4천만원**의 진단 비용이 발생했는데,
드론을 통한 균열 진단으로 드론을 조종하는 인력 1명만 필요

프로젝트 기획 의도 및 동기



육안 점검에 의한 한계

육안 점검

사고 직전 정기안전점검에서
송정1교는 양호(B등급),
성황교는 우수(A등급)

BUT,

지난 2020년 9월 태풍 마이삭 때
강원 평창군의 송정1교와 삼척시의 성황교가
붕괴되는 사고 발생

프로젝트 목표

06



비용, 시간, 안전 문제를 해결하기 위한 AI 모델 만들기



기업의 개발 목표 점수 달성



균열 사이즈 측정 알고리즘 개발



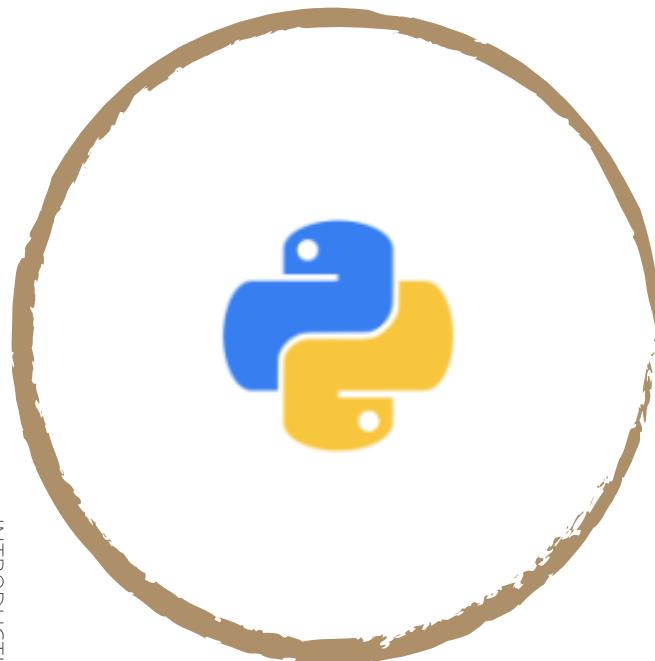
Parameters	DeepCrack	개발 목표
Accuracy	0.9930	0.9980
Precision	0.3384	0.8000
Recall	0.7015	0.9000
F1 Score	0.4564	0.8000

기업의 현재 모델 점수(DeepCrack)와 개발 목표

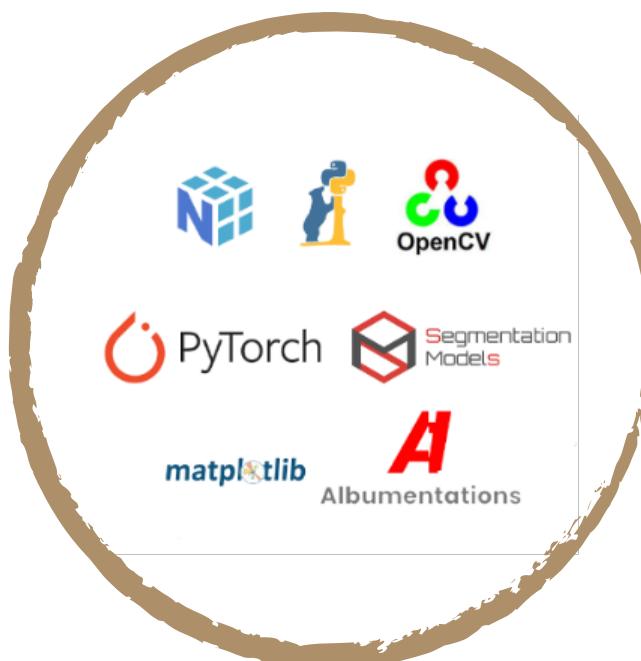
개발 환경 및 기술 스택

07

LANGUAGE



LIBRARY



TOOL



데이터셋 소개

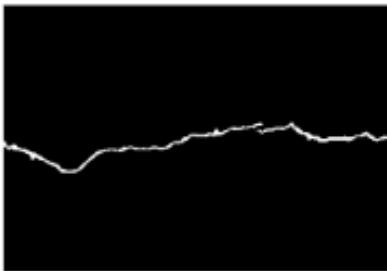
Training set

총 635장

Crack500 (width: 2560, height: 1440)

CrackForest (width: 480, height: 320)

DeepCrack (width: 544, height: 384)



데이터

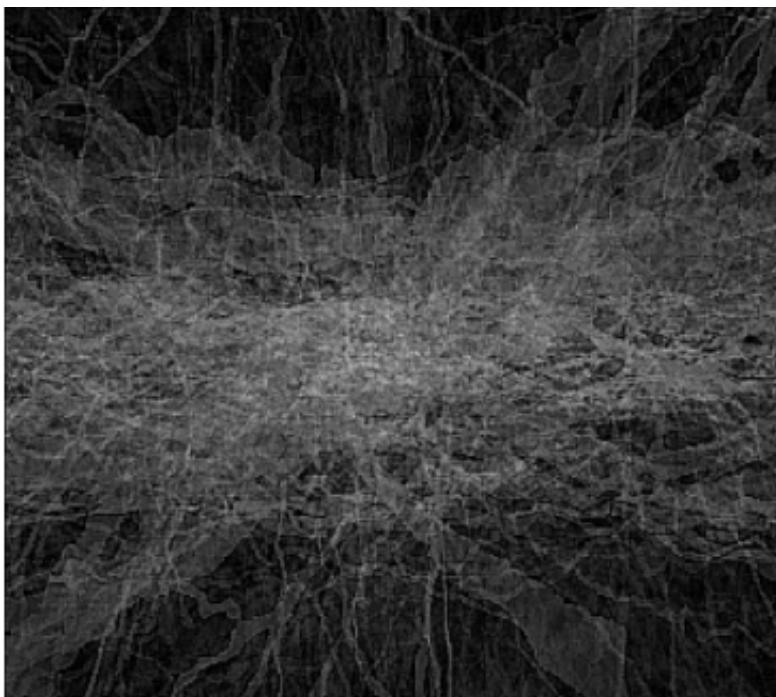
Test set

총 115장

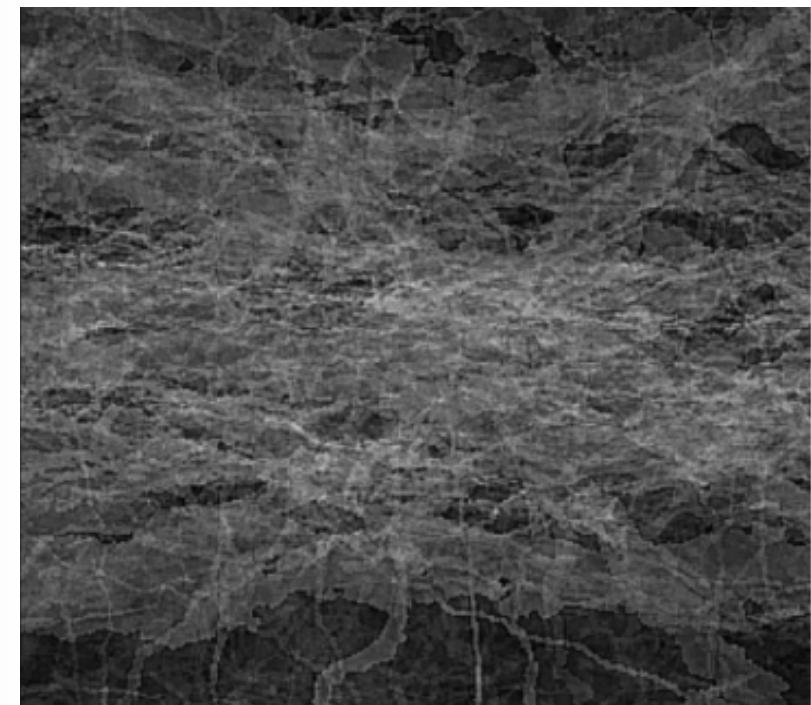
Crack500 (width: 2560, height: 1440)



EDA



Original Mask



Transformed mask

Alpha blending

Original mask의 균열 분포와 Augmentation 후

Transformed mask의 균열 분포 확인

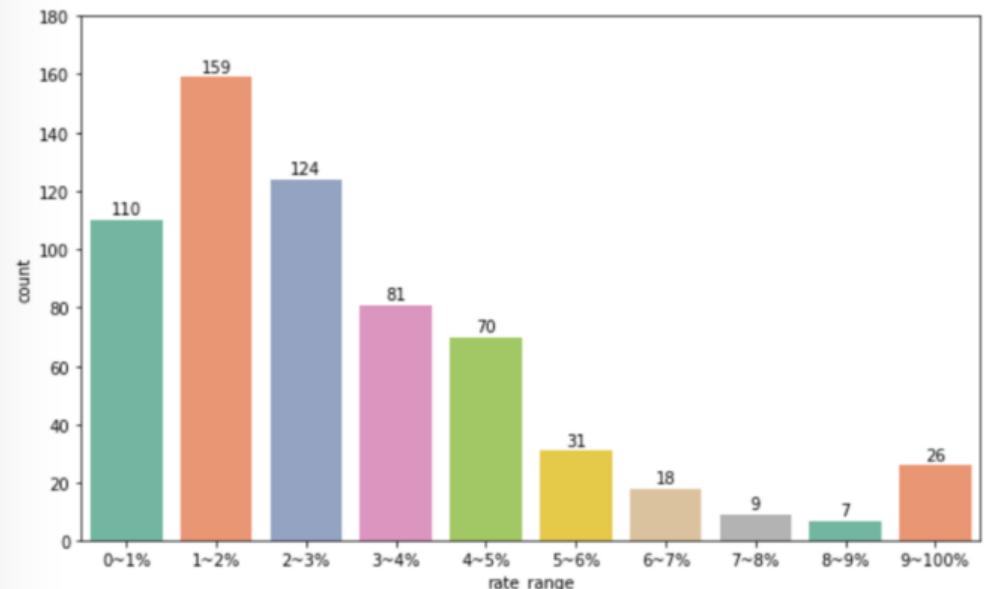
EDA

```
df_rate['rate'].describe()

...
count      635.000000
mean       3.087008
std        2.733552
min        0.260000
25%       1.245000
50%       2.320000
75%       4.050000
max       19.250000
Name: rate, dtype: float64
...
```

Target ratio

사진의 전체 픽셀 중 CRACK부분의 픽셀 비율 확인



Augmentation

Augmentations list

HorizontalFlip
VerticalFlip
ShiftScaleRotate
RandomRain
RandomBrightnessContrast
RandomCrop
CLAHE



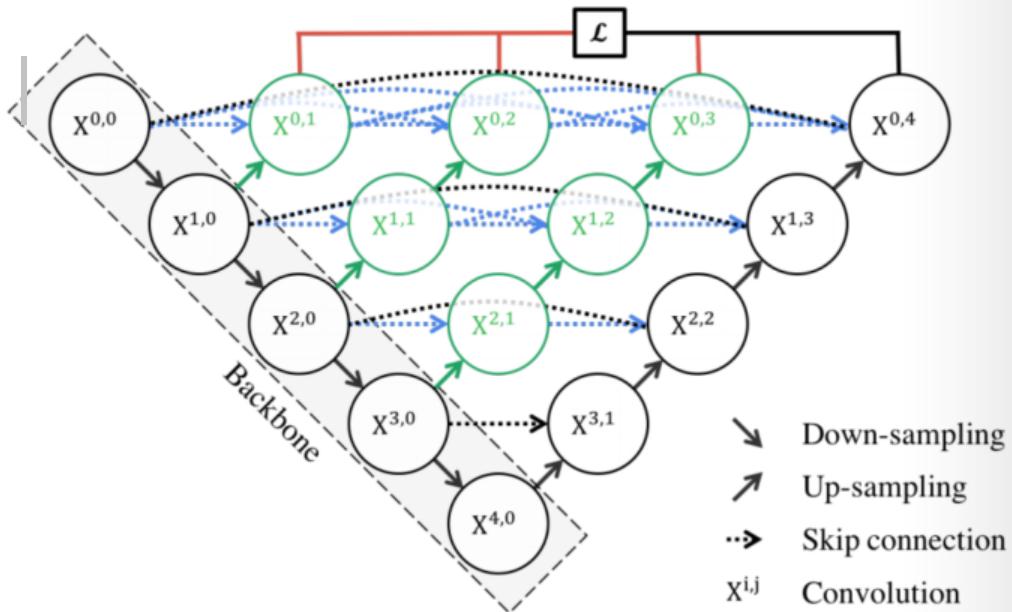
Augmentations setting

1. RandomRain만 사용
2. RandomBrightnessContrast만 사용
3. CLAHE만 사용
4. ShiftScaleRotate만 사용
5. CLAHE만 제외
6. RandomRain에서 blur_value 2로 변경



Modeling - 사용한 모델

① UNet++



UNet++ 구조

(<https://arxiv.org/pdf/1807.10165.pdf>)

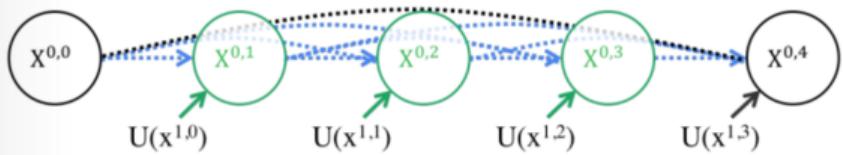
UNet++의 특징

- Semantic segmentation task에 특화된 모델
- Re-Designed skip pathway: skip connectiono| dense convolution bolck으로 이루어져 있어 Encoder-Decoder의 Feature map 간의 Semantic Gap을 더 줄일 수 있다.



UNet의 skip-pathway
(<https://arxiv.org/pdf/1505.04597.pdf>)

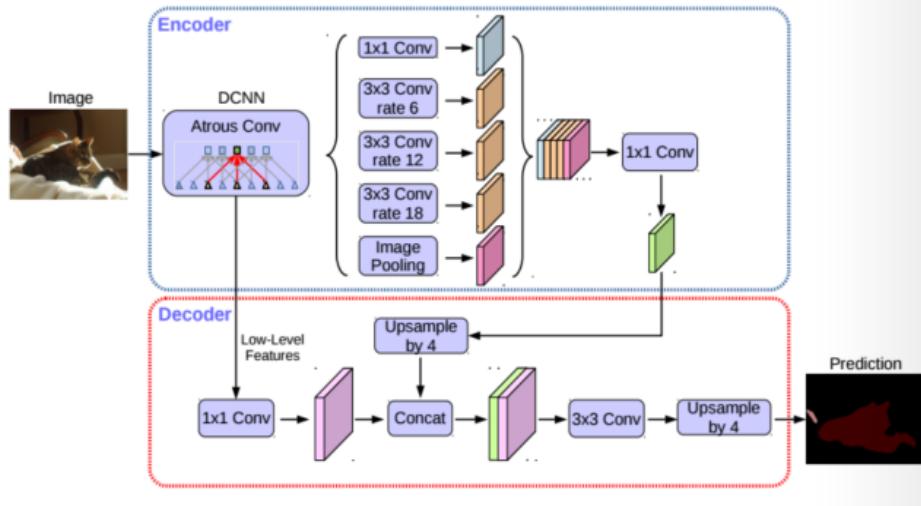
$$x^{0,1} = H[x^{0,0}, U(x^{1,0})] \quad x^{0,2} = H[x^{0,0}, x^{0,1}, U(x^{1,1})] \quad x^{0,3} = H[x^{0,0}, x^{0,1}, x^{0,2}, U(x^{1,2})]$$



UNet++의 skip-pathway
(<https://arxiv.org/pdf/1807.10165.pdf>)

Modeling - 사용한 모델

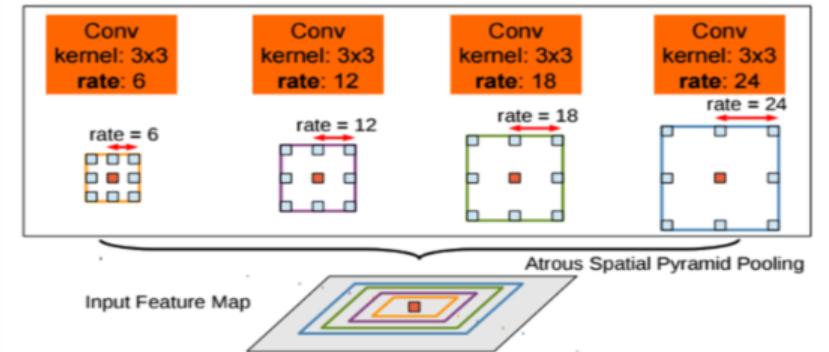
② DeepLabV3+



DeepLabV3+ 구조
(<https://arxiv.org/pdf/1802.02611.pdf>)

DeepLabV3+의 특징

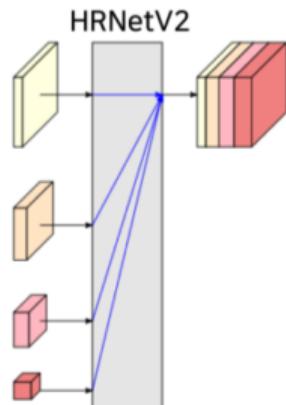
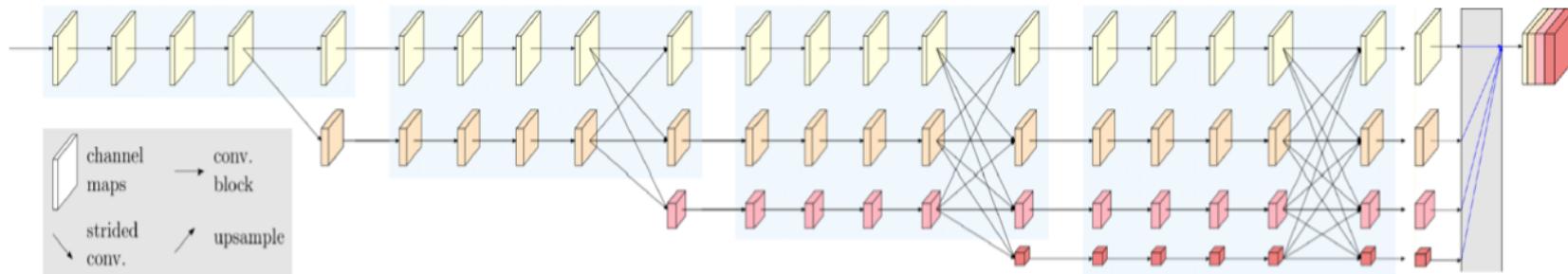
- Atrous Spatial Pyramid Pooling: 파라미터의 수를 늘리지 않고 receptive field 영역이 커지는 효과를 줄 수 있어 한 픽셀이 볼 수 있는 영역이 커진다.
- encoder-decoder: dilated 된 피쳐가 없기 때문에 encoder path가 빠르고 decoder 단에서 점진적으로 크기를 복원하기 때문에 디테일을 잘 잡을 수 있다.



ASPP 예시
(<https://arxiv.org/pdf/1606.00915v2.pdf>)

Modeling - 사용한 모델

③ HRNetV2



HRNetV2의 concat
[\(https://arxiv.org/pdf/1908.07919.pdf\)](https://arxiv.org/pdf/1908.07919.pdf)

HRNetV2의 특징

- Semantic segmentation task에 특화된 모델
- High resolution의 feature 유지: High-to-low representation을 모두 합침으로써 high resolution, low resolution을 parallel하게 유지해준다.

Modeling - 모델 학습



Encoder

- Efficientnet(b0~b7)
- Resnet34, Resnet50
- HRNetV2



Model

- Unet++
- DeepLabV3+



하이퍼파라미터

- Optimizer: adam, adamp, adamw
- Scheduler: reducelr, cosineanneal
- Learning rate: 0.01, 0.001
- Epoch: 30, 50, 100
- Batch size: 8, 16
- Loss function: CELoss, BCEWithLogitLoss
- threshold: 0.4, 0.5, 0.6



기타

- 평가지표: torchmetrics(f1-score, precision, recall)
- 실험결과 관리: wandb
- Cross validation: Kfold 5

Modeling - 모델 학습



Encoder

- Efficientnet(b0~b7)
- Resnet34, Resnet50
- HRNetV2



- ## Model
- Unet++
 - DeepLabV3+



하이퍼파라미터

- Optimizer: adam, adamp, adamw
- Scheduler: reducelr, cosineanneal
- Learning rate: 0.01, 0.001
- Epoch: 30, 50, 100
- Batch size: 8, 16
- Loss function: CELoss, BCEWithLogitLoss
- threshold: 0.4, 0.5, 0.6



기타

- 평가지표: torchmetrics(f1-score, precision, recall)
- 실험결과 관리: wandb
- Cross validation: Kfold 5

Modeling - 성능

DeepLabV3+



Unet++

test/f1



test/recall



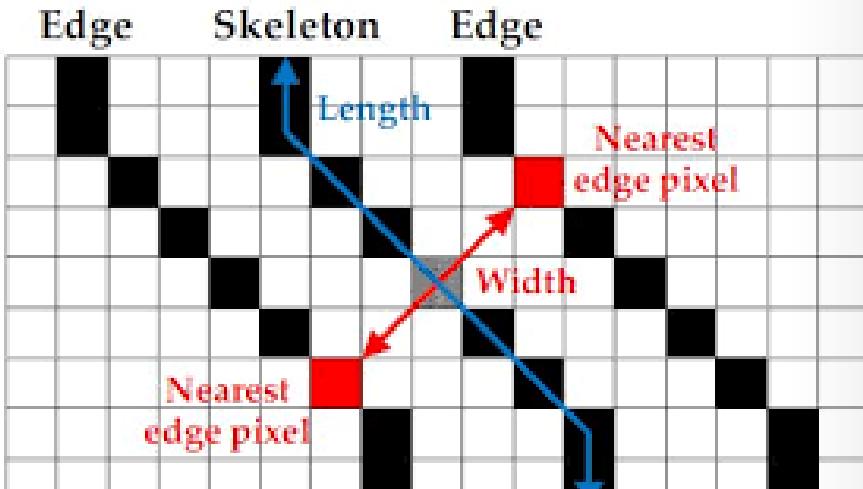
DeepLabV3+ 성능

Parameters	Validation	Test
Accuracy	0.990	0.983
Loss	0.028	0.041
Precision	0.836	0.650
Recall	0.876	0.780
F1 score	0.850	0.704

균열 사이즈 측정: 최대 폭

18

skeletonize + canny edge detection
통한 균열의 폭 측정



출처: https://github.com/Garamda/Concrete_Crack_Detection_and_Analysis_SW



skeletonize

boundary를 erode연산으로 반복적으로 깎아내어
중간 뼈대 부분 1pixel만 남김



original



skeleton



canny edge detection

다단계 알고리즘을 사용하여
이미지의 광범위한 에지를 검출



original



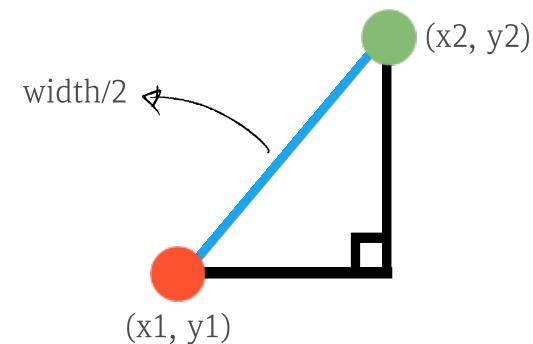
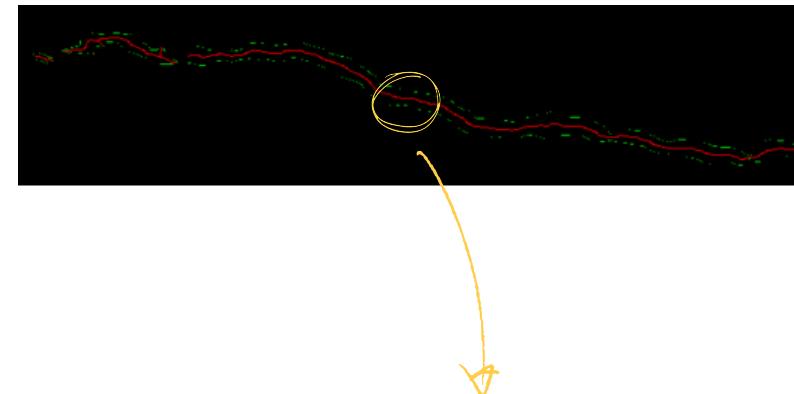
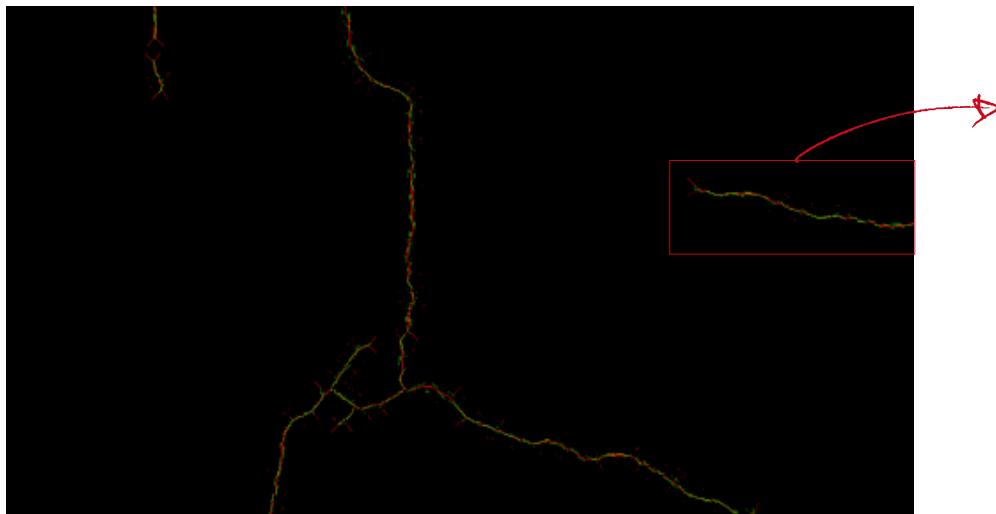
canny

균열 사이즈 측정: 최대 폭

19



kleton 상의 좌표와 최단 거리에 위치한 canny edge 상의 좌표를 갖고
피타고라스의 정리를 이용하여 두 좌표간 거리 계산

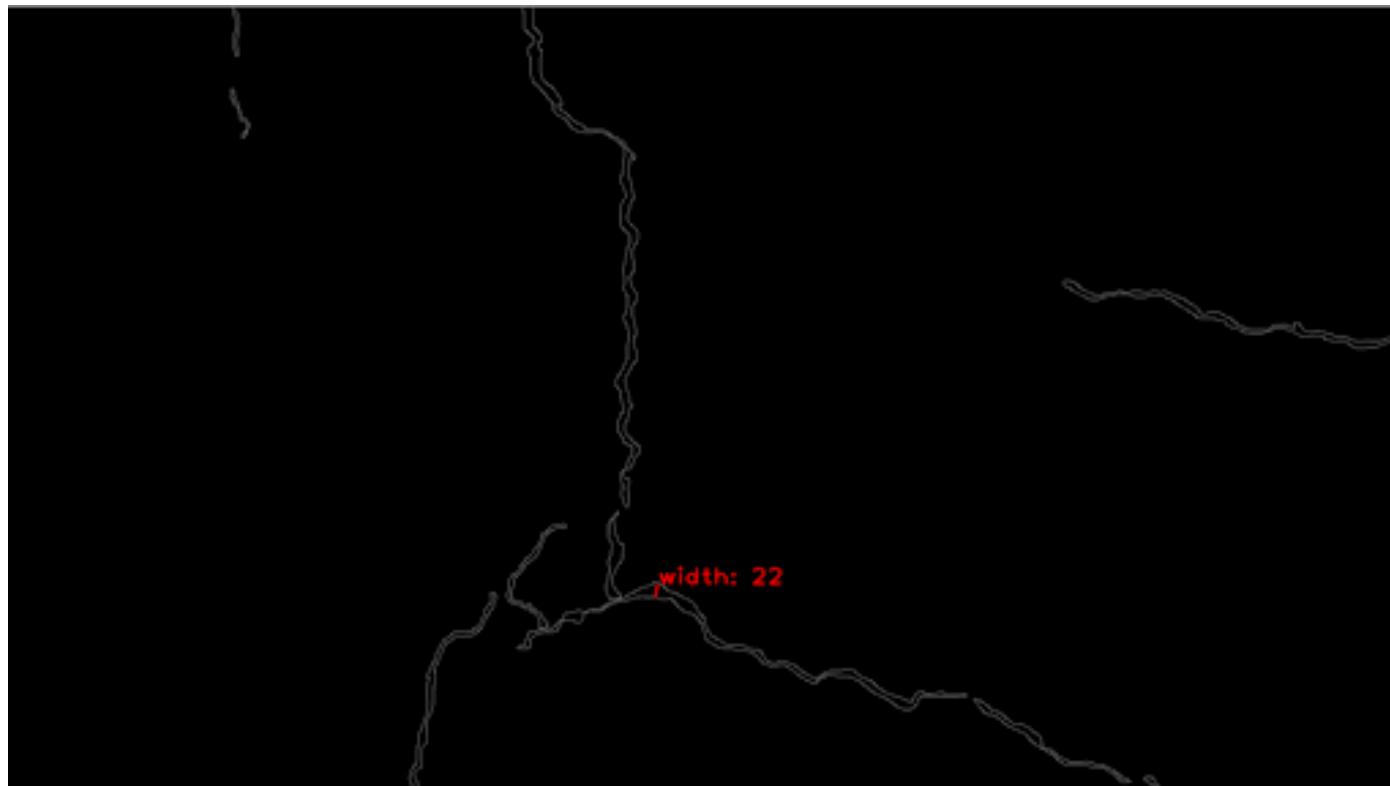


균열 사이즈 측정: 최대 폭

20



피타고라스 정리를 이용해 나온 width중 최대값을 구하고
해당 값에 x2를 하여 최대 폭의 픽셀 수를 나타내준다.

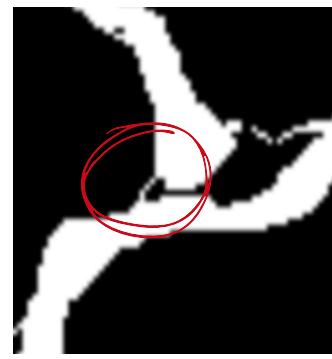
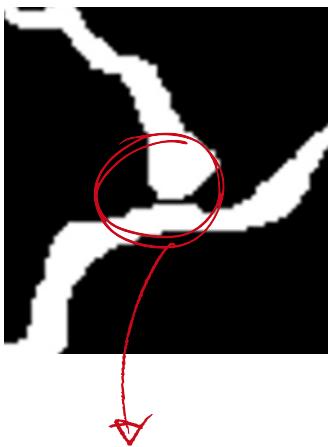


균열 사이즈 측정: 균열 면적

21

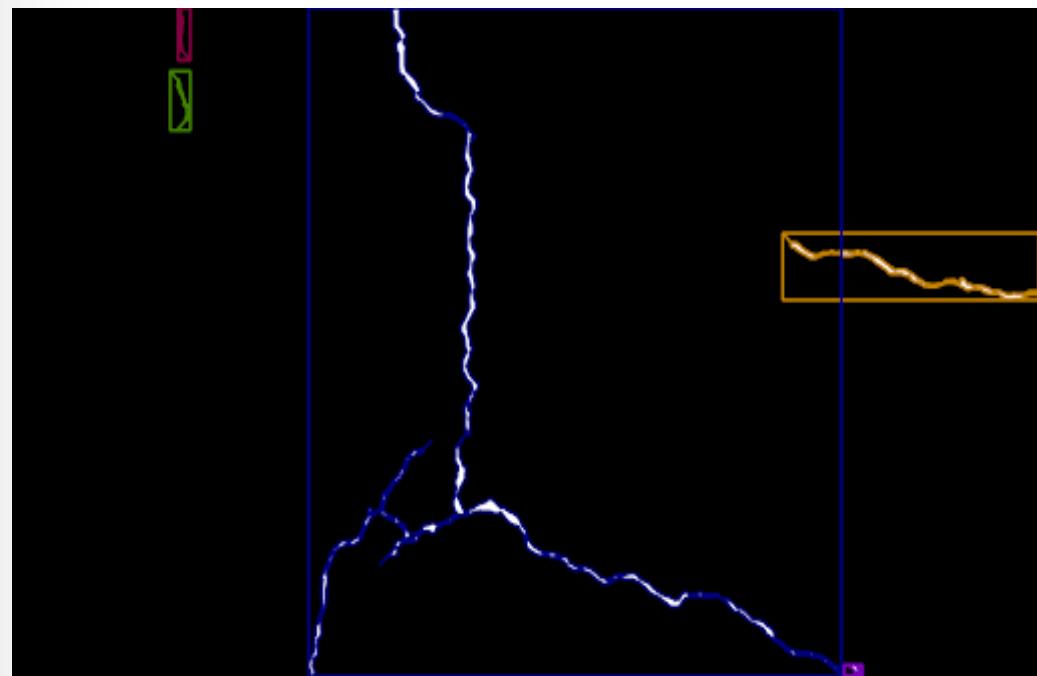
morphology(dilate/erode) + contour
통한 각 균열의 픽셀수 측정

skeletonize를 할 때 dilation 연산을 반복해서 떨어진 부분을 붙여주고 erode를 해주어 skeleton을 구한다. 그리고 기존의 마스크와 or연산을 통해 균열을 최대한 폐곡선으로 만들어준다



약간 떨어져 있어서 contour box가 두 개 생성된다

contourArea로 각 contour box 내부에 있는
균열의 픽셀 수 측정



추가적인 시도 사항



Label이미지에 Gaussian filter 적용

1) Gaussian filter 장점

미세한 균열도 잡아내어 안전 진단에 이점을 높힌다.

2) Gaussian filter 적용

- ① 안전 진단에 있어 recall이 더 중요한 지표라고 판단
- ② target인 crack이 얕기 때문에 crack을 선명하게 하는 dilate + Gaussian filter 적용 필요



원본 Label 이미지



dilate 적용



Gaussian filter 적용

3) 기업 측의 피드백

Gaussian filter 적용시킨다면 recall 값은 올라가겠지만, 실제 이미지에 모델을 적용할 때 label이 없기 때문에, 원본 사진 속 균열만 증폭시키는 부분에서 이슈가 생길 수 있다. 때문에 실제 inference에서는 성능 차이가 많이 날 수 있다.

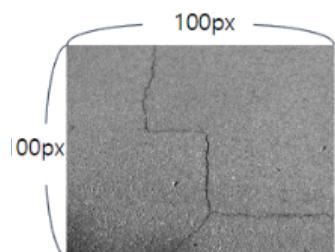
추가적인 시도 사항



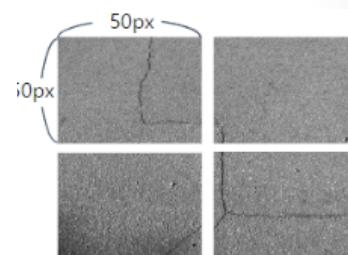
Cropping

원본 이미지를 모델 적합 사이즈로 자른 후에 균열 진단 후 병합

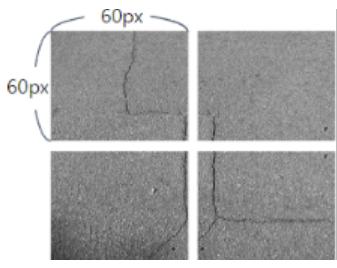
→ 정보 손실을 최소화하기 위해 resize하는 대신 일정한 크기로 이미지를 crop하여 추론한다.



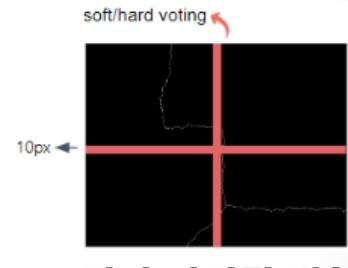
원본 이미지



일반 cropping



10% 겹치게 crop

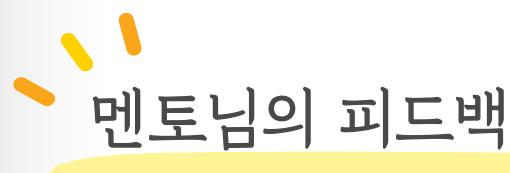


결과 이미지 병합

BUT.

crop 후 evaluation metric 결과값이 감소하는 경향을 보임

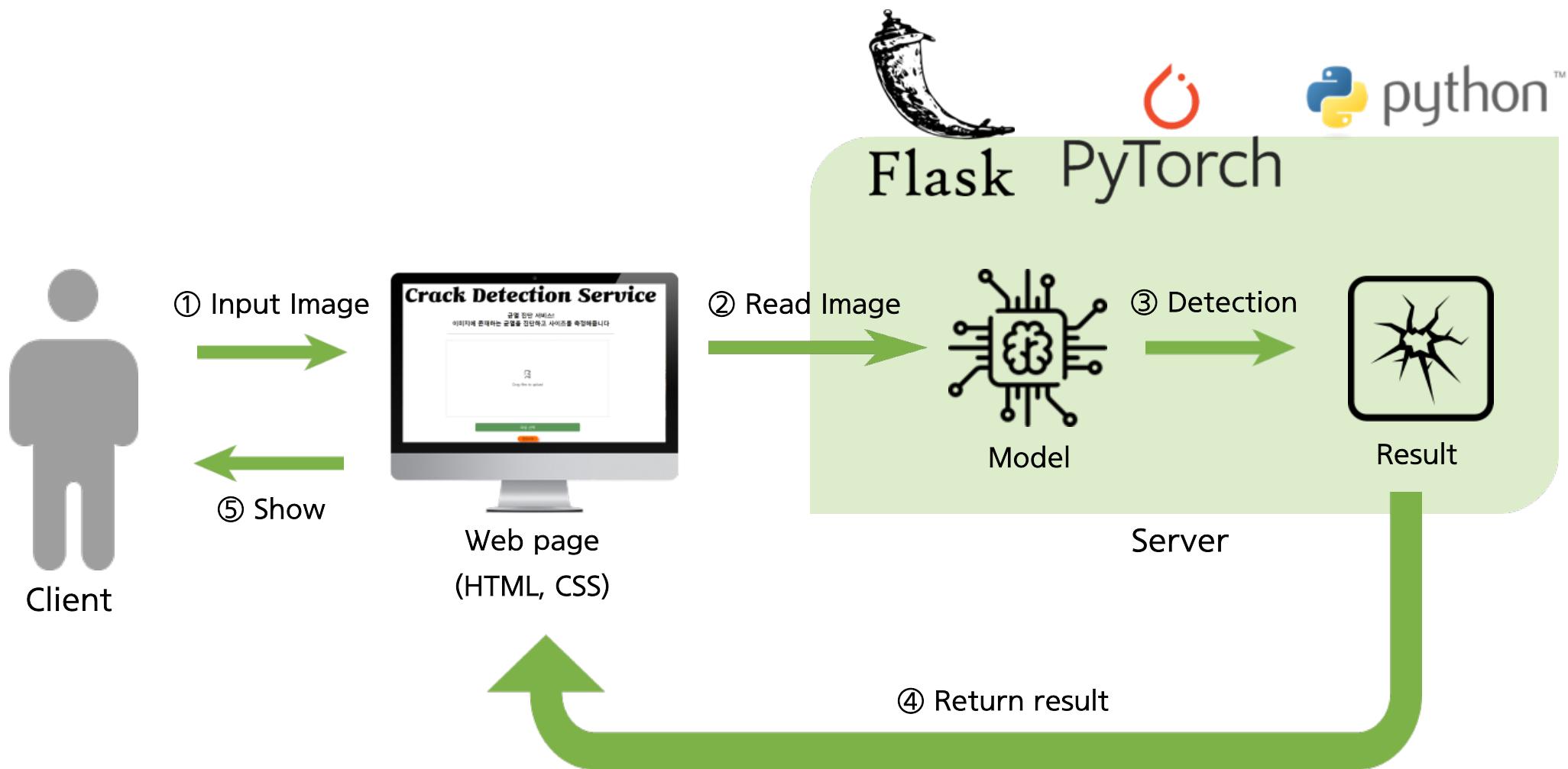
		recall	precision	f1 score
image1	non-crop	0.7529	0.737	0.7449
	crop	0.7596	0.7271	0.743
image2	non-crop	0.7788	0.886	0.8289
	crop	0.7661	0.8874	0.8223
image3	non-crop	0.7859	0.7029	0.7421
	crop	0.7485	0.7325	0.7404



멘토님의 피드백

convolution이 포함된 모델은 픽셀의 절대적 크기에 영향을 받는다.
GPU에 안 올라갈 정도로 큰 이미지는 crop해야 하지만,
현재 데이터셋의 이미지 사이즈 정도는
crop 여부가 결과에 영향을 미치지 않는다.

Model Serving

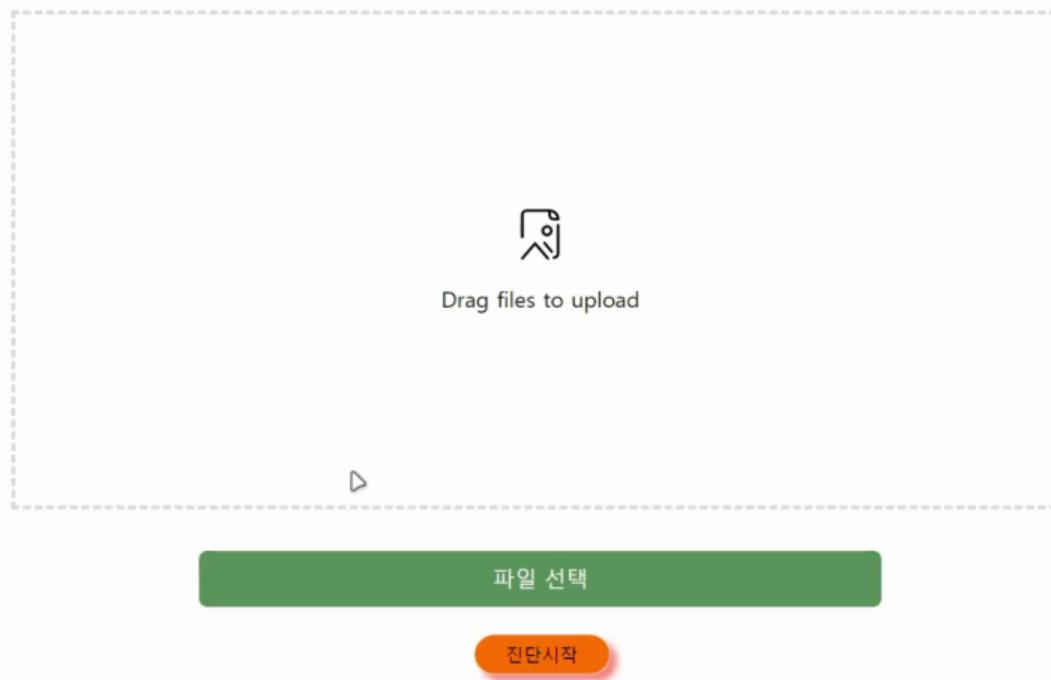


시연 영상

25

Crack Detection Service

균열 진단 서비스!
이미지에 존재하는 균열을 진단하고 사이즈를 측정해줍니다



어려운 점

26



GPU의 한계

- efficientnet-b7과 같이 무거운 모델 학습 어려움
- 하이퍼파라미터 튜닝 등 반복적인 실험을 하기 어려움



Target이 작은 이미지에서의 Semantic Segmentation

- target(건물 및 교량의 균열)이 작아 segmentation model이 과적합될 가능성이 커짐
- 실제 현장의 이미지에는 페인트 벗겨짐 등 균열과 비슷한 부분이 존재하는데 모델이 이런 부분까지 crack으로 분류해내는 문제 발생

느낀점

개선할 점

1. 모델 서빙

- 현재는 flask로 model serve 구현
- torch serve 이용하면 최적화된 서비스 제공 가능할 것

2. Inference 속도 개선

skeletonize과 canny edge detection으로 균열의 폭을 구할 때 이중 for문의 사용으로 시간이 많이 소요됨

3. 모델 정확도 개선

- 페인트 벗겨짐 등 균열과 유사하게 보이는 부분을 segmentation 모델만으로는 정확하게 분류하기 어려움
- object detection model을 추가로 생성하여 two step으로 균열 진단을 한다면 보다 정교한 균열 진단이 가능할 것

느낀점

이현민

같은 목표를 갖고 함께 나아가는 팀원들의 소중함을 느끼게 되었고, 매일 회의를 진행하며 활발한 의사소통이 중요하다는 것을 깨달았습니다.

작은 target을 검출해내는 segmentation의 여러 기법에 대해 공부할 수 있어 좋았고, Inference 속도 개선과 object detection 모델을 추가해서 지금보다 정확하고 최적화된 serving을 구현해 보아야겠다는 생각이 들었습니다.



김승빈

skeleton을 중심으로 양쪽에 있는 canny edge 상의 점을 잡아낼 수 있는 알고리즘을 떠올리지 못한 부분이 아쉽습니다.

프로젝트를 통해 수학적 기본기와 CS 지식을 갖춰, task에 맞는 최적의 알고리즘을 구상하고 이를 구현해내고 싶다는 목표를 갖게 되었습니다.

함께 해주신 팀원 분들께 감사합니다!

느낀 점

28



유대순

전문지식 필요하다고 생각이 든다. 모델서빙을 배울 필요성을 느꼈고, 최적화가 조금 떨어지지만 조금씩 개선해가면서 쓸수 있다고 생각했다. 논문의 중요성과 검색하면서 모르던 기법들을 습득. 폭을 구하는 코드는 있지만 정확하진 않았고 그거에 대해서 수학적으로 좀더 이해를 잘했더라면 현재 나와있는 기술보다 좋은 결과가 나올수있다고 본다.

팀원과 함께하다보면 혼자서는 몇번을 봐도 이해가 안되던 것을 집단 지성으로 해결할 수 있어서 좋았습니다.

함께 해주신 팀원분들 감사합니다.

느낀점



한세진

이번 프로젝트를 통해 스스로의 역량이 부족함을 많이 느꼈습니다. 체계적인 공부 및 정리의 필요성을 체감했습니다. 프로젝트를 진행하면서 문제 해결을 위한 다양한 시각과 방법이 존재함을 깨달았으며 협업의 중요성 또한 다시 한 번 체감했습니다. 이번 프로젝트를 함께 해준 팀원 분들과 많은 도움을 주신 멘토님께 감사 인사를 전합니다.

Q & A



Thank you for watching

