

Crack Segmentation

결과 보고

실험 순서

데이터 증강 실험

데이터에 알맞는
데이터 증강 기법 탐색

모델 실험

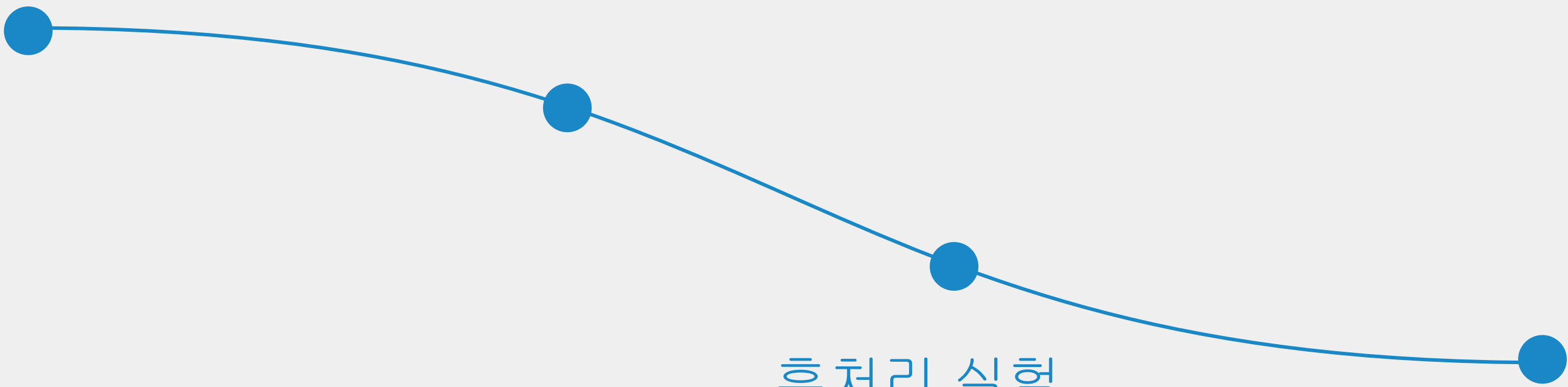
1 Class 모델, 2 Class 모델,
Dilation 데이터셋, Fine-tuning 실험 등

후처리 실험

Segmentation 모델의 결과물에
후처리를 통한 성능 향상 실험

배경 제거 실험

Segmentation 기반의
배경 제거 모델 개발



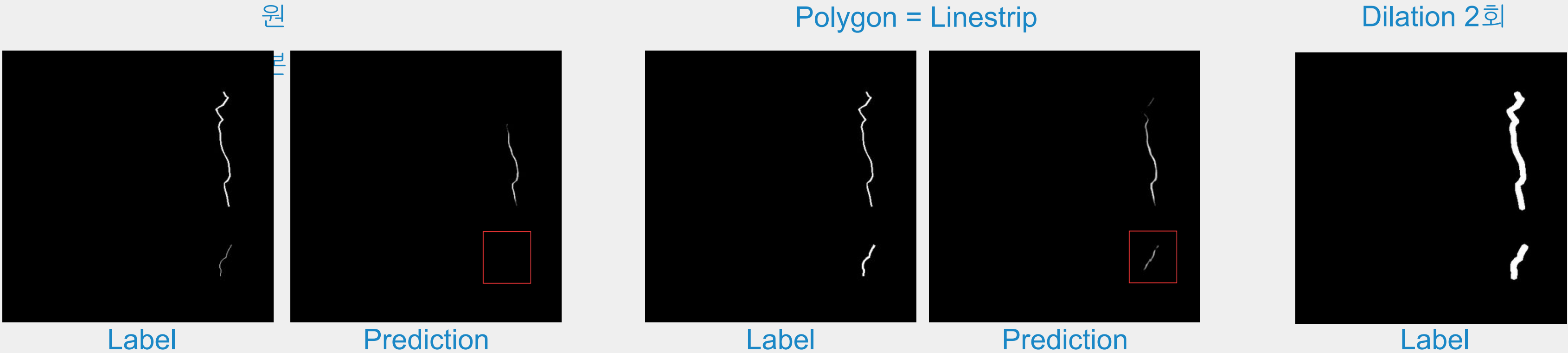
데이터 증강 실험

모델	적용 기법	성능 (max mIoU)	비고 (Baseline 대비)
SCSegamba	증강 없음	0.639	-
SCSegamba	Horizontal, Vertical Flip	0.641	+0.002
SCSegamba	Rotate	0.639	+0.000
SCSegamba	Elastic Transform	0.639	+0.000
SCSegamba	GridDistortion	0.639	+0.000
SCSegamba	GridShuffle	0.641	+0.002
SCSegamba	CLAHE	0.64	+0.001
SCSegamba	Random Brightness Contrast	0.637	-0.002
SCSegamba	Horizontal Flip + Vertical Flip + GridShuffle + CLAHE	0.645	+0.006

균열 쿼기 실험

모델	균열 쿼기	성능(max mIoU)
SCSegamba	원본	0.620
SCSegamba	Polygon = Linestrip	0.639 (+0.019)
SCSegamba	Dilation 2회	학습 예정

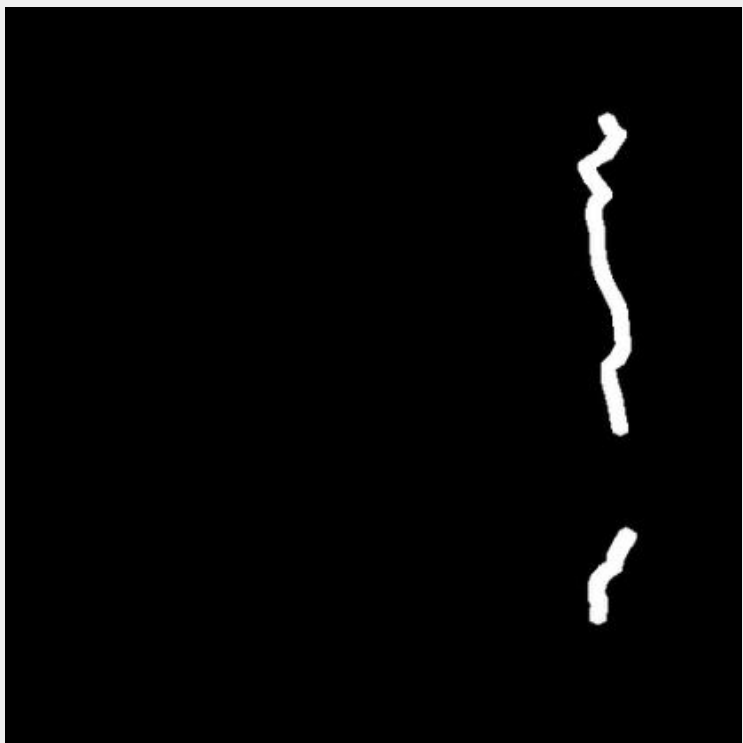
* Epoch: 30, LR: 5e-5



균열 쿼기 실험

◦Dilation 데이터셋 학습

Dilation 데이터셋을 이용해 학습 후 성능과 결과 확인



Label



Prediction

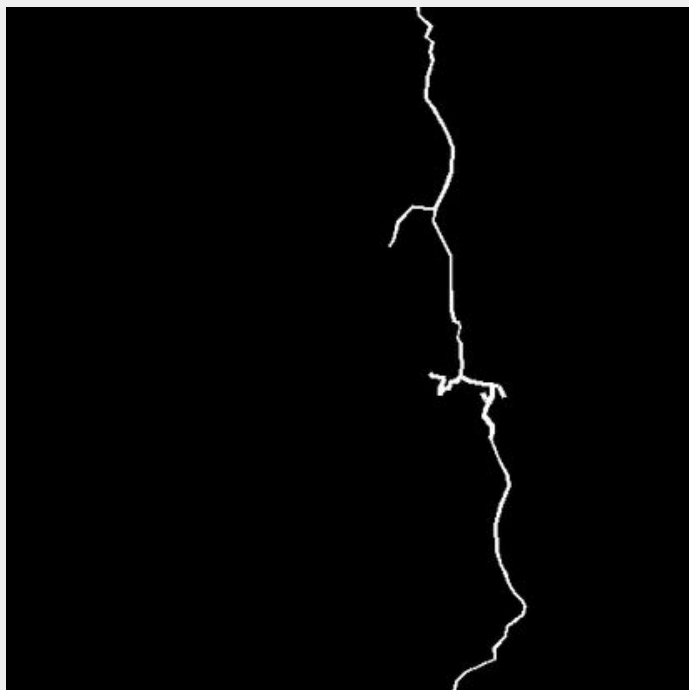
모델	Test Dataset	성능(max mIoU)	비고
SCSegamba	Dilation	0.716	Dilation Test 시 높은 성능을 기록
SCSegamba	polygon = linestrip	0.589 (-0.05)	성능이 크게 감소

초기 polygon = linestrip 실험 결과 성능 (0.639 대비)

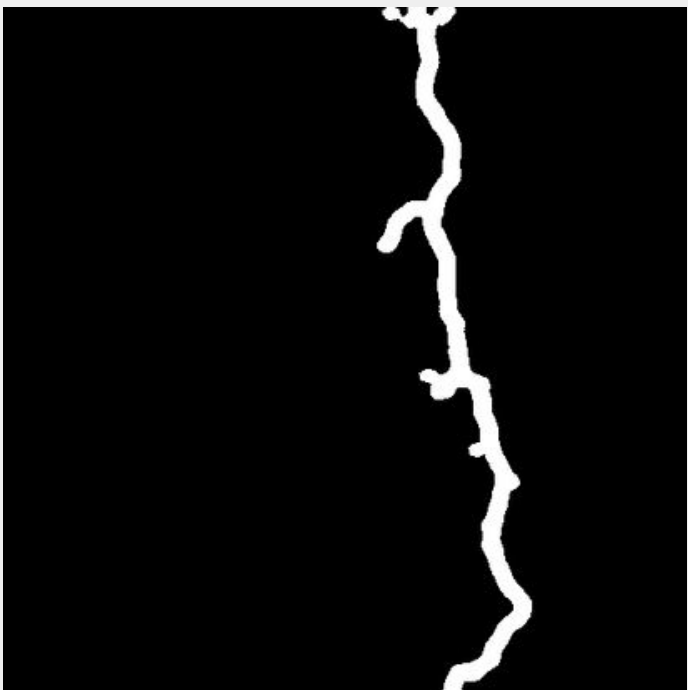
Fine-tuning 실험

◦Dilation 학습 모델 Fine-tuning

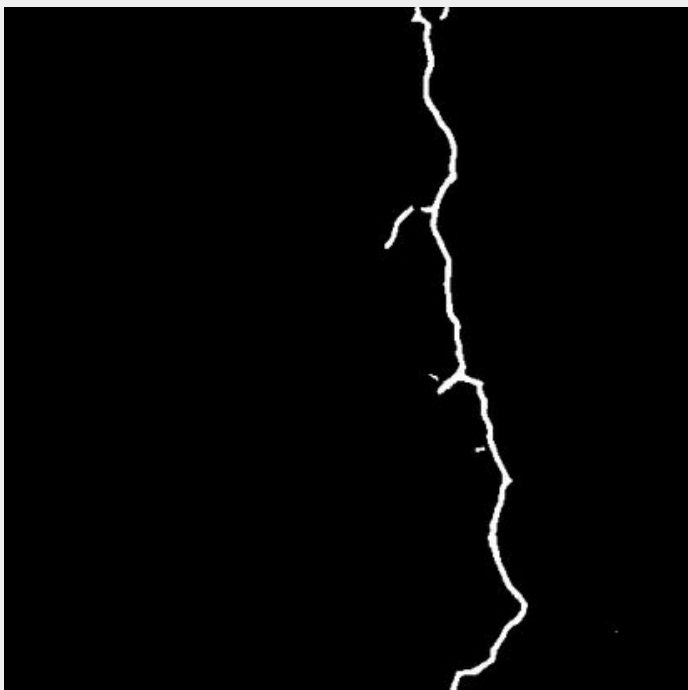
Dilation 데이터셋 학습 모델을 그대로 사용한다면 실제 Crack보다 더 크게 마스킹 되는 문제 발생
이를 해결하기위해 얇은(polygon=linestrip) 데이터셋으로 Fine-tuning



Label



Dilation Prediction



Fine-tuning Prediction

모델	Test Dataset	성능(max mIoU)	비고
SCSegamba	polygon = linestrip	0.589 (-0.05)	성능이 크게 감소
SCSegamba	polygon = linestrip	0.666 (+0.027)	Dilation 데이터셋을 학습한 모델을 Fine tuning

초기 polygon = linestrip 실험 결과 성능 (0.639 대비)

Multi-Class 실험

기존 균열(1 Class)을 탐지하는 방법
→ 배경, 균열 (2 Class)을 탐지하도록 모델 구조 수정

기존 bce + dice loss 함수
→ **CrossEntropyLoss** 함수로 변경

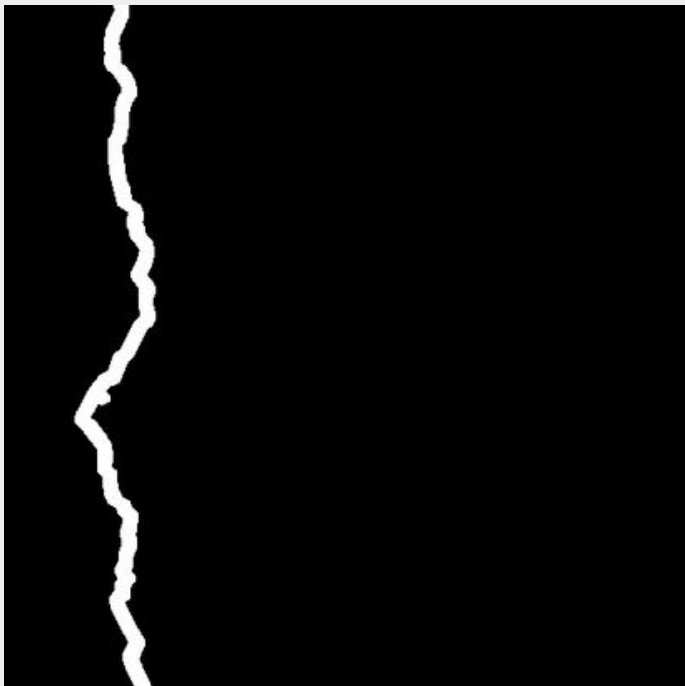
CrossEntropyLoss 함수의 **weights**
변수로 가중치 부여

모델	가중치 (배경, 균열)	성능(max mIoU)
SCSegamba	[0.1, 0.9]	0.611
SCSegamba	[0.2, 0.8]	0.636
SCSegamba	[0.3, 0.7]	0.605
SCSegamba	[0.4, 0.6]	0.585
SCSegamba	[0.5, 0.5]	0.581
SCSegamba	[0.1, 1.9]	0.600
SCSegamba	[0.2, 1.8]	0.604
SCSegamba	[0.3, 1.7]	0.625
SCSegamba	[0.4, 1.6]	0.633
SCSegamba	[0.5, 1.5]	0.625

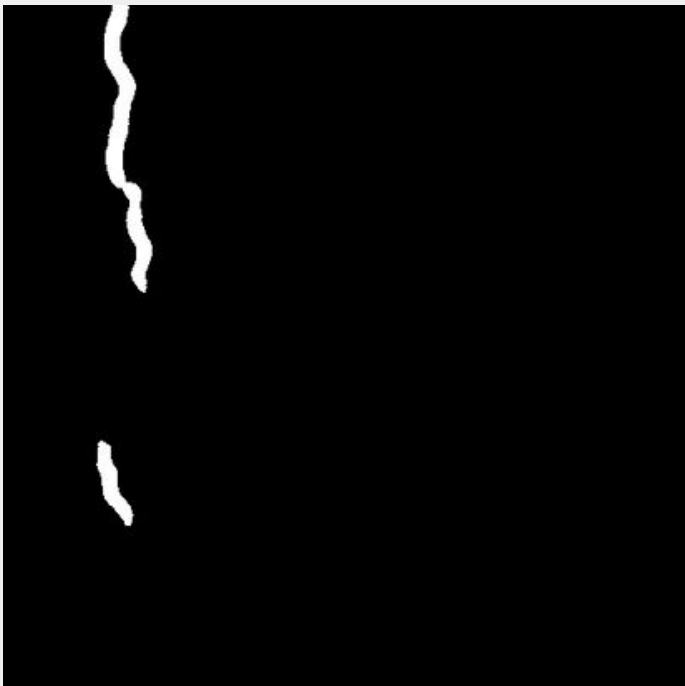
Multi-Class 실험

◦Dilation 데이터셋 학습

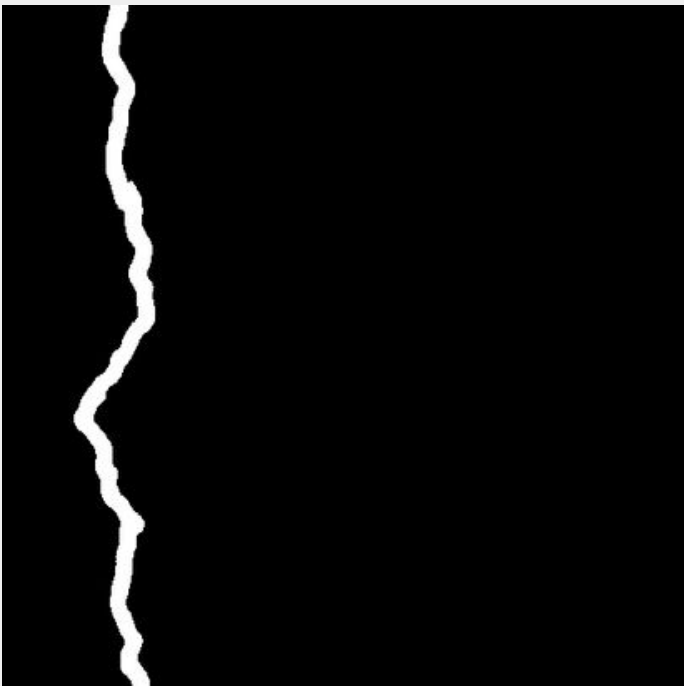
Dilation 데이터셋을 이용해 학습 후 성능과 결과 확인



Label



1 Class Prediction



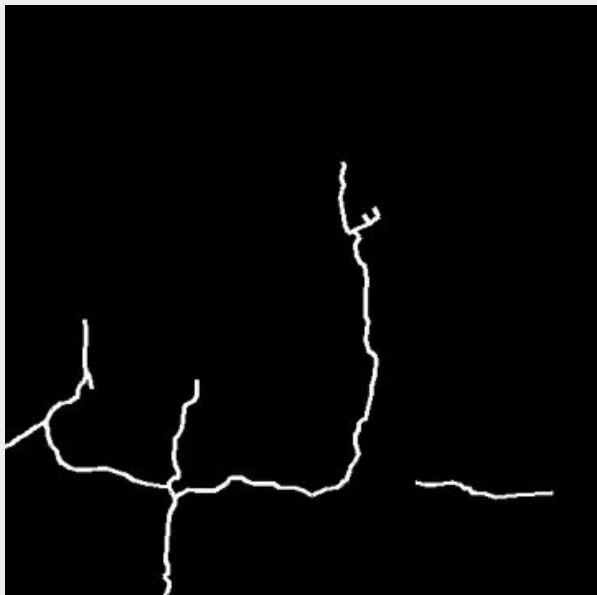
2 Class Prediction

모델	weights	Test Dataset	성능(max mIoU)	비고
SCSegamba	-	Dilation	0.716	1 Class 실험
SCSegamba	[0.2, 0.8]	Dilation	0.753(+0.37)	1 Class 실험보다 더 높은 성능

Multi-Class Fine-tuning 실험

◦Dilation 학습 모델 Fine-tuning

Multi-Class 실험에서 Fine-tuning 여부에 따른 성능 변화 확인



Label



No Fine-tuning
Prediction



Fine-tuning
Prediction

모델	Test Dataset	성능(max mIoU)	비고
SCSegamba	polygon = linestrip	0.7 (+0.061)	Dilation 데이터셋을 학습한 모델을 Fine tuning
SCSegamba	polygon = linestrip	0.697 (+0.058)	Fine tuning 없이 polygon = linestrip 데이터로 학습

초기 polygon = linestrip 실험 결과 성능 (0.639 대비)

후처리 실험 - Morphology

◦Erosion, Dilation, Skeleton 등 Morphology 실험 진행

Dilation 데이터셋을 학습한 모델의 예측을 실제와 같이 얇게 만들기 위함

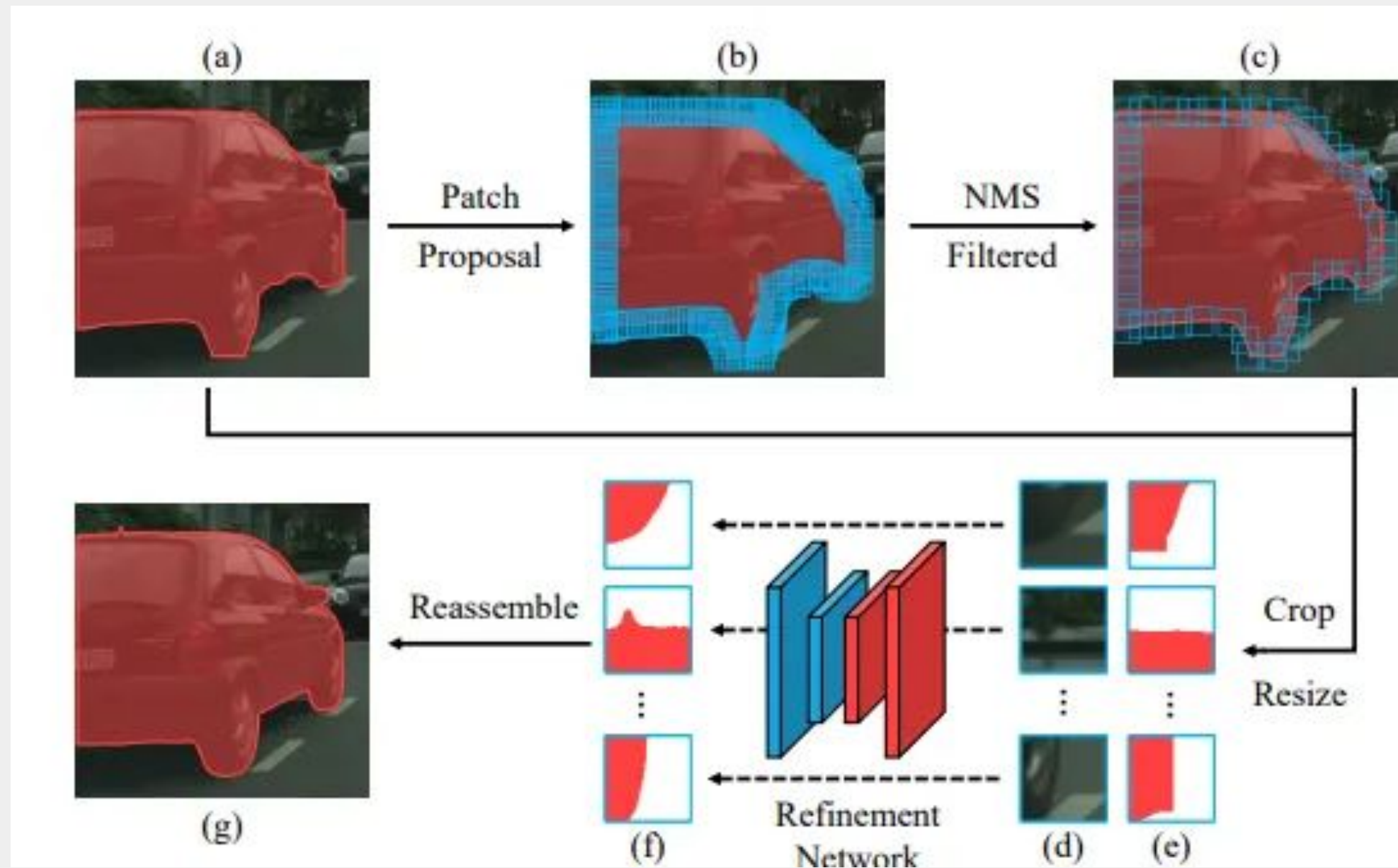
polygon = linestrip 데이터 기준 최고 성능이 0.7에 비해 낮은 성능을 확인

실험 이름	Train Dataset	Test Dataset	후처리 작업	kernel 크기	성능 (max mIoU)
erosion 1	Dilation	Polygon = Linestrip	erosion 1회	5	0.63
erosion 2	Dilation	Polygon = Linestrip	erosion 2회	5	0.666
erosion 3	Dilation	Polygon = Linestrip	erosion 3회	3	0.643
erosion 4	Dilation	Polygon = Linestrip	erosion 4회	3	0.651
erosion 5	Dilation	Polygon = Linestrip	erosion 5회	3	0.65
skeleton 1	Dilation	Polygon = Linestrip	중심축 기준 dilation 2회	5	0.653
skeleton 2	Dilation	Polygon = Linestrip	중심축 기준 dilation 1회	3	0.657

후처리 실험 - BPR(Boundary Patch Refinement)

◦ Segmentation 후처리 기법 중 하나인 BPR 적용

BPR이란, Segmentation mask의 경계 부분 분석하는 새로운 Network를 구성하여 경계 품질을 향상시키는 방법

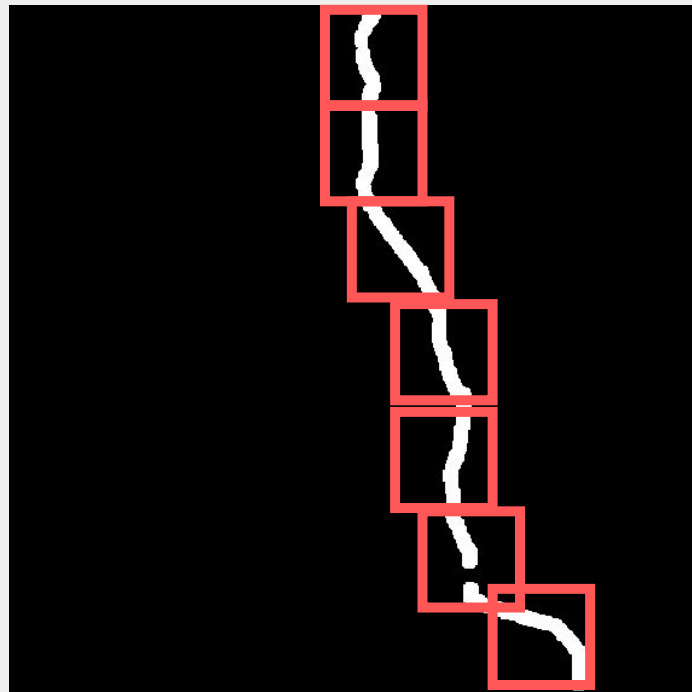


후처리 실험 - BPR(Boundary Patch Refinement)

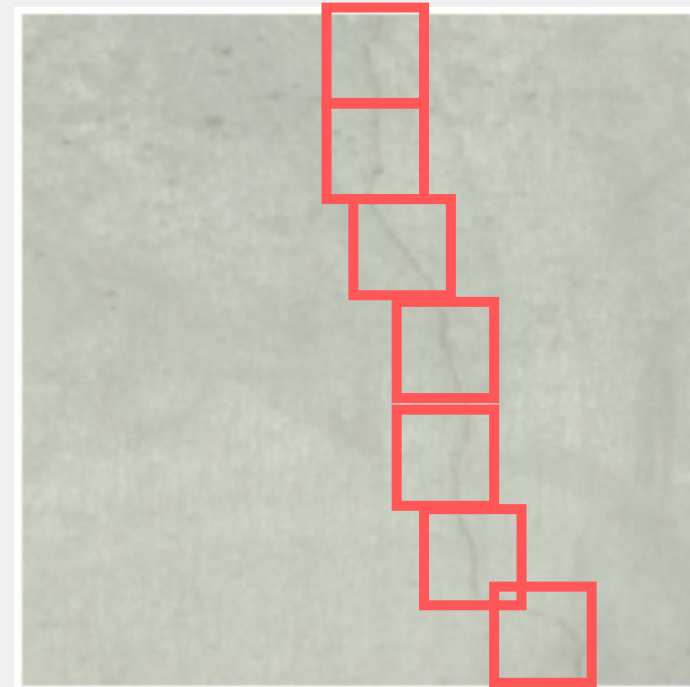
◦BPR을 바로 Crack Dataset에 적용하기는 어려움

Crack Dataset의 특성 상 얇은 mask가 대부분이기에 경계 패치를 가져오기 까다로움

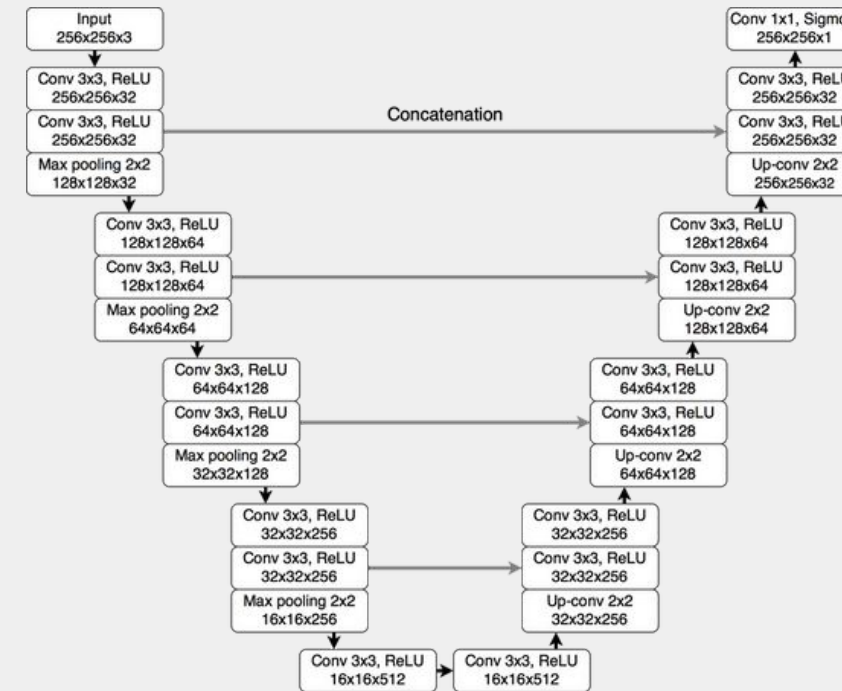
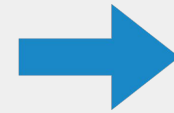
따라서 Crack Dataset에 맞도록 경계가 아닌 Crack이 포함 되어 있는 패치를 기준으로 Network 구성



Dilation Prediction



Original Image



UNet



BPR Result

후처리 실험 - BPR(Boundary Patch Refinement)

◦BPR 모델인 UNet 실험 진행

현재 패치의 Resolution 값이 작기 때문에 UNet 모델 구조 변경 실험 진행
패치 크기와, Stride 실험 진행을 통해 패치크기와 Stride 결정

모델	layer 갯수	패치 크기	Stride	성능 (max mIoU)	비고
UNet	4	256	128	0.672	Default
UNet	4	128	64	0.679	-
UNet	4	64	32	0.671	-
UNet	3	128	64	0.678	제일 깊은 층 제거
UNet	3	128	64	0.670	제일 얇은 층 제거
UNet	3	64	32	0.671	제일 깊은 층 제거

후처리 실험 - BPR(Boundary Patch Refinement)

○ 최종 UNet 모델을 이용한 BPR 적용 실험

최종 UNet의 성능 max mIoU 기준 0.7145 달성

모델	Patch size	stride	layer 갯수	best epoch	epoch	max mIoU
UNet	128	64	4	30	50	0.7145

Dilation 데이터셋 학습 모델 결과에 적용

실험 이름	후처리	test dataset	성능 max mIoU
bpr 1	Dilation 데이터셋, 1 Class 실험결과에 bpr 적용	polygon = linestrip	0.673 (-0.027)
bpr 2	Dilation 데이터셋, 2 Class 실험결과에 bpr 적용	polygon = linestrip	0.680 (-0.02)

현재 가장 높은 성능의 모델 (0.7 대비)

후처리 실험 - BPR(Boundary Patch Refinement)

○ 최종 UNet 모델을 이용한 BPR 적용 실험

 polygon = linestrip 데이터 셋을 학습한 multi class 실험 결과에 적용

실험 이름	후처리	test dataset	성능 max mIoU
bpr 3	multi class 실험결과에 bpr 적용	polygon = linestrip	0.690 (-0.01)
bpr 4	multi class 실험결과에 bpr 적용 → erosion	polygon = linestrip	0.683 (-0.017)
bpr 5	multi class 실험결과 → dilation → bpr	polygon = linestrip	0.687 (-0.013)

현재 가장 높은 성능의 모델 (0.7 대비)

BPR을 적용한 결과를 확인했을 때 성능 향상을 보이는 경우는 없기에 후처리
과정에서 제외

후처리 실험 - TTA

○ Test Time Augmentation

모델의 예측 과정에 Data Augmentation 기법을 적용하여 성능 향상

실험 이름	후처리	test dataset	성능 max mIoU
TTA 1	RandomBrightnessContrast	polygon = linestrip	0.689 (-0.011)
TTA 2	CLAHE	polygon = linestrip	0.701 (+0.001)
TTA 3	d4_transform (H, V flip + rotation 0, 90, 180, 270)	polygon = linestrip	0.704 (+0.004)
TTA 4	d4_transform + CLAHE	polygon = linestrip	0.698 (-0.002)

현재 가장 높은 성능의 모델 (0.7 대비)

배경 제거 실험

- 기존 낮은 성능의 배경 제거 **Classification** 모델 문제
 - 교각 부분만을 예측하는 **Segmentation** 모델을 이용해 교각 외부 오탐을 제거
 - rembg** 라이브러리를 이용해 데이터셋 구성

모델	Encoder	Encoder Weights	Epoch	성능 max mIoU
UNet	resnet34	imagenet	50	0.9493
Segforemr	resnet34	imagenet	50	0.9635

