

# MATLAB EXPO 2019

## 철도 궤도 결함 탐지를 위한 영역 기반 및 픽셀 기반 딥러닝 기법 적용 사례

Detection of track defects by  
region- and pixel-based deep learning approaches

한국철도기술연구원 / 첨단궤도토목본부  
황성호 선임연구원



# 목차

1. 회사 및 발표자 소개
2. 철도 및 궤도 소개
3. 프로젝트 개요
4. 연구 내용
  - . 영역 기반 결함 탐지
  - . 픽셀 기반 결함 탐지
  - . Network 수정
5. 기술적 어려움
6. 향후 연구 방향
7. MATLAB에 대한 소감

# 한국철도기술연구원은?

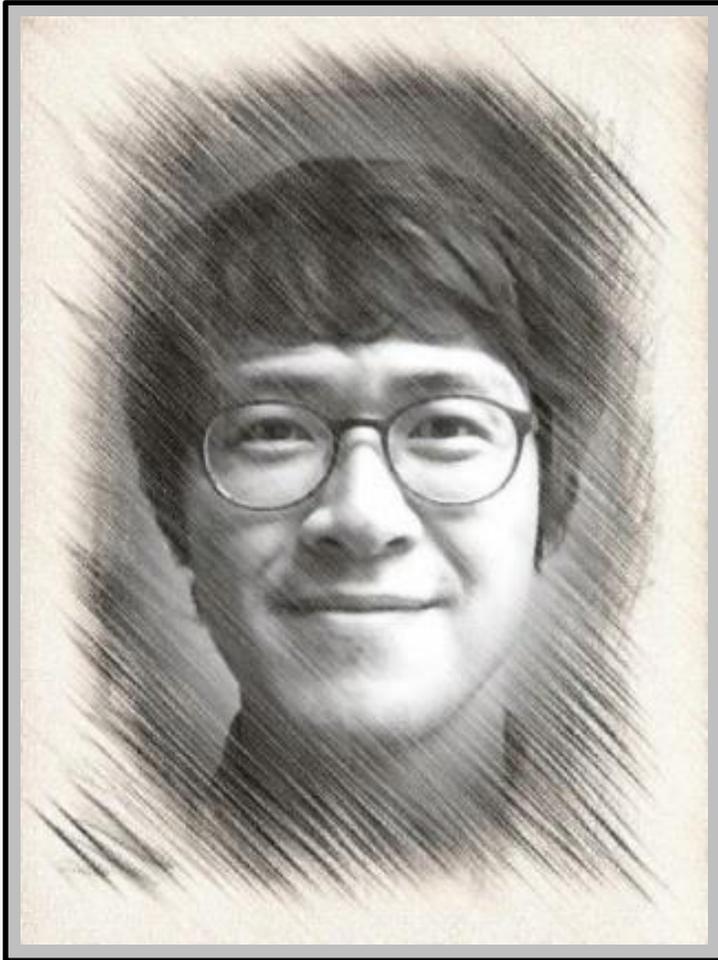


Korea Railroad Research Institute

<https://www.krri.re.kr>

- 소속 및 소재지 : 과학기술정보통신부 산하 정부출연연구기관 / 경기도 의왕
- 임무 : 철도, 대중교통, 물류 등 공공교통 분야의 연구개발 및 성과확산을 통해 국가 및 산업계 발전에 기여
- 주요 기능 및 역할
  - . 고속철도, 일반철도, 도시철도 및 경량전철시스템 연구개발
  - . 차세대 대중교통시스템 연구개발
  - . 철도안전, 표준화, 철도정책 및 물류기술 연구개발
  - . 남북철도 및 대륙철도 연계기술 연구개발
  - . 철도, 대중교통, 물류 등 공공교통시스템 핵심원천기술 연구개발
  - . 중소·중견기업 등 관련 산업계 협력·지원 및 기술사업화 등

## 발표자 황 성 호 선임연구원은?



이름 : 황 성 호

부서 : 첨단궤도토목본부, 궤도노반연구팀

경력 : 2003.07.~현재 한국철도기술연구원

전공 : 토목공학

주요연구분야 : 궤도공학, 차량궤도 상호작용

E-mail : forever7@krri.re.kr

# 철도란?

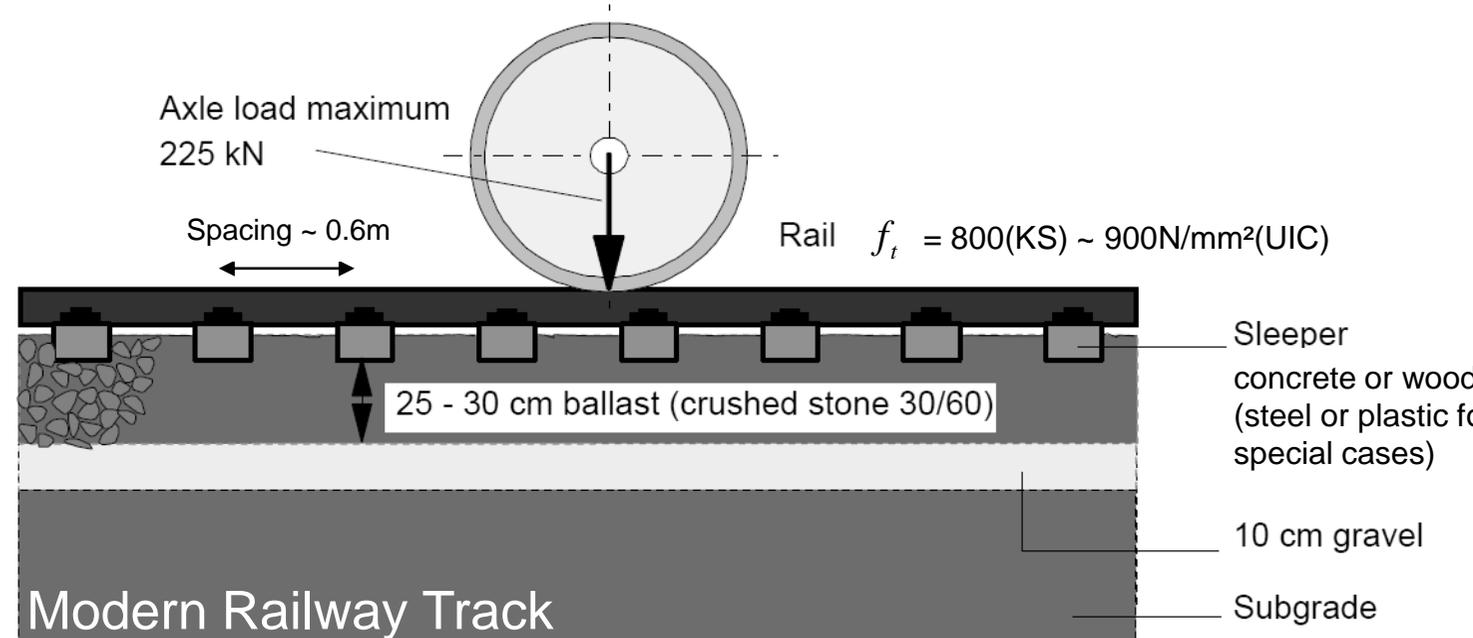
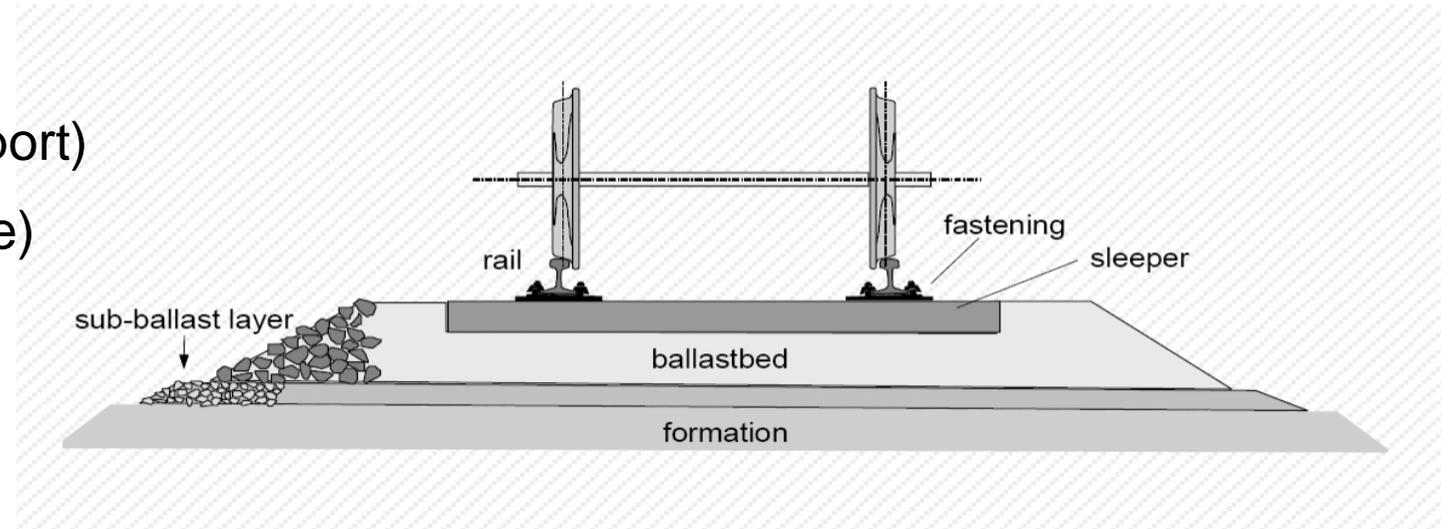


# 궤도란?

- 궤도의 역할 : 차량을 직접 지지(support)  
정해진 길로 유도(guide)

- 기본 궤도의 구성 :

- . 레일(rail)
- . 체결장치(fastener)
- . 침목(sleeper, tie)
- . 자갈도상(ballast)
- . 노반(roadbed)



# 아스팔트 도로포장의 파손

- 균열



[거북등 균열]



[단부 균열]

# 아스팔트 도로포장의 파손

## - 균열



[차로와 길어깨 줄눈 균열]



[세로방향 균열]

# 아스팔트 도로포장의 파손

- 변형



[러팅]



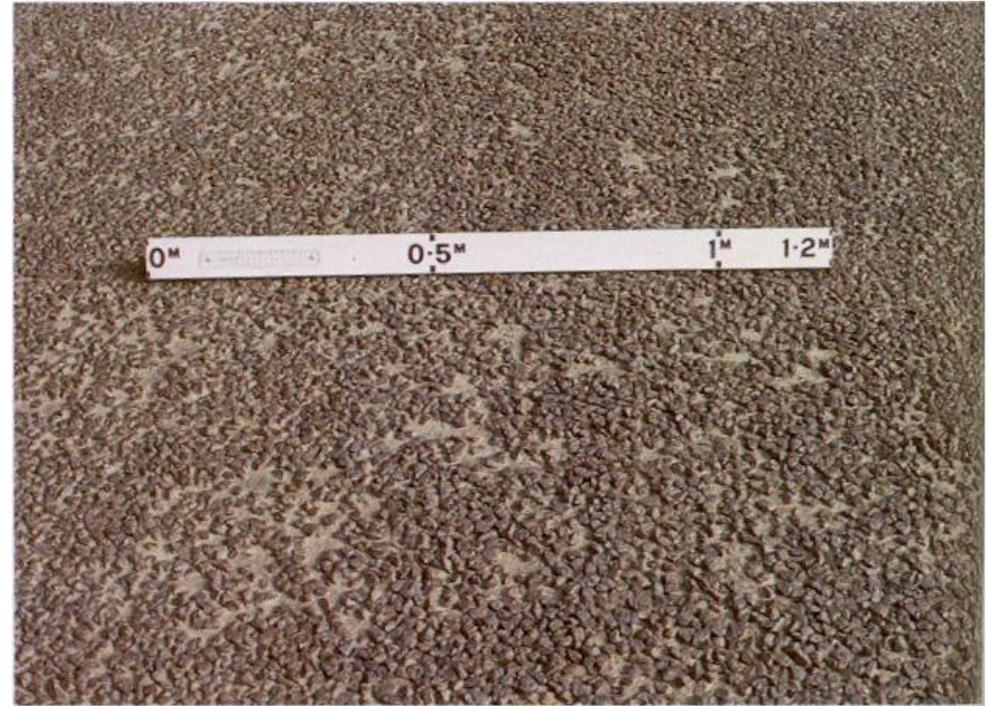
[코러게이션]

# 아스팔트 도로포장의 파손

- 탈리



[포트홀]



[박리]

# 궤도 구성품의 파손

- 레일



<https://www.picswe.com>

[레일 파단]



RailCorp Engineering Manual  
- Track Surface Defects in Rails

[레일 스퀘트]

# 궤도 구성품의 파손

- 레일



RailCorp Engineering Ma  
- Track Surface Defects

[레일 코러게이션]



RailCorp Engine  
- Track Surface

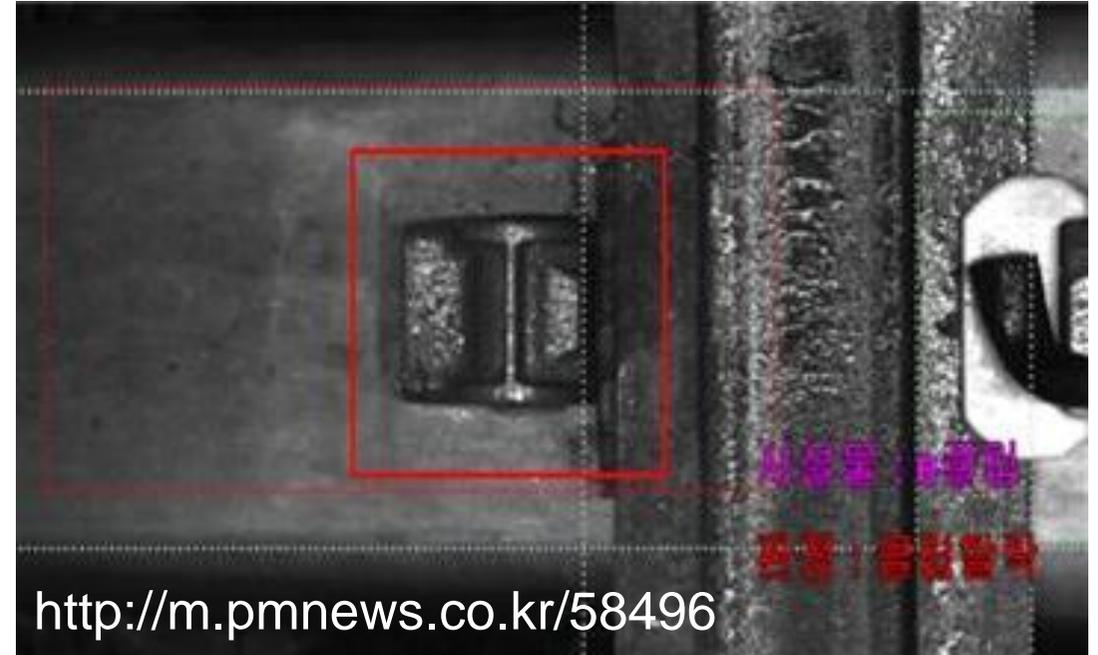
[헤드 체크]

# 궤도 구성품의 파손

- 체결장치



[파손]



[탈락]

# 궤도 구성품의 파손

- 침목



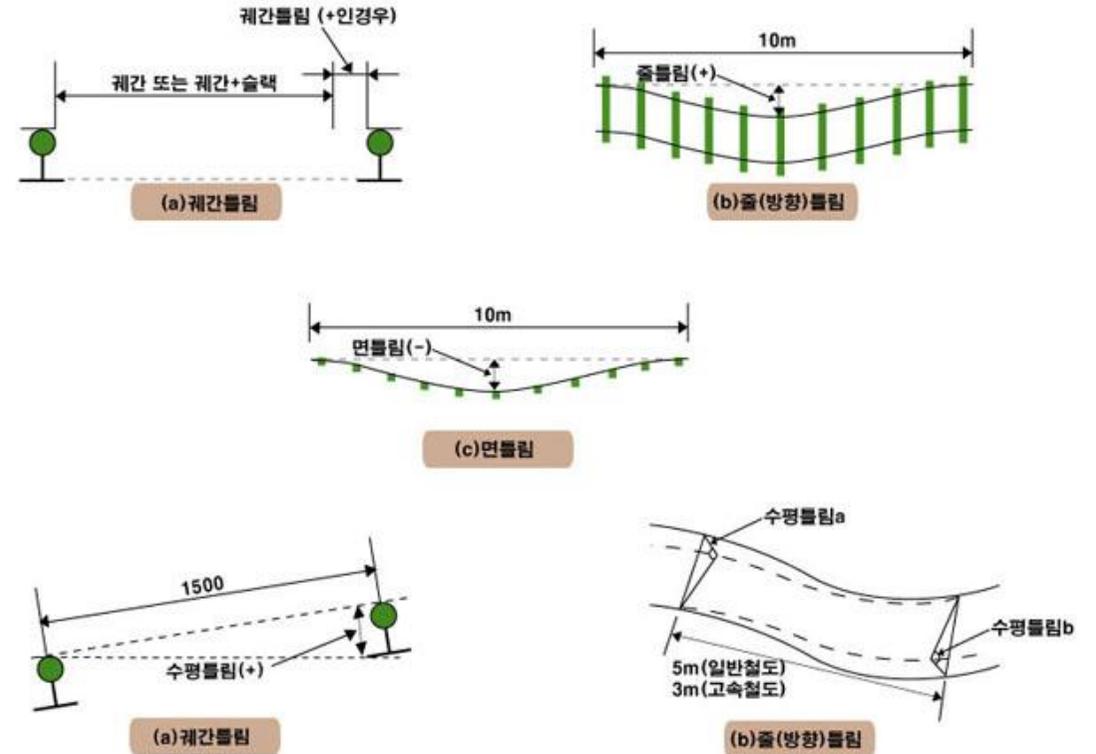
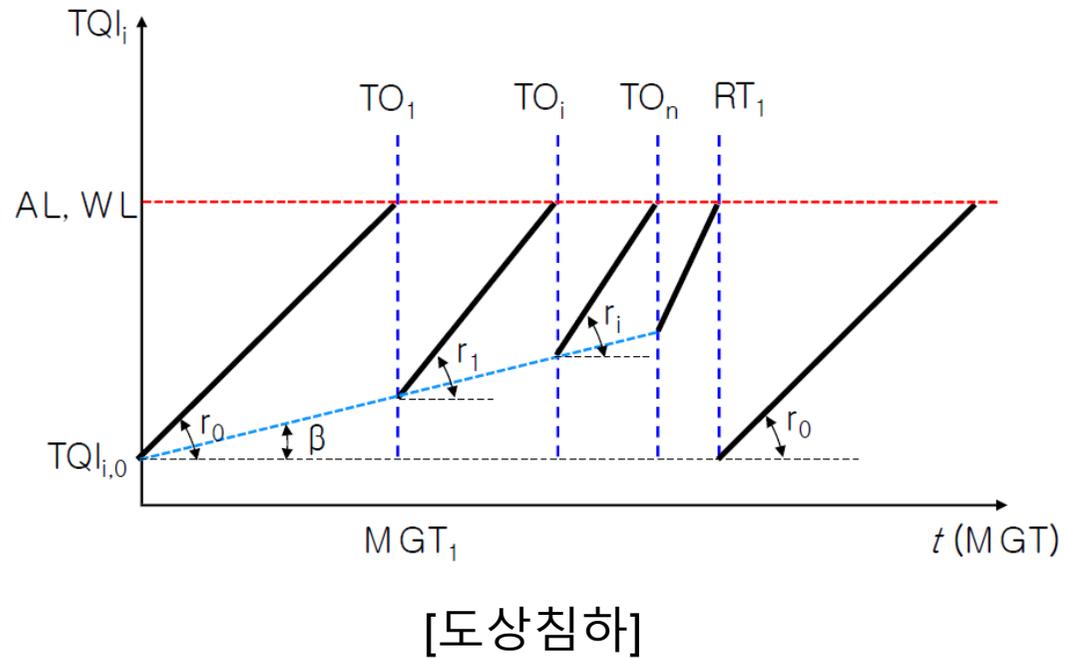
[균열]



[침목 파손]

# 궤도 구성품의 파손

- 자갈도상 : 침하, 궤도틀림



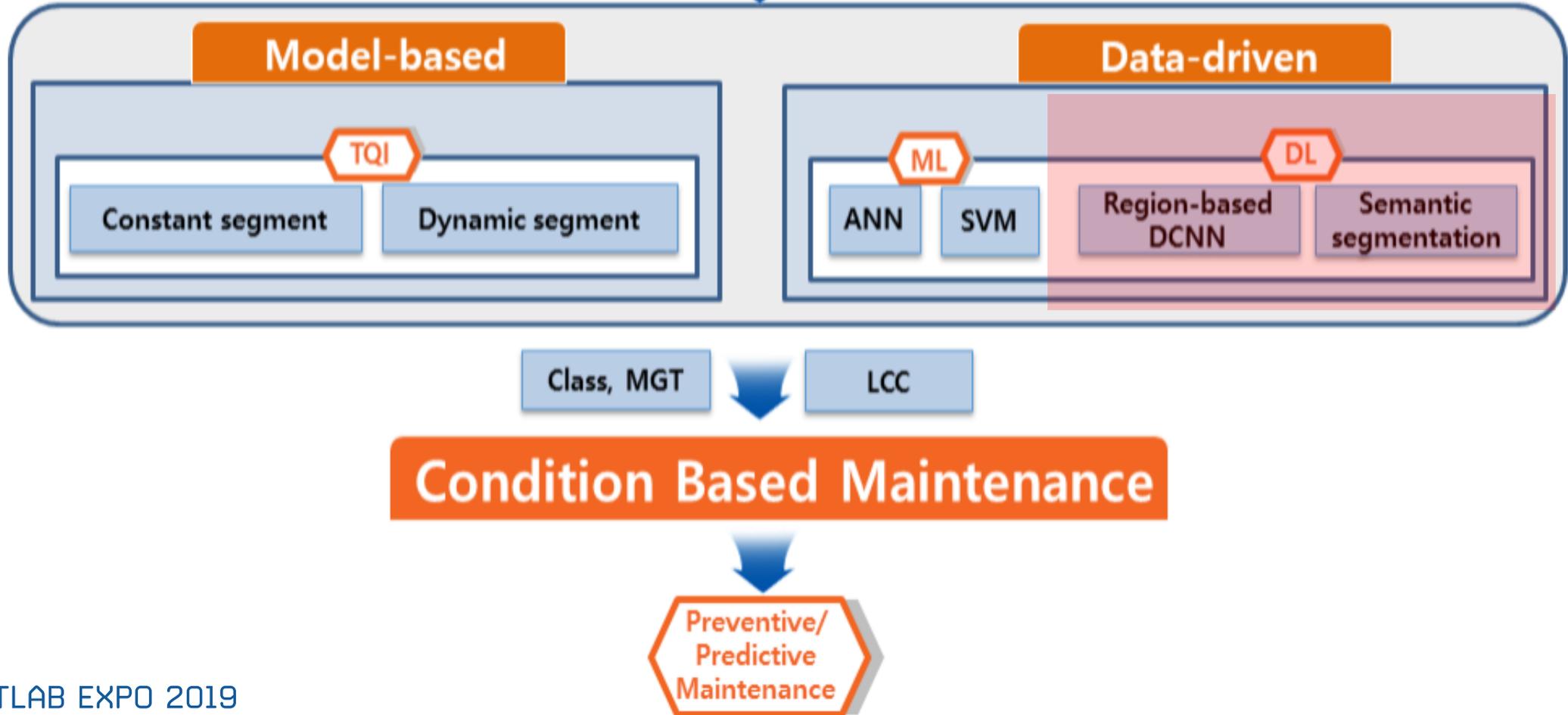
©2008 철도산업정보센터

[궤도틀림]

# 궤도 유지보수 의사결정 체계



TMV



## 과업 개요

- 과업 목표 : 궤도 결함(주) 탐지를 위한 이미지 프로세싱 Deep Learning 기법 도입 및 적용
- 과업 기간 : 2018. 01. ~ 2020. 12. (3년)
- 진행 방향 :
  - . Network : 기존 trained network(SegNet)을 이용하여 transfer learning 수행
  - . 영역 기반 결함탐지 우선 수행 [1]
  - . 픽셀 기반 결함탐지를 통해 세부적인 결함 탐지 [2]
  - . 정확도를 높이기 위해 Network 수정 [3]
- **MATLAB 2018b or 2019a**의 machine learning 및 deep learning toolbox를 이용

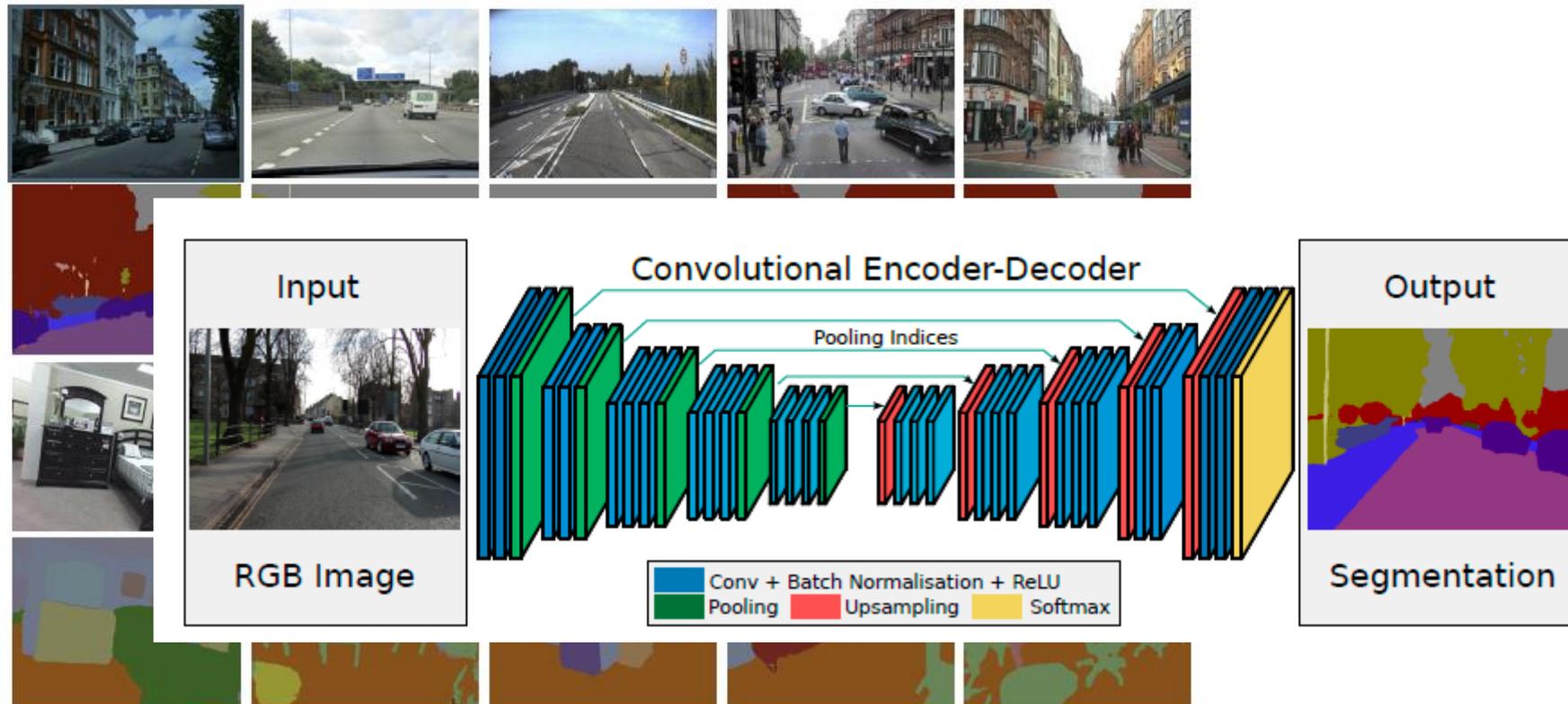
주) 궤도 결함 : 다양한 외부 환경 및 열차 하중에 따라 궤도 구성품에 발생하는 결함을 의미.

예) 레일 표면 결함, 침목 균열 및 파손, 체결구 탈락 등

# SegNet

- A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

*Vijay Badrinarayanan, Alex Kendall, etc*



# 1. 영역 기반 결함 탐지 - input

- Bounding Box(BB)



[당초 이미지]

- 개체 구분 : 나뭇가지, 솔잎



[영역 지정 이미지]

# 1. 영역 기반 결함 탐지 – method

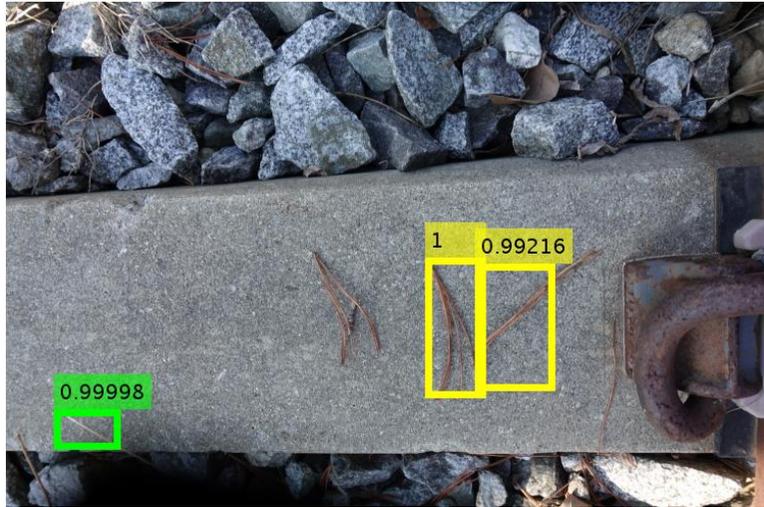
- RCNN : selective searching 기법<sup>주1)</sup>
- FCNN : Fast RCNN<sup>주2)</sup> 영역기반 가속 심층학습
- FFCNN : Faster RCNN<sup>주3)</sup>

주1) Girshick, R., et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”, Proc. IEEE, CVPR, 580-587, 2014.

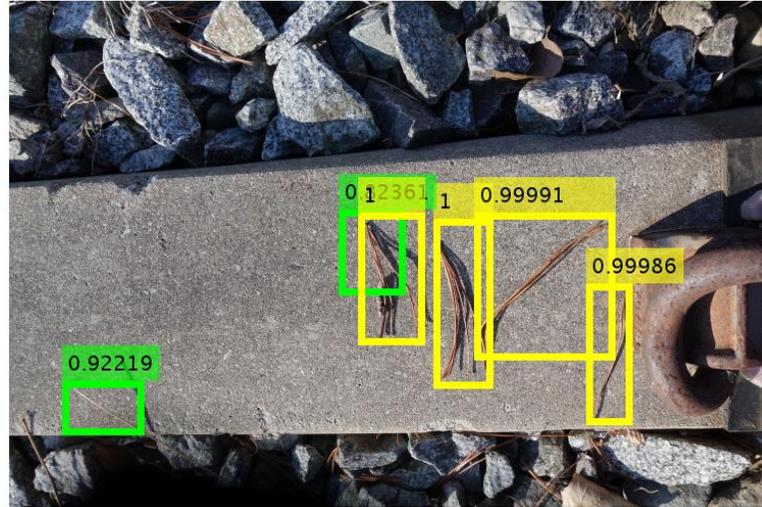
주2) Girshick, R. “Fast R-CNN”, Proc. IEEE ICCV, 1440-1448, 2015.

주3) Ren, S., et al. “Faster R-CNN: Towards real-time object detection with region proposal networks”, arXiv:1506.01497, 2016.

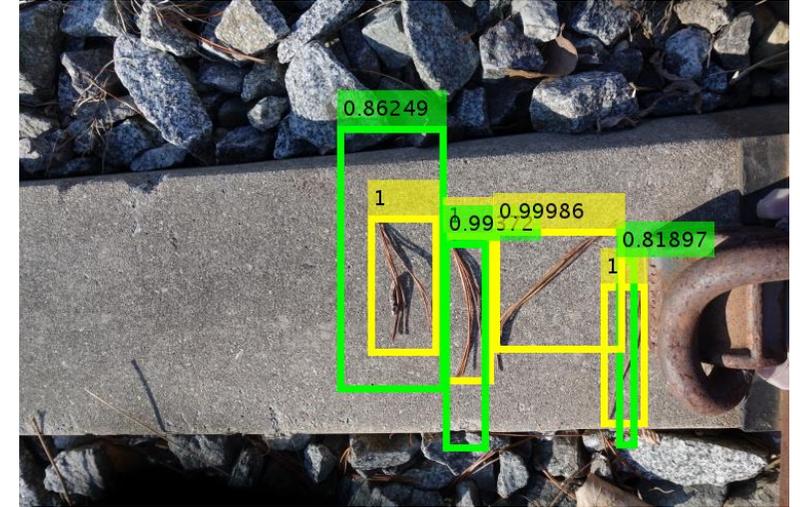
# 1. 영역 기반 결함 탐지 - 결과



[RCNN]



[FCNN]



[FFCNN]

Method	RCNN	FCNN	FFCNN
Pr (leaf)	0.04	0.04	0.35
Pr (branch)	0.03	0.0	0.34

## 2. 픽셀 기반 결함 탐지 – method

### - Semantic Segmentation(SS)<sup>주)</sup>

- . 기존 DCNN 모형에 적용한 계층구조를 적용하여 encoder를 구성
  - pooling 층에 대응하는 upsampling 층을 도입
  - decoder 모형을 작성하는 동시에 픽셀단위의 softmax 함수를 통해 대상물을 구분
- . Decoder는 encoder에서 max pooling 층을 사용하여 구축한 특성맵의 해상도를 다시 향상시키는 과정으로 이해할 수 있으며, 이 때 pooling층에서 사용한 인수를 기억한 후 upsampling 층에서 다시 사용함으로써 원하는 특성맵을 획득
- . 따라서 균열 등과 같이 길이 대비 폭이 작은 대상물의 검지에 RCNN 대비 상대적으로 우월한 결과 획득 가능

주) Badrinarayanan, V., et al. "SegNet: A deep convolutional encoder-decoder architecture for image segmentation", IEEE Transactions on pattern Analysis and Machine Intelligence, 39(12), pp. 2481-2495, 2017.

Garcia-Garcia, A, et al. "A review on deep learning techniques applied to semantic segmentation", arXiv:1704.06857, 2017.

## 2. 픽셀 기반 결함 탐지 1 - input



[당초 이미지]



[픽셀 지정 이미지]

- 개체 구분 : 자갈, 침목, 체결장치, 레일, 나뭇가지, 솔잎

Label | Label Pixels

Label | Polygon | Smart Polygon | Smart Polygon Editor | Assisted Freehand | Brush | Erase | Brush Size | Flood Fill | Label Opacity

MODE | POLYGON | FREEHAND | BRUSH | FLOOD FILL | LABEL OPACITY

ROI Label Definition

- re\_DSC01793
- Define New ROI Label
- ▶ Ballast
  - ▶ Sleeper
  - ▶ Rail
  - ▶ Eclip
  - ▶ Crack
  - ▶ Branch
  - ▶ Stone
  - ▶ Leaf
  - ▶ **Cavity**

Scene Label Definition

Define new scene label

Apply to Image

Remove from Image

To label a scene, you must first define a scene label.

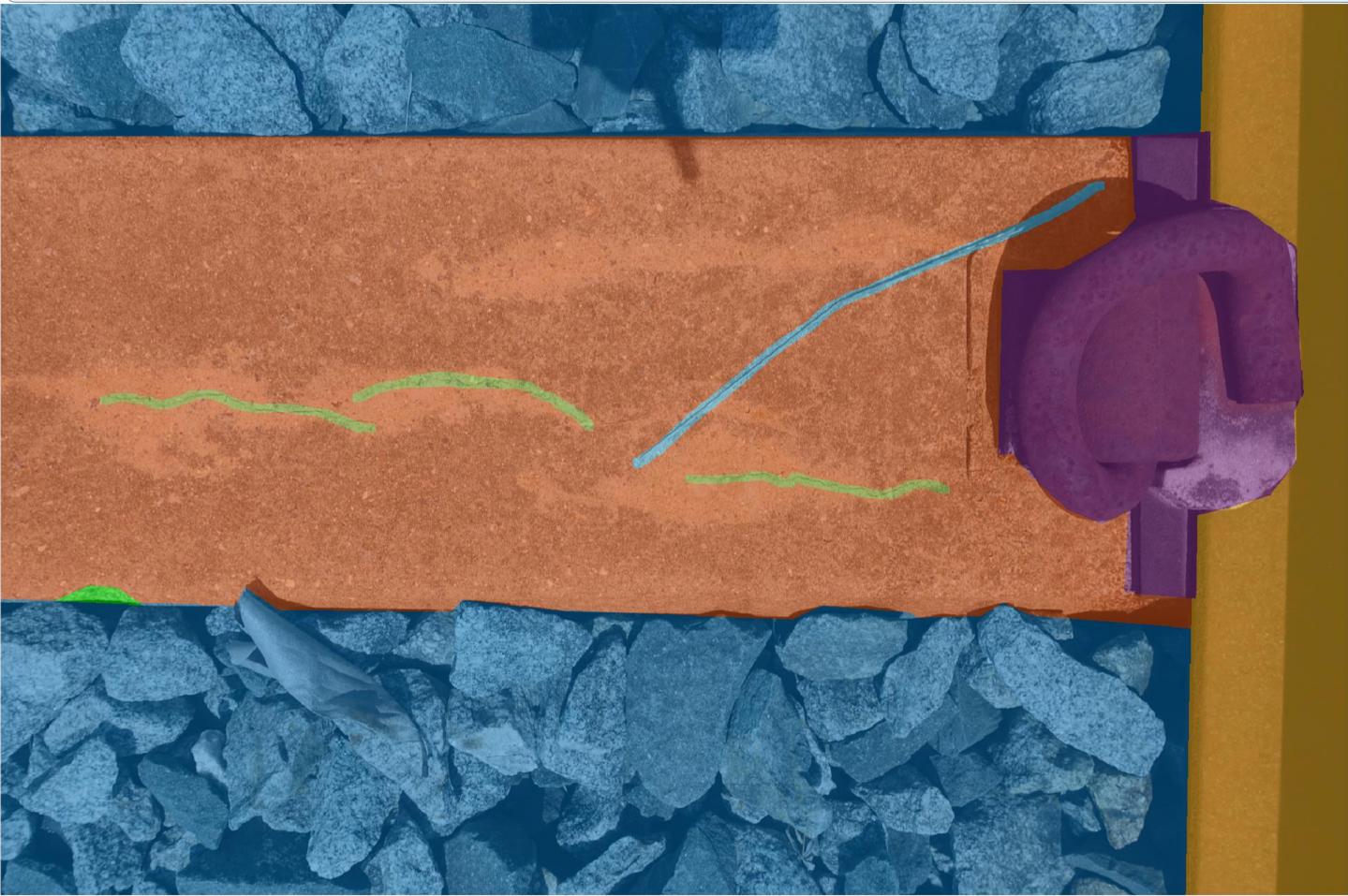


Image Labeler - 1

LABEL LABEL PIXELS

Zoom In Zoom Out Pan Label Polygon Smart Polygon Smart Polygon Editor Assisted Freehand Brush Erase Flood Fill Label Opacity

MODE POLYGON FREEHAND BRUSH FLOOD FILL LABEL OPACITY

ROI Label Definition re\_DSC01793

Define New ROI Label

- Ballast
- Sleeper
- Rail
- Eclip
- Crack
- Branch
- Stone
- Leaf
- Cavity**

Scene Label Definition

Define new scene label

Apply to Image

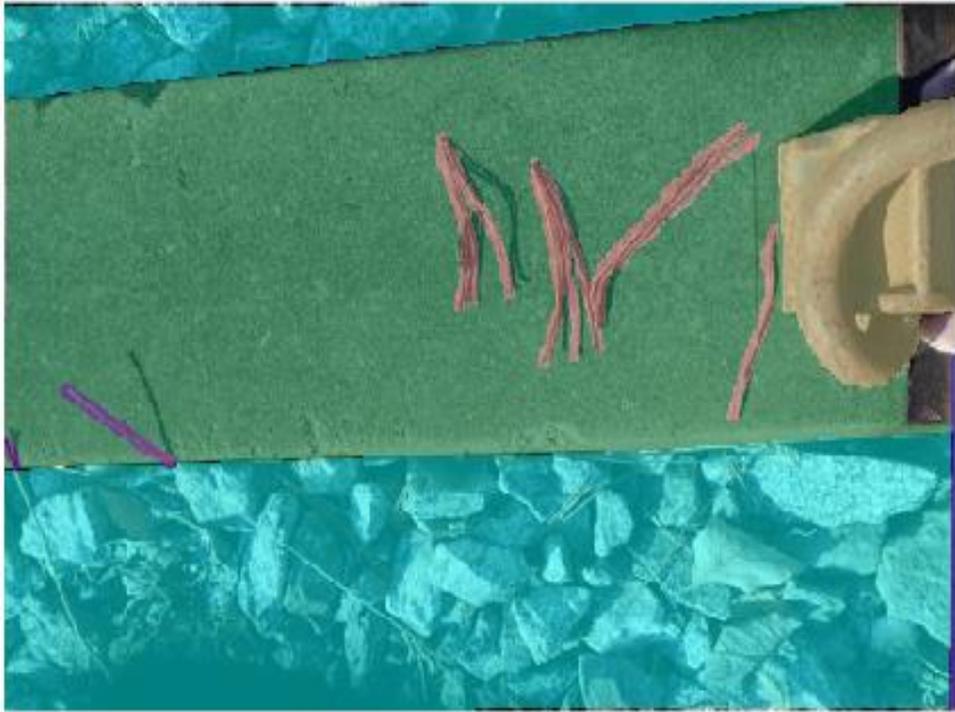
Remove from Image



Brush  
→ digitizer



## 2. 픽셀 기반 결함 탐지 1 - 결과



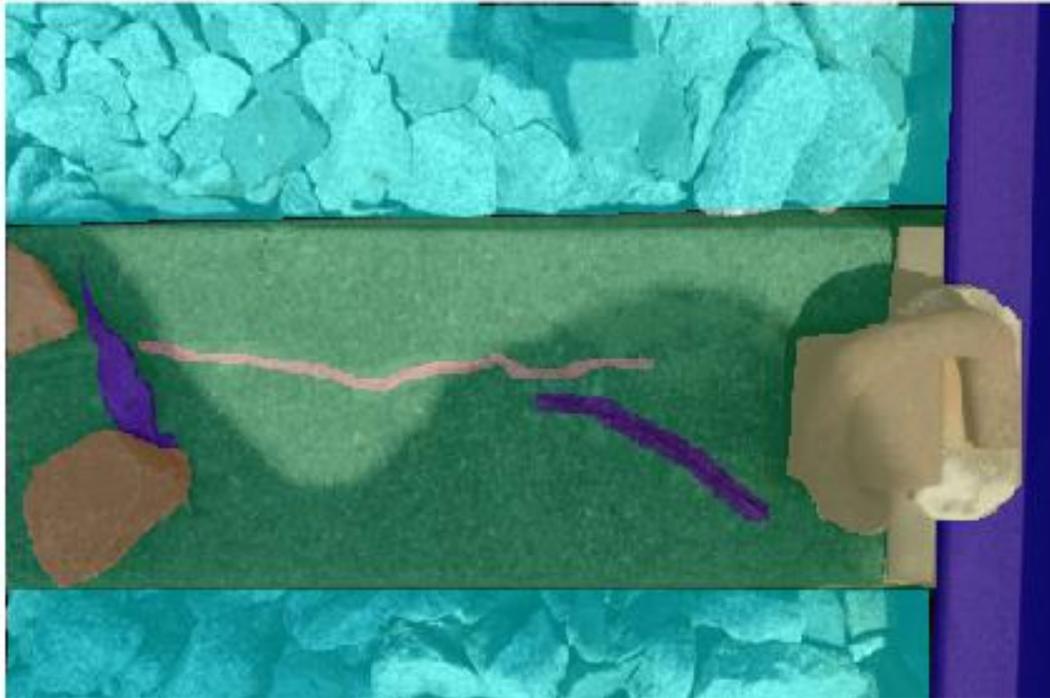
[입력 이미지]



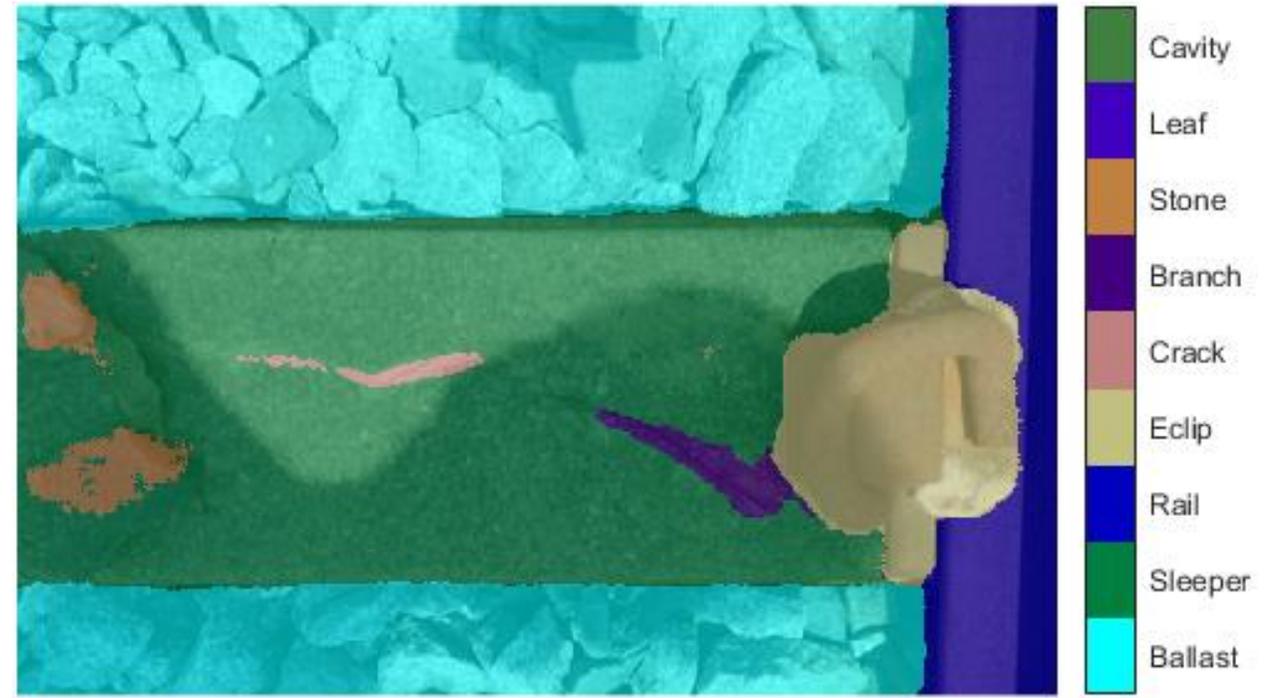
[결과 이미지]

Model	Leaf		Branch	
	Pr	IoU	Pr	IoU
SS DCNN	0.96	0.50	0.81	0.48

## 2. 픽셀 기반 결함 탐지 2



[입력 이미지]

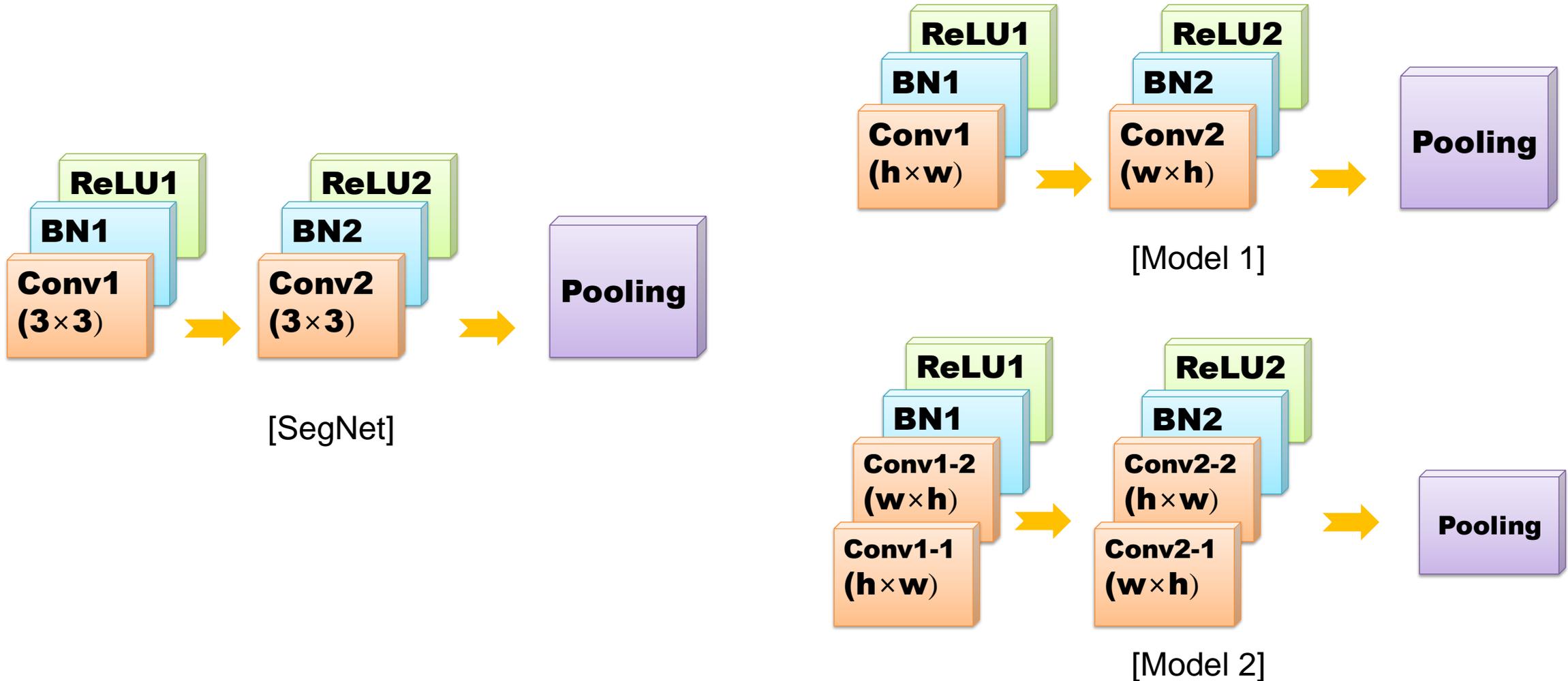


[결과 이미지]

Object	avg. Pr	IoU	Object	avg. Pr	IoU
Ballast	0.99	0.97	Crack	0.76	0.43
Sleeper	0.96	0.94	Branch	0.81	0.53
Rail	0.99	0.96	Stone	0.89	0.69
E-clip	0.99	0.93	Leaf	0.46	0.28

### 3. 정확도 향상을 위한 방안 - Kernel size 변화

- Concept



```
%% DeepRail layers with rectangular kernels          2019. 2. 1.
% Needs unique names for each layer                19. 2. 2.
```

```
%% ht : Kernel height
%% wt : Kernel width
%% pl : pooling size
%% pa : padding size
```

```
ht0 = 3;
wt0 = 3;
pl0 = 2;
ht = 3;
wt = 3;
pl = 2;
pah = (ht-1)/2;
paw = (wt-1)/2;
```

### Kernel size 입력

```
%%%%%%%%% 3, 5, 7, 9
%%%%%%%%% 3, 5, 7, 9
%%%%%%%%% 19.2.4.
%%%%%%%%% 19.2.4.
```

```
% https://kr.mathworks.com/help/vision/ug/semantic-segmentation-examples.html?searchHighlight=
%%
% Model 2 %%%%%%%%%%
```

```
array3_2 = [convolution2dLayer([ht wt], 256, 'Padding', 'same', ... % [pah pah paw paw], ...
    'Name', 'mconvhw3_2')
    convolution2dLayer([wt ht], 256, 'Padding', 'same', ... % [paw paw pah pah], ...
    'Name', 'mconvvh3_2')];
array3_3 = [convolution2dLayer([wt ht], 256, 'Padding', 'same', ... % [paw paw pah pah], ...
    'Name', 'mconvvh3_3')
    convolution2dLayer([ht wt], 256, 'Padding', 'same', ... % [pah pah paw paw], ...
    'Name', 'mconvhw3_3')];
array4_2 = [convolution2dLayer([ht wt], 512, 'Padding', 'same', ... % [pah pah paw paw], ...
    'Name', 'mconvhw4_2')
    convolution2dLayer([wt ht], 512, 'Padding', 'same', ... % [paw paw pah pah], ...
    'Name', 'mconvvh4_2')];
```

```
drray3_3 = [convolution2dLayer([wt ht], 256, 'Padding', 'same', ... % [paw paw pah pah], ...
    'Name', 'mdecoder3wh_conv3')
    convolution2dLayer([ht wt], 256, 'Padding', 'same', ... % [pah pah paw paw], ...
    'Name', 'mdecoder3hw_conv3')];
drray3_2 = [convolution2dLayer([ht wt], 256, 'Padding', 'same', ... % [pah pah paw paw], ...
    'Name', 'mdecoder3hw_conv2')
    convolution2dLayer([wt ht], 256, 'Padding', 'same', ... % [paw paw pah pah], ...
    'Name', 'mdecoder3wh_conv2')];
```

```
lgraph = replaceLayer(lgraph, 'conv3_2', array3_2);
lgraph = replaceLayer(lgraph, 'conv3_3', array3_3);
lgraph = replaceLayer(lgraph, 'conv4_2', array4_2);
lgraph = replaceLayer(lgraph, 'conv4_3', array4_3);
lgraph = replaceLayer(lgraph, 'conv5_2', array5_2);
lgraph = replaceLayer(lgraph, 'conv5_3', array5_3);
```

### Layer 치환 / 삭제 / 추가

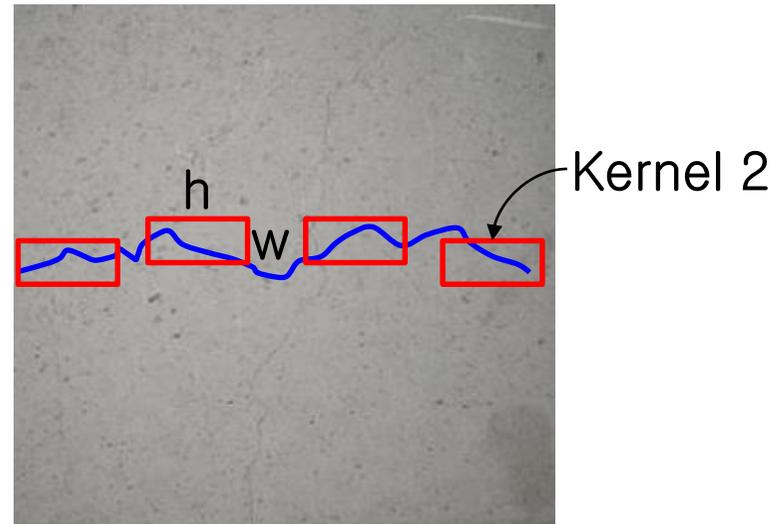
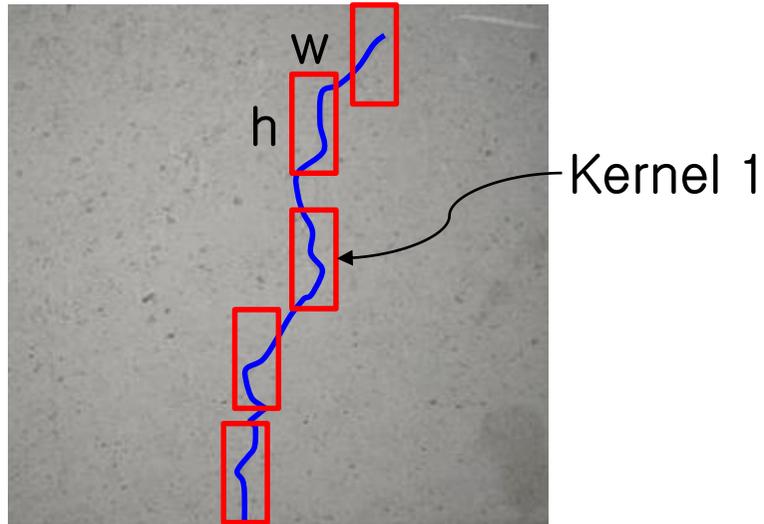
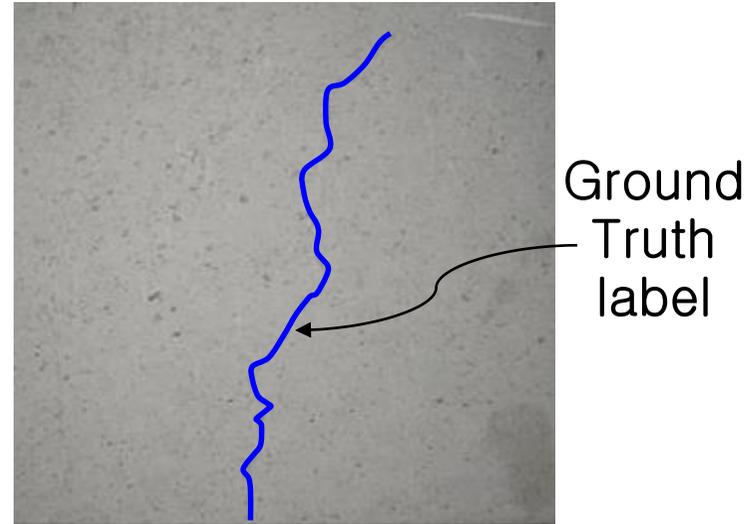
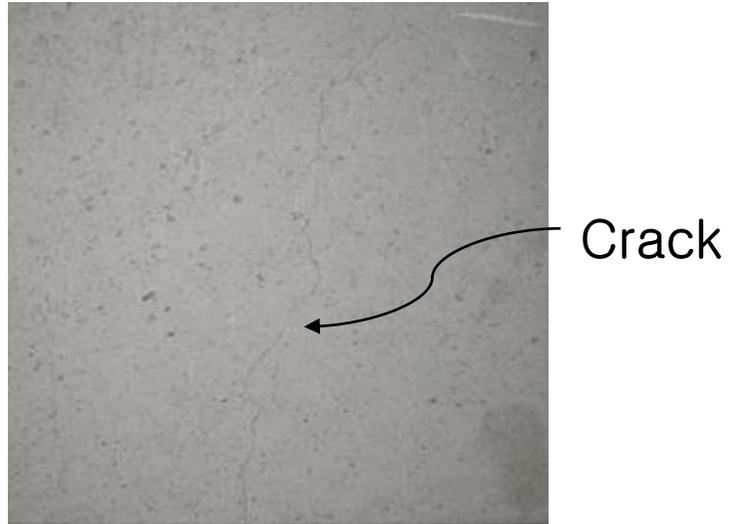
```
lgraph = replaceLayer(lgraph, 'decoder5_conv3', drray5_3);
lgraph = replaceLayer(lgraph, 'decoder5_conv2', drray5_2);
lgraph = replaceLayer(lgraph, 'decoder4_conv3', drray4_3);
lgraph = replaceLayer(lgraph, 'decoder4_conv2', drray4_2);
lgraph = replaceLayer(lgraph, 'decoder3_conv3', drray3_3);
lgraph = replaceLayer(lgraph, 'decoder3_conv2', drray3_2);

% Specify the class weights using a |pixelClassificationLayer|.
pxLayer = pixelClassificationLayer('Name','labels','ClassNames',tbl.Name, 'ClassWeights', classWeights)
% % needs additional work!!!!

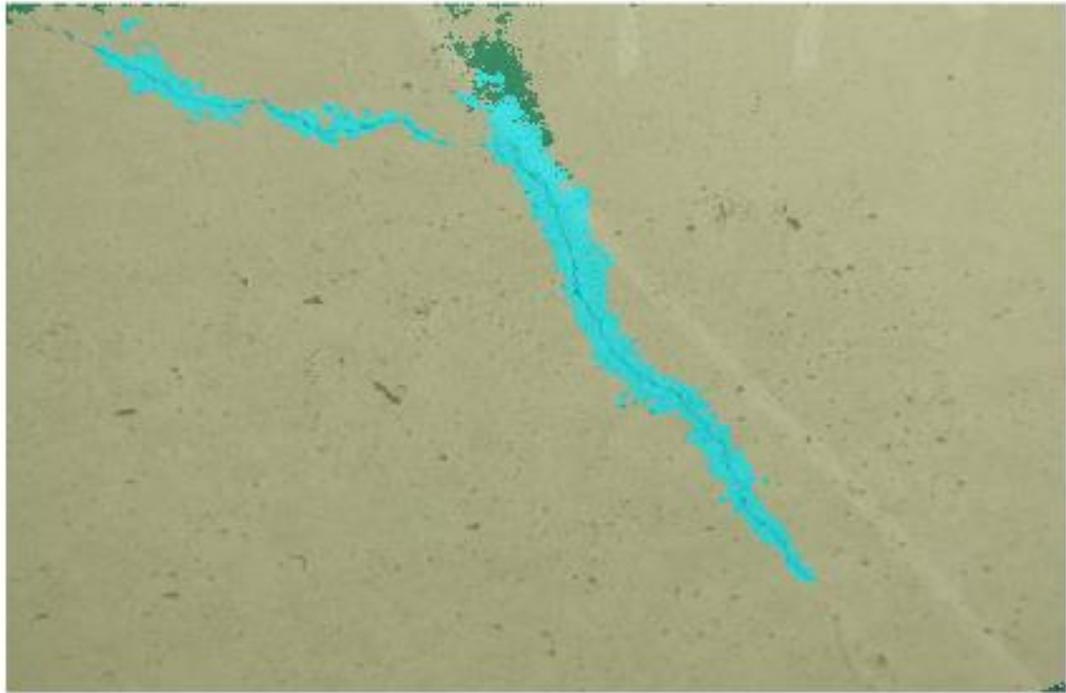
lgraph = removeLayers(lgraph, 'pixelLabels');
lgraph = addLayers(lgraph, pxLayer);
lgraph = connectLayers(lgraph, 'softmax', 'labels');
```

### 3. 정확도 향상을 위한 방안 – Kernel size 변화

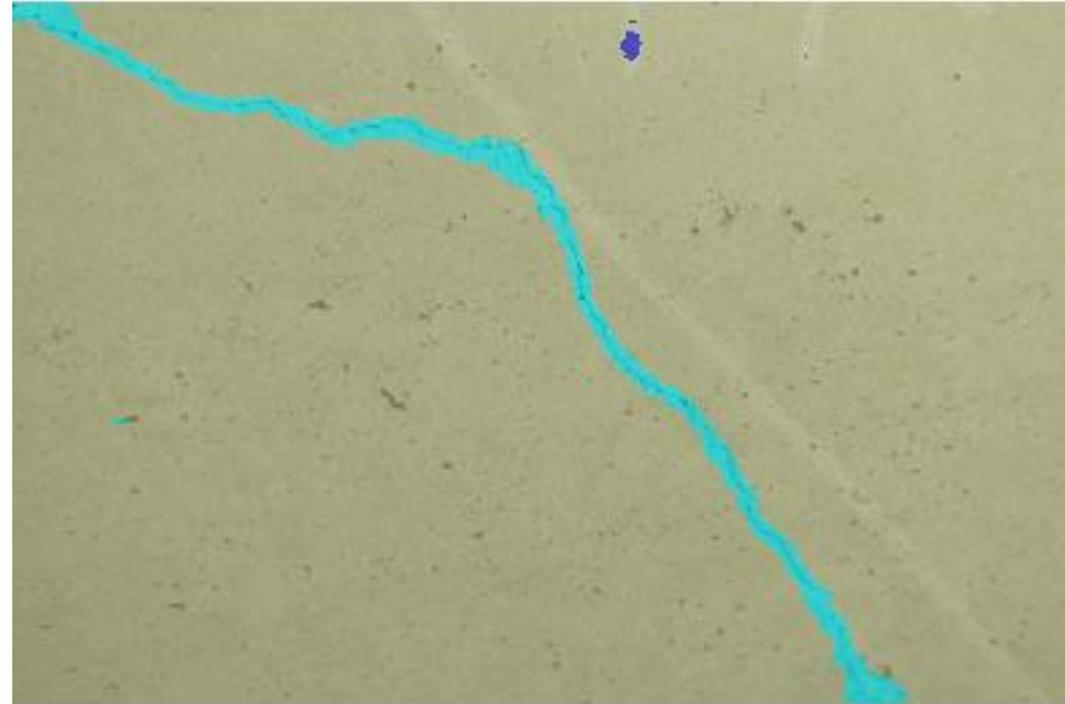
- Concept



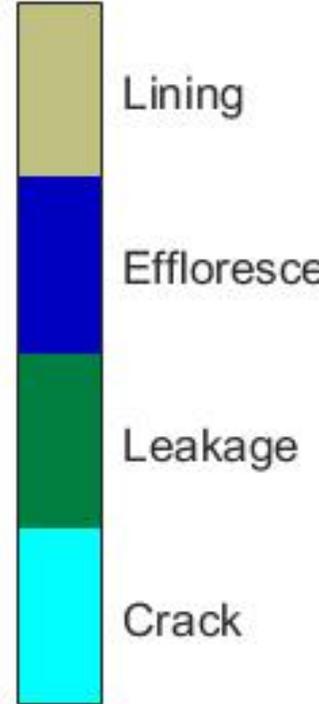
### 3. 정확도 향상을 위한 방안 - 결과



[Segnet(2017)]



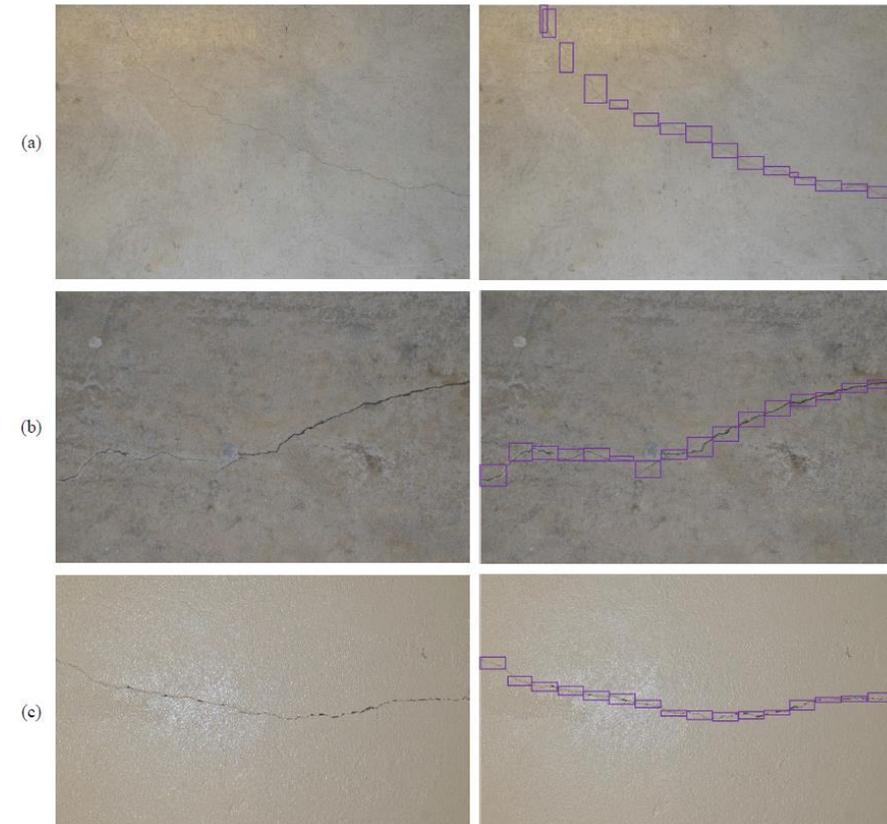
[Model 2]



# Trouble

- 균열 검지의 어려움

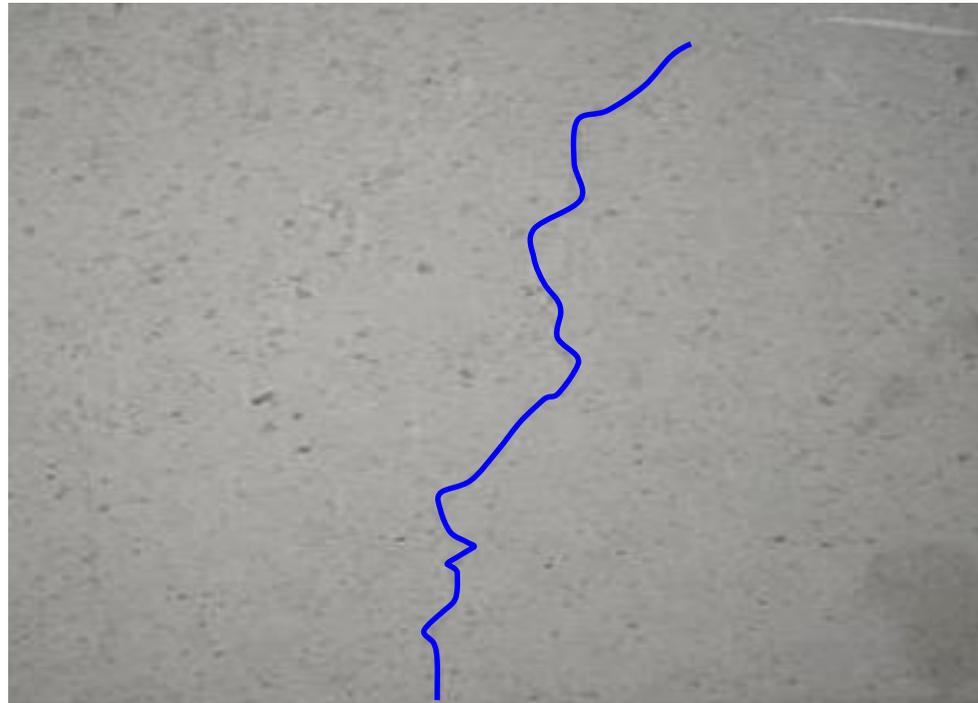
. 균열은 bounding box로 labelling 하기에는 실제 균열의 면적비가 작음



Gahyun Suh, Young-Jin Cha, "Deep faster R-CNN-based automated detection and localization of multiple types of damage," Proc. SPIE 10598, Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018, 105980T (27 March 2018); doi:10.1117/12.2295954

# Trouble

- 균열 검지의 어려움
  - . Pixel labelling의 경우도 labelling의 정밀도에 따라 결과가 달라짐
  - . 균열의 경계를 찾기가 난해



Label | Label Pixels | Zoom In | Zoom Out | Pan | Polygon | Smart Polygon | Smart Polygon Editor | Assisted Freehand | Brush | Erase | Brush Size | Flood Fill | Label Opacity

MODE | POLYGON | FREEHAND | BRUSH | FLOOD FILL | LABEL OPACITY

ROI Label Definition

re\_DSC01793

- Define New ROI Label
- ▶ Ballast
  - ▶ Sleeper
  - ▶ Rail
  - ▶ Eclip
  - ▶ **Crack**
  - ▶ Branch
  - ▶ Stone
  - ▶ Leaf
  - ▶ Cavity



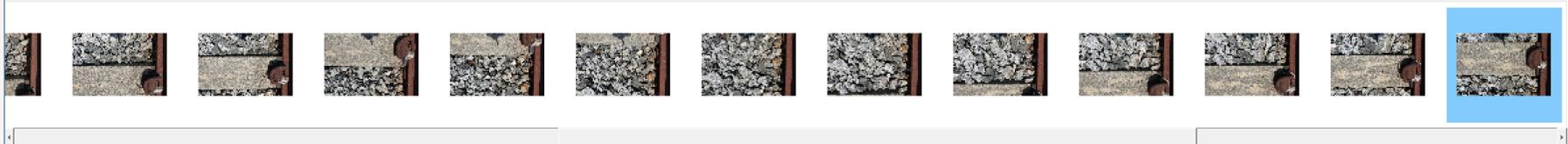
Scene Label Definition

Define new scene label

Apply to Image

Remove from Image

To label a scene, you must first define a scene label.



# Trouble

- Ground Truth Data 확보의 어려움
  - . Bounding Box의 경우는 box 지정이 용이한 편임
  - . Semantic Segmentation은 image labelling 작업이 어렵고 시간소요가 많음
  - . 비록 MATLAB에서 제공하는 자동 labelling tool이 있지만, 아직 실효성은 낮으며, 균열과 같은 특이한 부분을 자동으로 해주지 못함

→ *Image Labeler App에서 Custom algorithm을 넣어서 자동화 가능*

Home Plot Apps

도움말 검색 Caleb님

추가 앱 다운로드 앱 설치 앱 패키지

Curve Fitting Optimization PID Tuner Analog Input Recorder System Identification 신호 분석기 Image Acquisition Instrument Control SimBiology MATLAB Coder Application Compiler Fuel Economy Analysis

C:\Caleb\customer\_support\HS\PixelLabel\_HyunadiSteel

현재 폴더 명령 창

- 이름-
- SNAP-214428-0013.jpg
- SNAP-214417-0012.jpg
- SNAP-214220-0011.jpg
- SNAP-214034-0010.jpg
- SNAP-214001-0009.jpg
- SNAP-213935-0008.jpg
- SNAP-213700-0007.jpg
- SNAP-213004-0005.jpg
- SNAP-212949-0004.jpg
- SNAP-212844-0003.jpg
- SNAP-212736-0002.jpg
- SNAP-212704-0001.jpg
- pixellabel.m
- PixelLabelData
- +vision

SNAP-214428-0013.jpg (JPG File)

작업 공간

이름 -	값

```
fx >>
```

## 진행 중인 연구 방향

- 대량의 데이터 확보
  - . Ground Truth Data 확보가 정밀도 향상에 가장 지름길이라 판단
  - . Time-consuming한 작업이지만 균열 검지, 균열폭 확인을 위해서는 필요한 작업
- Augmentation을 통한 정밀도 향상
  - . 균열은 방향성을 가지고 있기 때문에 직사각형 kernel을 사용할 경우 augmentation에 따른 정밀도 향상을 기대할 수 있음
- Edge detection
  - . 균열 폭을 검지하기 위해서는 edge detection이 필수
  - . how?

# MATLAB을 활용한 Deep Learning 적용은?

## - 사용자 interface 우수

- . Deep learning, Image processing, Machine learning, Statistics 등 다양한 toolbox를 동시에 활용할 수 있어 상호 보완적인 효과

*ex) 기 구축된 DB를 활용할 경우 Label에 대한 전처리를 손쉽게 수행 가능*

- . Embedded app을 통해 사용자 편의성 고려

*ex) Image labeler를 활용하여 별도의 image tool 없이도 작업 가능 (automated tool 등)*

## - Trained network 수정 용이

- . 기존의 network를 불러와서 custom layer를 추가하고 삭제하는 부분이 용이

MATLAB EXPO 2019

경청해주셔서  
감사합니다.

Q&A

