

영상기반 콘크리트 균열 탐지 딥러닝 모델의 유형별 성능 비교

김병현* · 김건순** · 진수민*** · 조수진****†

A Comparative Study on Performance of Deep Learning Models for Vision-based Concrete Crack Detection according to Model Types

Byunghyun Kim* · Geonsoon Kim** · Soomin Jin*** · Soojin Cho****†

†Corresponding Author

Soojin Cho

Tel : +82-2-6490-2434

E-mail : soojin@uos.ac.kr

Received : November 27, 2019

Revised : December 11, 2019

Accepted : December 12, 2019

Abstract : In this study, various types of deep learning models that have been proposed recently are classified according to data input / output types and analyzed to find the deep learning model suitable for constructing a crack detection model. First the deep learning models are classified into image classification model, object segmentation model, object detection model, and instance segmentation model. ResNet-101, DeepLab V2, Faster R-CNN, and Mask R-CNN were selected as representative deep learning model of each type. For the comparison, ResNet-101 was implemented for all the types of deep learning model as a backbone network which serves as a main feature extractor. The four types of deep learning models were trained with 500 crack images taken from real concrete structures and collected from the Internet. The four types of deep learning models showed high accuracy above 94% during the training. Comparative evaluation was conducted using 40 images taken from real concrete structures. The performance of each type of deep learning model was measured using precision and recall. In the experimental result, Mask R-CNN, an instance segmentation deep learning model showed the highest precision and recall on crack detection. Qualitative analysis also shows that Mask R-CNN could detect crack shapes most similarly to the real crack shapes.

Copyright©2019 by The Korean Society of Safety All right reserved.

Key Words : crack detection, deep learning, image classification, object detection, semantic segmentation, instance segmentation

1. 서론

균열은 콘크리트에서 발생하는 가장 대표적 손상 유형으로 구조물 성능 감소로 인한 안전성 저하, 수분 침투에 의한 철근 부식, 외관 해침으로 인한 불안감 유발 등의 원인이 된다. 콘크리트 구조물에서 균열의 관리를 위해서는 정기적인 육안검사를 통해 균열의 위치, 폭, 길이와 같은 정보를 취득하고 이를 외관조사망도 등의 정보화된 형태로 관리하여야 한다. 그러나 일반적으로 건설 구조물들은 규모가 크고 접근성이 높지 않아 육안검사를 이용한 구조물의 손상 탐지에 많은 비용과 시간이 요구되고 있는 실정이다.

육안검사의 어려움 극복을 위해 최근 영상장비와 딥러닝(Deep Learning)을 활용한 균열 탐지 기술들이 매우 활발하게 연구되고 있다^{1,2)}. 딥러닝 기술은 깊은(Deep) 구조의 인공신경망을 이용하여 다수의 이미지 데이터셋(Dataset)으로부터 물체의 주요 특징(Feature)을 컴퓨터 스스로 찾아내고, 그에 따라 물체를 분류하거나 탐지하는 등의 임무(Task)를 수행하는 기술이다³⁾. 즉, 구조물의 표면 이미지를 수집한 뒤 미리 학습된 딥러닝 모델에 대입하면, 컴퓨터가 객관적이고 신속하게 균열을 찾고 정량화한다. 이 경우, 검사자가 직접 균열을 찾는 과정이 필요 없으므로, 검사자의 주관적 판단에 따른 오탐지, 육체적·정신적 피로에 따른 과실, 구

*서울시립대학교 토목공학과 박사과정 (Department of Civil Engineering, University of Seoul)

**서울시립대학교 산업기술연구소 연구원 (Institute for Industrial Technology, University of Seoul)

***서울시립대학교 토목공학과 학사과정 (Department of Civil Engineering, University of Seoul)

****서울시립대학교 토목공학과 조교수 (Department of Civil Engineering, University of Seoul)

조물 접근 환경에 따른 위험도를 최소화할 수 있다.

영상에 활용하는 딥러닝 모델은 물체 분류 모델에서 시작되어 물체 탐지 모델, 형태 세분화 모델 등 다양한 유형으로 진화되었으며, 그에 발맞추어 균열 탐지에도 다양한 딥러닝 유형의 모델들이 활용되었다. 활용결과를 개별적으로 살펴보면, 개발된 모두는 균열 탐지에 충분한 성능을 보이는 것처럼 보인다. 좀 더 정밀한 유형별 성능 비교를 위하여, 본 연구에서는 네 가지 유형(물체 분류 모델, 물체 탐지 모델, 형태 세분화 모델, 인스턴스 세분화 모델)의 딥러닝 모델 중 대표적인 모델(ResNet-101, Faster R-CNN, DeepLab, Mask R-CNN)을 동일한 이미지 데이터셋을 이용하여 학습하고, 40장의 동일한 테스트 데이터에 대해 균열탐지 성능을 비교평가하였다. 각각의 딥러닝 모델이 차이가 있으므로, 학습 방법, 학습종류 기준, 백본 네트워크(Backbone Network), 검증 지표와 같이 동일하게 설정할 수 있는 부분은 최대한 유사하게 설정하였으며, 각 모델의 특성을 고려한 균열 탐지 성능지표를 제안하고 이를 이용하여 비교평가를 수행하였다.

2. 균열 탐지를 위한 딥러닝 모델

2.1 Convolutional Neural Network(CNN)

CNN은 이미지의 특징을 추출하기 위한 컨볼루션(Convolution) 기반 필터링을 신경망과 결합시킨 구조로서, LeCun 등에 의해 처음 제안되었다⁴⁾. 이미지를 특정한 필터 행렬과 컨볼루션할 경우 이미지를 변화시키거나 이미지에 포함되어 있는 특징(Feature)를 추출할 수 있는데, CNN에서는 기존의 신경망에서 주로 사용되던 완전연결계층(Fully Connected Network)에서 뉴런 간 입력-출력의 관계를 설정하는 가중치(Weighting)

대신에 다수의 필터(Filter)를 학습시킴으로서 입력 이미지와 출력간의 관계를 설정하게 된다. 이와 같은 방식은 실제 동물 또는 인간이 물체를 시각을 통해 인지하는 방식과 유사하며⁵⁾, 다수의 층(Layer)를 갖는 신경망 모델에서 학습 변수의 수를 현저하게 감소시키는 장점을 가지고 있다.

2.2 CNN의 유형

CNN은 결과물의 출력 형태에 따라서, 물체 분류(Image Classification) 모델, 물체 탐지(Object Detection) 모델, 그리고 형태 세분화(Semantic Segmentation) 모델, 인스턴스 세분화(Instance Segmentation) 모델로 크게 구분할 수 있다. Fig. 1는 동일한 트럭 이미지에 대한 물체 분류 모델, 물체 탐지 모델, 형태 세분화 모델, 인스턴스 세분화 모델의 결과물 형태를 보여준다.

물체 분류 모델은 일반적으로 고정된 크기의 이미지를 입력받는 입력층(Input Layer)을 가지며, 특징을 추출하기 위한 컨볼루션 층, 특징을 축약하고 강화하기 위한 풀링층(Pooling Layer), 축약된 특징을 분류될 클래스에 매칭하기 위한 완전연결계층 등으로 구성된다. CNN의 초기 형태는 물체 분류 문제를 위해 고안되었다. 모델 구축 방법이 간단하고 하나의 이미지에 하나의 클래스를 부여하여 학습 데이터를 쉽게 생성할 수 있다는 장점이 있다. 대표적인 모델에는 AlexNet⁶⁾, VGG-16⁷⁾ 및 ResNet⁸⁾ 등이 있다.

물체 탐지 모델은 입력 이미지에서 탐지할 클래스에 해당하는 모든 물체의 개별 위치를 찾는다. 물체 탐지 모델은 다수의 슬라이딩 윈도우(Sliding Window)의 크기를 다양하게 변화시켜가며 대상이 될 확률이 가장 높은 부분을 추출한다. 물체 탐지 모델은 물체의 위치를 상자(Box) 형태로 표시하기 때문에 물체의 위치 및

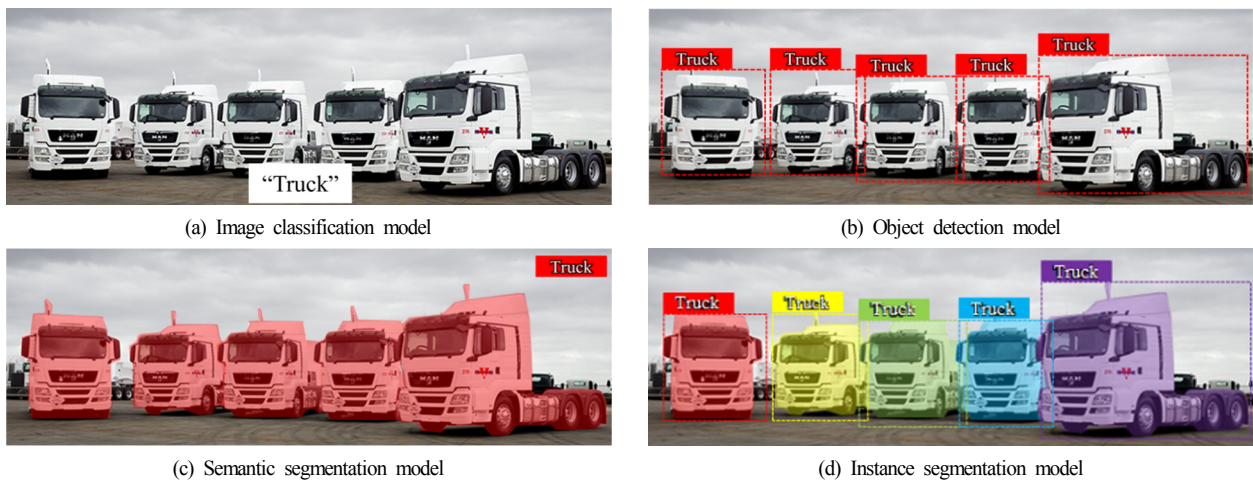


Fig. 1. Example of a truck image analysis according to CNN types.

상대적 크기가 결정되며, 물체의 모양은 판단할 수 없다. 대표적인 모델에는 Faster R-CNN⁹⁾, Single Shot Detector (SSD)¹⁰⁾, YOLO¹¹⁾ 등이 있다.

형태 세분화 모델은 입력 이미지에서 탐지할 클래스의 물체를 모두 찾아서 픽셀 단위로 영역을 지정하며, 동일한 의미의 물체를 탐지하는 특성으로 인해 의미론적 세분화 모델로도 불리운다. 결과를 픽셀 단위로 출력하기 위하여 이미지의 특징을 추출하는 컨벌루션 층과, 특징에 따라 분류된 부분을 Upsampling하기 위한 역컨벌루션(Deconvolution) 층을 활용한다. 물체를 픽셀 단위로 세분화하기 때문에 물체의 형상을 정확하게 추출할 수 있으나, 노이즈에 대한 취약성이 존재한다. 대표적인 모델에는 SegNet¹²⁾, DeepLab¹³⁾, U-Net¹⁴⁾ 등이 있다.

마지막으로, 물체 탐지 모델과 형태 세분화 모델을 결합한 인스턴스 세분화 모델이 있다. 형태 세분화 모델은 특정 클래스의 모든 물체를 찾은 뒤 그 형태를 세분화하나, 인스턴스 세분화 모델은 객체(인스턴스)를 탐지한 뒤 탐지된 객체의 세분화를 수행하는 것이 큰 차이점이다. 이때, 입력된 이미지에 대해 물체의 위치를 탐지하고, 탐지된 물체의 클래스를 분류한 뒤, 물체의 형태를 세분화하는 3단계의 과정을 거친다. 대표적 모델에는 Mask R-CNN (Mask and Region-based Convolutional Neural Network)¹⁵⁾이 있다.

3. 비교 연구를 위한 유형별 딥러닝 모델

3.1 물체 분류 모델: ResNet-101

본 연구에서 사용한 물체 분류 모델은 ResNet⁸⁾이다. ResNet은 2015년 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge)에서 Top-5 error 3.5%를 성취하여 사람의 분류 성능인 5%를 뛰어넘어 큰 주목을 받았다. 기존 신경망은 입력(x)과 출력(y)을 맵핑하는 모델(Hypothesis, H(x))과 출력을 맞추기 위한 학습을 수행하였으나, ResNet은 아래 Fig. 2과 같이 Skipping Layer를 추가한 블록층(Block Layer)을 사용하여 몇 개의 층을 통과한 값(F(x))에 입력(x)을 더해 모델의 출력(H(x)) = F(x)+x)을 결정하고, 모델과 입력의 잔차(Residual, F(x)-H(x))를 최소화하도록 학습한다. 다양한 출력에 모델을 맞추는 기존 학습 목표와 달리 잔차가 0이 되는 뚜렷한 목표가 정해지므로, 깊은 모델에 대해서도 잘 학습이 되는 특징을 갖는다. 이러한 블록층을 여러 개 활용하여 깊지만(Deep) 학습이 잘 되는 모델을 구현하였다.

ResNet-101의 경우 총 101개의 층을 갖는 모델로서,

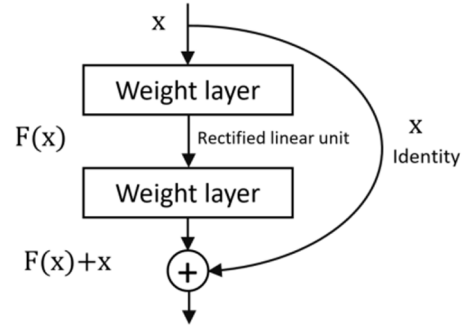


Fig. 2. Single residual block of ResNet⁸⁾.

물체 탐지 모델의 경우 ILSVRC 2015 Dataset에 기 학습되어 있으며, 나머지 유형의 모델들은 COCO Dataset에 대해 기 학습되어 있다. COCO Dataset은 물체의 탐지, 세분화 등을 위해 미리 제작된 80개의 클래스에 대해 라벨링된 20만개 이상의 이미지로 구성된 데이터셋이다¹⁶⁾. 참고로, 아래의 물체 탐지 모델, 물체 세분화 모델에서도 일반적 물체 분류 모델을 함께 활용하는데, 이때 동일한 ResNet-101을 사용하여 모델의 유형간 정량적인 비교를 수행할 수 있도록 하였다.

3.2 물체 탐지 모델: Faster R-CNN

물체 탐지 모델 중 가장 대표적인 모델은 R-CNN (Region-based CNN) 모델이다¹⁷⁾. R-CNN 모델은 이미지에서 Region Proposal 알고리즘인 Selective Search¹⁸⁾를 이용하여 물체가 존재할만한 지역(Region)을 우선 정하고, 각각의 지역에 물체 분류모델을 적용하여 물체의 탐지를 수행한다. 이때, 각 지역에 모두 CNN을 적용하게 되어 속도가 느려지므로, 이를 개선하기 위하여 Fast R-CNN¹⁹⁾, Faster R-CNN⁹⁾이 개발되었다. Faster R-CNN은 빠른 물체 탐지를 위하여 이미지에 CNN을 적용하여 특징지도를 한 번 추출하고, 이 특징으로부터 물체의 위치를 출력받을 수 있는 Region Proposal Network (RPN)라는 추가적인 계층을 학습하여 활용한다. 이때 정확한 물체의 크기 탐지를 위하여 특징지도 상에서 하나의 특정픽셀(Anchor)을 중심으로 다양한 크기와 모

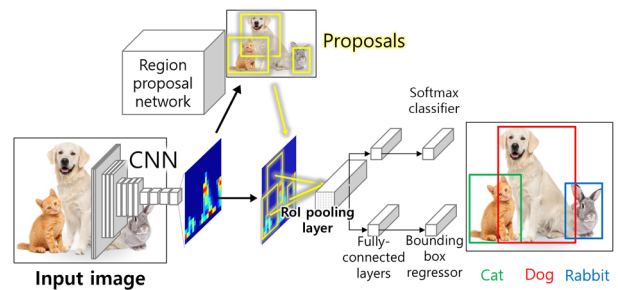


Fig. 3. Overall architecture of Faster R-CNN⁹⁾.

양의 영역(Anchor Box)에 대해 학습을 수행한다. Faster R-CNN 모델은 그 성능과 속도면에서 다른 물체 탐지 모델에 비해 월등함을 보였다²⁰⁾. Fig. 3는 본 연구에서 사용한 Faster R-CNN의 구조를 나타낸다.

3.3 형태 세분화 모델: DeepLab

본 연구에서 활용한 형태 세분화 모델은 DeepLab¹³⁾이다. 형태 세분화를 위해서는 대상이미지에 대해 컨볼루션을 여러 번 적용시켜 특징지도를 얻는 Encoder와, 특징으로부터 정의된 물체의 픽셀을 복원시키기 위한 Decoder를 활용하게 된다. DeepLab에서는 Encoder에 일반적인 CNN에서 활용하는 컨볼루션 대신, 더 넓은 영역에서 특징을 추출하기 위한 Atrous Convolution을 활용함으로써 변수의 수 증가 없이 수용범위(Receptive Field)를 늘리며, 크기가 다른 Atrous Convolution을 병렬로 연산한 뒤 결과를 결합하여 더 정확한 특징지도를 생성하는 Atrous Spatial Pyramid Pooling 기법을 활용한다. Decoder에는 정확한 픽셀 복원을 위하여 Encoder에서 획득한 특징맵과 ASPP를 통해 획득한 특징지도를 결합하고, Bilinear Upsampling함으로써 픽셀을 정확하게 복구한다. Fig. 4는 DeepLab의 구조를 나타낸 그림이다.

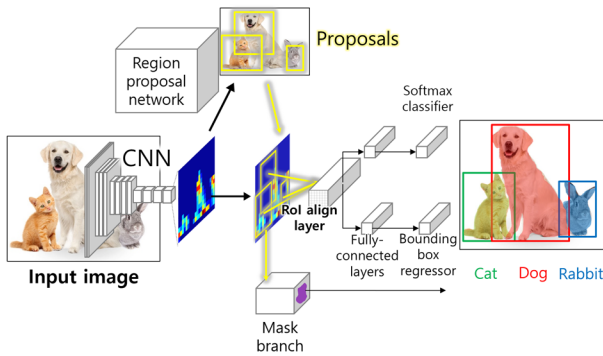


Fig. 4. Overall architecture of DeepLab_v2¹³⁾.

3.4 인스턴스 세분화 모델: Mask R-CNN

Mask R-CNN¹⁵⁾은 물체의 탐지와 세분화를 동시에 진행하는 인스턴스 세분화 모델이다. Mask R-CNN은 기존에 물체 탐지의 목적으로 개발된 Faster R-CNN⁹⁾에 탐지된 물체의 형태를 픽셀단위로 인식하여 세분화하기 위한 추가 층(Mask Branch)를 합성하여 제작되었다. Mask R-CNN은 RPN(Region Proposal Network)을 이용하여 이미지 상에서 찾고자 하는 물체의 후보 영역을 제안(Region Proposal)하고, 각각의 후보 영역으로부터 RoIAlign이라고 하는 쌍일차 보간(Bilinear Interpolation) 모델을 사용하여 물체의 특징지도를 높은 정확도로 생

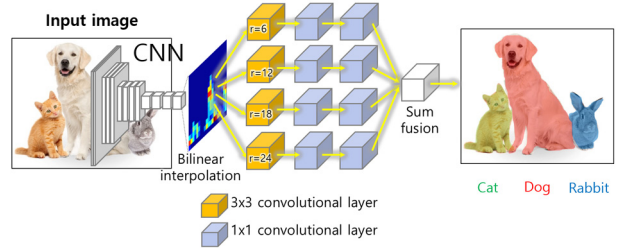


Fig. 5. Overall architecture of Mask R-CNN¹⁵⁾.

성한다. 이후 특징 지도는 Faster R-CNN의 분류층(Classification Layer) 및 경계상자 회귀층(Bounding-Box Regressor Layer)으로 전달되어 물체의 경계상자를 정확하게 탐지하고, 동시에 Mask Branch로 입력되어 경계 상자 내에서 물체의 형태를 결정한다. Fig. 5은 이와 같은 Mask R-CNN 모델의 구조를 나타낸 것이다.

4. 평가 지표

딥러닝 모델의 성능을 평가하기 위한 지표로는 탐지 대상 물체를 놓치지 않고 찾아낼 수 있는지 평가하는 재현율(Recall)과 모델의 예측에 오보(False Alarm)가 없이 정확하게 물체를 찾는지 평가하는 지표인 정밀도(Precision)를 주로 사용한다. 일반적으로 재현율과 정밀도는 다음과 같은 식으로 표현된다.

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

여기에서 TP (True Positive)는 실제 균열을 균열로 탐지한 경우, FN (False Negative)는 실제 균열을 균열로 탐지하지 않은 경우, FP (False Positive)는 실제 균열이 아닌 부분을 균열로 탐지한 경우를 의미한다.

정확한 성능의 비교를 위해서는 그림 2에서 나타난 각 딥러닝 모델의 결과물을 우선 살펴볼 필요가 있다. 영상 분류 모델의 경우, 입력된 이미지 전체에 대해 균열 또는 비균열의 결과를 도출하므로, 균열로 판단하는 영역이 매우 넓게 나타난다. 물체 탐지 모델은 입력된 이미지 내에서 균열이 포함된 영역을 박스의 형태로 도출하므로 영상 분류 모델보다는 작으나 여전히 균열로 판단하는 영역이 매우 넓다. 반면, 형태 세분화 모델이나 인스턴스 세분화 모델은 균열의 형상만을 결과물로 도출하므로, 균열로 판단하는 영역이 매우 좁다. 따라서, 위 식을 그대로 사용하여 재현율과 정밀도를 도출할 경우, 탐지 영역이 크면 클수록 TP가 많아져 재현율

과 정밀도가 높아지게 된다. 본 연구에서는 이를 감안하여 균열 객체와 탐지된 영역의 중심선(Skeleton)을 이용한 수정된 재현율과 정밀도를 제안하였다.

우선, 재현율의 경우 모든 모델에 대해 다음의 식을 이용하였다.

$$Recall = \frac{\#S(D \cap C)}{\#S(C)} \quad (3)$$

여기에서 S(A)는 객체 또는 영역 A의 중심선 픽셀, C는 실제 균열 객체, D는 모델이 균열로 판단한 영역, #은 해당되는 픽셀의 수를 의미한다. 즉, 수정된 재현율은 실제 균열의 중심선 픽셀 수와 모델이 균열로 판단한 영역 내에 포함된 실제 균열의 중심선 픽셀 수의 비율로 계산한다. 이렇게 할 경우, 좁은 영역을 탐지하는 형태 세분화 모델과 인스턴스 세분화 모델이 균열의 형상을 정확하게 탐지하지 못하여 발생하는 재현율의 감소를 보정할 수 있다.

정밀도의 경우에는 넓은 영역을 탐지하는 영상 분류 모델과 물체 탐지 모델에 대해서는 다음의 식을 이용하였다.

$$Precision = \frac{\#S_{\epsilon}(D)}{\#S(D)} \quad (4)$$

여기에서 $S_{\epsilon}(D)$ 는 모델이 균열로 판단한 영역의 중심선 픽셀 중 실제 균열의 중심선 픽셀과의 가장 짧은 거리가 특정 거리(ϵ) 이내로 근접한 중심선 픽셀을 의미한다. 즉, 탐지된 영역의 중심선이 실제 균열의 중심선과 근접한 경우를 식 (2)의 TP로 인정하는 것이다. 본 논문에서는 이미지 장변 픽셀 수의 1/10을 ϵ 으로 하였다. 반면, 좁은 영역을 탐지하는 형태 세분화 모델과 인스턴스 세분화 모델에 대해서는 다음의 식을 이용하였다.

$$Precision = \frac{\#(S(D) \cap C)}{\#S(D)} \quad (5)$$

즉, 모델이 균열로 판단한 영역의 중심선 픽셀 중 실제 균열 객체와 겹치는 픽셀의 수를 TP로 인정하였다.

5. 딥러닝 모델 학습

5.1 학습 데이터 생성

각각의 모델들은 서로 다른 입출력 형식을 가지고 있기 때문에 그에 맞는 학습 데이터 구축이 필요하다. 학습을 위하여 균열 이미지 400장, 검증을 위한 균열 이미지 100장을 확보하였다. 이후 인스턴스 세분화 모델의 학습데이터를 만든 뒤, 이를 축소 또는 변경하는 방법으로 나머지 모델의 학습데이터를 제작하였다.

Fig. 6은 하나의 이미지를 가지고 각 모델의 학습데이터를 제작하는 방법을 설명한다. 먼저 Fig. 6(a)의 원본 균열 이미지에서 Fig. 6(b)와 같이 인스턴스 세분화 모델 학습 데이터를 제작한다. 인스턴스 세분화 모델은 물체 세분화와 탐지를 동시에 수행하므로, 균열 객체에 대해 경계 상자(Bounding Box)와 균열 픽셀 단위의 라벨링을 모두 수행해야 한다. 라벨링은 상용 이미지 편집 프로그램인 Photoshop을 사용하여 이미지 상에서 균열부를 표시하고, COCO dataset 변환용 라이브러리²¹⁾를 사용하여 경계 상자와 균열 픽셀 단위의 라벨링 데이터로 변환하였다. 여기에서 라벨링 정보 중 경계 상자의 정보만을 제거하면 Fig. 6(c)와 같이 물체 세분화 모델을 위한 학습 데이터를 제작할 수 있다. 반대로, 픽셀 단위 라벨링 정보만을 제거하면 Fig. 6(d)와 같이 물체 탐지 모델을 위한 학습 데이터를 제작할 수 있다. 마지막으로 영상 분류 모델을 위한 학습 데이터는 물체 세분화 모델의 학습 데이터를 Fig. 6(e)와 같이 작은 이미지로 쪼개고, 균열이 차지하는 면적이 일정 수준(500 픽셀) 이상인 경우 균열 이미지로, 그렇지 않은 경우 비균열(Non-crack) 이미지로 분류하여 균열 이미지, 비균열 이미지 각 2000장씩을 제작하였다. 제작된 균열 및 비균열 이미지는 사용된 ResNet-101의 이미지 입력 크기인 224×224로 조정되어 저장되었다. 균열 이미지와 비균열 이미지는 각각 별도의 폴더에

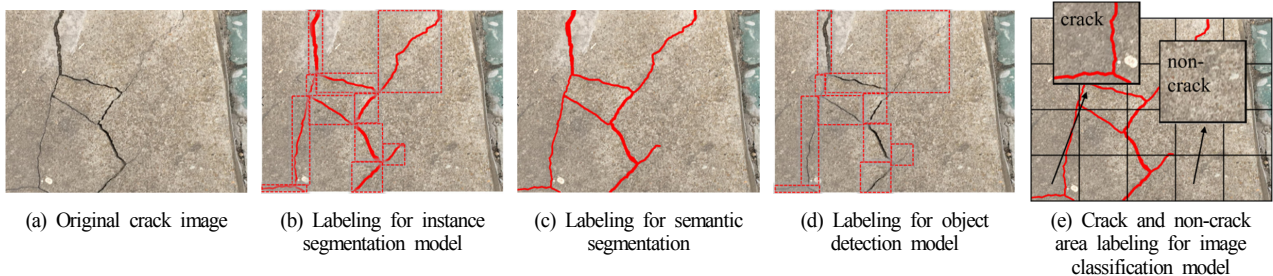


Fig. 6. Crack image labeling examples for each type of deep learning model.

Table 1. Summary of training dataset and training result for each type of deep learning model

| | | Image classification | Object detection | Semantic segmentation | Instance segmentation |
|---|-------------------------|---|--------------------|-----------------------|-----------------------|
| No. of Total training images (Training set / Validation set) | | 2000 (1400 / 600) | 500 (400 / 100) | 500 (400 / 100) | 500 (400 / 100) |
| Applied image augmentation | | Flipping up and down, Flipping left and right | | | |
| Training result | Train accuracy (%) | 100 | 96.2 | 95.3 | 95.9 |
| | Validation accuracy (%) | 98.8 | 94.6 | 96.2 | 95.7 |

저장되었으며, 각 폴더의 이름에 따라 입력 이미지의 클래스를 판단하도록 프로그래밍을 구현하였다. 학습의 정확도 향상을 위하여 상하 및 좌우 반전의 이미지 증강(Augmentation) 기술을 통해 수집된 학습데이터의 4배의 데이터를 실제 학습에 사용하였다. Table 1은 학습데이터에 대해 정리한 것이다.

5.2 학습 수행

학습을 위해 2개의 Intel Xeon 프로세서(E5-2620 v4) 과 4개의 NVIDIA GTX 1080Ti 그래픽 카드가 장착된 워크스테이션을 사용하였다. 각각의 모델은 과적합(Overfitting)을 최소화하기 위하여 학습정확도와 검증정확도가 90%를 넘어서면서 가장 비슷해질 때의 모델을 최종적인 학습 모델로 선정하였다. Table 1에서 보듯, 모든 모델에서 94% 이상의 학습 및 검증 정확도를 나타내어, 충분한 학습이 이루어졌음을 확인하였다.

6. 딥러닝 모델 성능 비교 분석

실제 현장환경에서 수집된 40장의 이미지를 사용하여 학습된 네 가지 딥러닝 모델의 성능 비교를 수행하

였다. Fig. 7은 40장의 이미지 중 한 장의 이미지에 대해 각각의 모델을 이용하여 탐지한 균열 영역을 붉은 색으로 표시하여 비교한 것이다. 40장의 이미지들은 Fig. 7(a)에서와 같이, 균열 탐지 성능의 저하를 유발하는 다양한 균열 이외의 물체들을 포함한 사진들로 선별되었다.

Fig. 7(b)에서 보듯, 영상 분류 모델은 하나의 이미지에 하나의 분류만을 부여하기 때문에 모델 구현이 비교적 간단하며, 균열 주변을 넓게 탐지하는 특징을 갖는다. 이 과정에서 하나의 이미지를 분할하여 모델에 반복적으로 적용하여야 하므로, 연산 속도는 가장 느리다. 넓은 영역을 탐지하므로 균열의 유무는 잘 탐지할 수 있으나, 균열의 정확한 위치와 형상을 추출하기 위해서는 더욱 정교한 작업이 필요한 단점을 가지고 있다. Fig. 7(d)에서 보듯, 물체 탐지 모델은 본래 박스 형태로 물체를 탐지하기 때문에 균열을 탐지할 때는 균열 주변부를 연속된 경계 상자 형태로 표시하여 탐지하게 된다. RPN을 사용하여 영상 분류 모델보다는 더 빠르고 정교한 영역의 추출이 가능하지만, 마찬가지로 균열의 정확한 위치와 형상을 추출하기 위해서는 추가적으로 균열영역만을 획득하는 과정이 필요하다.

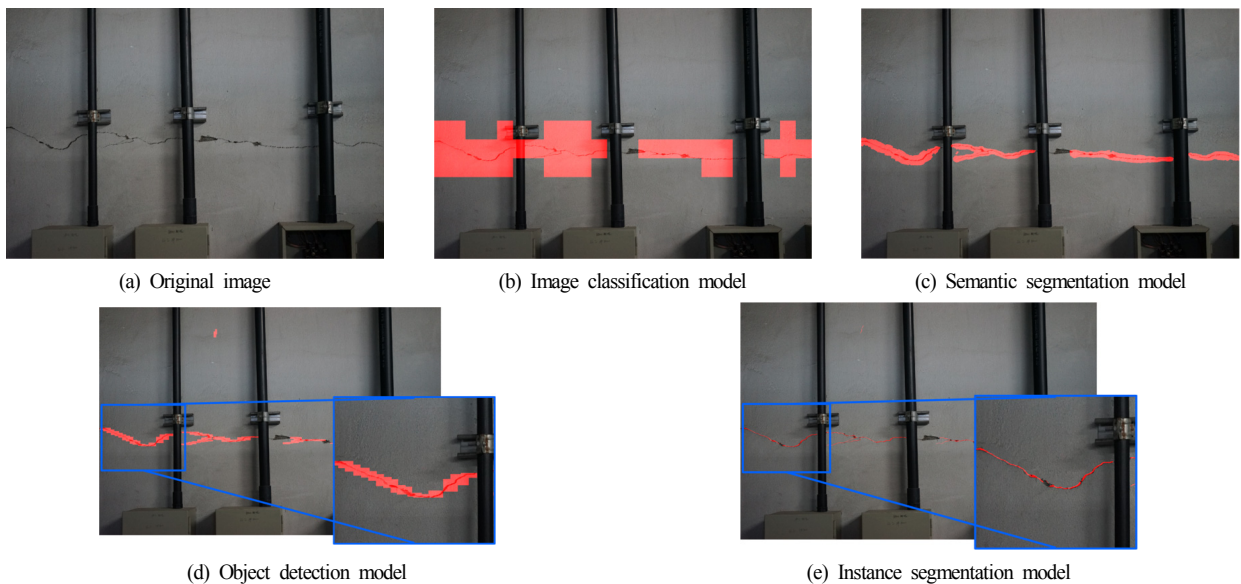


Fig. 7. Example of crack detection results for each type of deep learning model.

Table 2. Qualitative comparison of the crack detection result

| | Image classification | Object detection |
|---------------|--|---|
| Advantages | Simple implementation of a model | Faster detection process with region proposal network |
| Disadvantages | Slow detection process due to sliding window manner | rectangular detection result due to bounding box |
| | Semantic segmentation | Instance segmentation |
| Advantages | Pixel-wise crack detection result display | Pixel-wise crack detection with low noise rate |
| Disadvantages | Higher noise rate because every pixel has to be classified | Complicate network architecture |

Fig. 7(c)에서 보듯, 물체 세분화 모델은 픽셀 단위로 균열 분포를 획득하여 이미지에 표시한다. 하지만, 균열의 형상을 정확하게 추출하는 성능은 높지 않은데, 이는 전체 이미지의 픽셀에 대해 오차를 계산하는 특성 때문이다. Fig. 7(e)에서 보듯, 인스턴스 세분화 모델은 균열부를 정교하게 추출한다. 이는 우선 물체 탐지 모델에 의해 균열의 영역을 경계 상자의 형태로 획득한 뒤 상자 내부에서 픽셀 단위로 균열의 형상을 추출하므로, 형태 세분화 모델보다 픽셀 단위로 분류해야 하는 영역이 작아지기 때문이다. 인스턴스 세분화 모델이 물체탐지 모델과 물체 세분화 모델이 결합된 형태이기 때문에 비교적 구현이 어렵지만, 더 정교한 균열 탐지 성능을 보이는 것을 확인할 수 있다. 본 단락의 정성적인 모델별 균열 탐지 결과 비교는 Table 2에 정리되어 있다.

Table 3은 40장의 이미지에 대하여 4장에서 서술한 수정된 정밀도와 재현율을 계산한 결과이다. 먼저 영상 분류 모델은 실험에 사용된 40장의 이미지에 대하여 89%의 재현율과 68%의 비교적 낮은 정밀도를 보였다. 이는 넓은 영역을 탐지하는 모델의 특성상, 실제 균열을 탐지해내는 성능인 재현율은 높은 대신 정확하게 탐지하는 능력인 정밀도가 낮게 나타났음을 알 수 있다. 물체 분류 모델의 경우에는 재현율 85%, 정밀도 86%로서, 영상 분류모델 대비 좁은 탐지영역으로 인해 낮은 재현율과 높은 정밀도가 나타났음을 알 수 있다. 형상 세분화 모델의 경우 재현율 89%, 정밀도 75%를 보였으며, 이는 균열의 위치를 잘 찾아내었으나 균열의 형상 추정에서는 낮은 성능을 보임을 나타낸다. 마

Table 3. Crack detection result using 40 images

| Measure | Image classification | Object detection | Semantic segmentation | Instance segmentation |
|--------------|----------------------|------------------|-----------------------|-----------------------|
| Recall(%) | 89 | 85 | 89 | 94 |
| Precision(%) | 68 | 86 | 75 | 84 |

지막으로 인스턴스 세분화 모델은 재현율 94%, 정밀도 84%로 균열의 위치뿐만 아니라 균열의 형상 추정도 매우 정확하게 할 수 있음을 나타낸다.

7. 결론

본 연구에서는 최근 이미지 분석 분야에서 활발하게 활용되는 딥러닝 모델들을 크게 4가지 유형(이미지 분류 모델, 물체 탐지 모델, 형태 세분화 모델, 인스턴스 세분화 모델)으로 분류하고, 각 유형의 대표적인 모델(ResNet-101, Faster R-CNN, DeepLab, Mask R-CNN)들의 균열 탐지에 대한 성능을 비교 분석하였다. 총 40장의 현장 이미지를 이용하여 성능을 비교한 결과를 정리하면 다음과 같다.

1) 각 모델의 균열 탐지영역이 다름을 감안하여, 균열 객체와 탐지된 영역의 중심선(Skeleton)을 이용한 수정된 재현율과 정밀도를 제안하고, 이를 이용하여 균열의 위치와 형상을 탐지하는 성능을 평가하였다.

2) 40장의 이미지에 대해 인스턴스 세분화 모델이 가장 높은 수준의 재현율(94%)과 정밀도(84%)를 보여주었다. 특히 균열의 위치를 경계 상자의 형태로 선택 후 형상을 세분화하는 모델의 특성으로 인해, 균열의 형상을 매우 근사하여 탐지할 수 있음을 보였다.

3) 모든 딥러닝 모델들은 85% 이상의 높은 재현율을 나타내어, 현장에서 실제 균열의 위치를 탐지하는 데에는 모든 모델이 잘 활용될 수 있음을 보였다.

4) 영상 분류 모델과 형태 세분화 모델의 경우 70% 전후의 낮은 정밀도를 보여, 균열의 형상을 정확하게 추정하는 성능이 타 모델보다 낮은 것으로 나타났다.

5) 본 결과는 균열 폭이 적정한 해상도(2-3픽셀 이상)를 갖는 이미지에 대해 획득한 결과로서, 좋은 탐지 결과를 얻기 위해서는 근접촬영이나 줌렌즈를 활용한 촬영을 통해 고해상도 이미지를 획득할 필요가 있다. 또한, 균열 폭 및 길이의 정량화를 위해서는 촬영거리를 측정하여 픽셀 길이를 실제 길이로 변환하여야 한다.

감사의 글: 본 연구는 2019년도 정부(과학기술정보통신부)의 재원으로 한국정보통신진흥원의 지원을 받아 수행되었습니다(과제명-국가인프라 지능정보화사업: 인공지능 기반 시설물 안전진단).

References

1) K. Jang et al., "Deep Learning-Based Autonomous Concrete Crack Evaluation through Hybrid Image

- Scanning”, *Structural Health Monitoring*, Vol. 18, No. 5-6, pp. 1722-1737, 2019.
- 2) C. V. Dung and D. A. Le, “Autonomous Concrete Crack Detection Using Deep Fully Convolutional Neural Network”, *Automation in Construction*, Vol. 99, pp. 52-58, 2019.
 - 3) J. Schmidhuber, “Deep Learning in Neural Networks: An Overview”, *Neural Networks*, Vol. 61, Elsevier Ltd, pp. 85-117, 2015.
 - 4) Y. LeCun et al., “Gradient-Based Learning Applied to Document Recognition”, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278-2323, 1998.
 - 5) K. Fukushima, “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, *Biological Cybernetics*, Vol. 36, No. 4, Springer-Verlag, pp. 193-202, 1980.
 - 6) A. Krizhevsky et al., “ImageNet Classification with Deep Convolutional Neural Networks”, *Advances In Neural Information Processing Systems*, pp. 1-9, 2012.
 - 7) K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, *International Conference on Learning Representations (ICRL)*, pp. 1-14, 2015.
 - 8) K. He et al., “Deep Residual Learning for Image Recognition”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.
 - 9) S. Ren et al., “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, pp. 1137-1149, 2017.
 - 10) W. Liu et al., “SSD: Single Shot Multibox Detector”, *European Conference on Computer Vision*, Vol. 9905 LNCS, pp. 21-37, 2016.
 - 11) J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.
 - 12) V. Badrinarayanan et al., “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 12, pp. 2481-2495, 2017.
 - 13) L. C. Chen et al., “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, pp. 834-848, 2018.
 - 14) O. Ronneberger et al., “U-Net: Convolutional Networks for Biomedical Image Segmentation”, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Vol. 9351, pp. 234-241, 2015.
 - 15) K. He et al., “Mask R-CNN”, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961-2969, 2017.
 - 16) T. Y. Lin et al., *Microsoft COCO: Common Objects in Context*, 2014.
 - 17) R. Girshick et al., “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580-587, 2014.
 - 18) J. R. R. Uijlings et al., “Selective Search for Object Recognition”, *International Journal of Computer Vision*, Vol. 104, No. 2, Springer US, pp. 154-1571, 2013, doi: 10.1007/s11263-013-0620-5.
 - 19) R. Girshick, “Fast R-CNN”, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448, 2015.
 - 20) J. Huang et al., “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors”, *The IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp. 7310-7311, 2017.
 - 21) GitHub - Waspinator/Pycococreator: Helper Functions to Create COCO Datasets. <https://github.com/waspinator/pycococreator>. Accessed 8 Aug. 2019.