

시설물의 유지관리를 위한 기계학습 기반 콘크리트 균열 감지 프레임워크

Machine Learning-based Concrete Crack Detection Framework for Facility Maintenance

지 봉 준[†]
Bongjun Ji

Received: August 6th, 2021; Revised: August 13th, 2021; Accepted: September 25th, 2021

ABSTRACT : The deterioration of facilities is an unavoidable phenomenon. For the management of aging facilities, cracks can be detected and tracked, and the condition of the facilities can be indirectly inferred. Therefore, crack detection plays a crucial role in the management of aged facilities. Conventional maintenances are conducted using the crack detection results. For example, maintenance activities to prevent further deterioration can be performed. However, currently, most crack detection relies only on human judgment, so if the area of the facility is large, cost and time are excessively used, and different judgment results may occur depending on the expert's competence, it causes reliability problems. This paper proposes a concrete crack detection framework based on machine learning to overcome these limitations. Fully automated concrete crack detection was possible through the proposed framework, which showed a high accuracy of 96%. It is expected that effective and efficient management will be possible through the proposed framework in this paper.

Keywords : Machine learning, Deep learning, Crack detection, Data analytics, Maintenance

요 지 : 시설물의 노후화는 피할 수 없는 현상이다. 노후화된 시설물의 관리를 위해 균열을 감지하고 이를 추적하면서 시설물의 상태를 간접적으로 추론할 수 있다. 따라서 균열 감지는 노후화된 시설물의 관리를 위해 필수적 역할을 하며 감지 결과를 바탕으로 더 이상의 노후화를 막기 위한 활동을 할 수 있다. 하지만, 현재 대부분의 균열 감지는 전문가의 판단에만 의존하기에 시설물의 면적이 큰 경우 비용과 시간이 과도하게 사용되고, 전문가의 역량에 따라 다른 판단 결과가 발생할 수 있어 신뢰성에 문제가 있었다. 본 논문에서는 이러한 한계를 극복하기 위해 기계학습 기반의 콘크리트 균열 감지 프레임워크를 제안한다. 제안된 프레임워크는 데이터 분류, 기계학습 모델 학습, 학습된 모델의 검증과 테스트를 포함하는 프레임워크로 완전 자동화된 콘크리트 균열 감지가 가능하다. 제안된 프레임워크를 통해 학습된 기계학습 모델은 콘크리트 균열 이미지와 정상 이미지를 96%의 높은 정확도로 분류할 수 있었다. 본 논문에서 제안된 프레임워크를 적용하여 기존의 전문가 중심의 시설물 유지관리보다 더욱 효과적이고 효율적인 시설물의 유지관리가 가능할 것으로 기대된다.

주요어 : 기계학습, 기계학습, 균열 감지, 데이터 분석, 유지관리

1. 서 론

시설물의 손상은 피할 수 없는 현상으로 시간의 따른 재료의 열화나 기후와 기상의 변화 등으로 발생한다. 시설물의 손상은 붕괴와 같은 치명적인 사고를 유발시켜 재산상, 인사상 손해를 발생시킬 수 있으므로 중요하게 관리되어야 한다. 특히 노후화된 시설물의 비율은 나날이 증가하고 있으며 사용 과정에서 증축이나 용도변경 등으로 처음 건설되었을 당시보다 안전에 취약할 수 있다(Kim, 2020). 균열은 시설물의 손상을 추론하기 위해 널리 사용된다(Chun et al., 2021). 예를 들어 정기점검 가이드라인에서는 육안으로 균

열을 검출하여 시설물의 상태를 평가한다(Ministry of Land, Infrastructure, Transport and Tourism., 2019). 하지만 시설물의 면적이 큰 경우, 사람이 모든 곳을 검사하기 쉽지 않아 이로 인해 시간과 비용이 많이 소요된다. 또한, 사람의 경우 역량 및 컨디션에 따라 판단 결과가 달라지기 때문에 휴먼 에러로 인해 검출에 실패할 수 있다. 또한 점검자가 변경될 경우, 손상의 진행 여부에 대한 모니터링이 힘들고, 이루어진다 하더라도 객관성과 신뢰성을 보장할 수 없다. 이러한 문제를 극복하기 위해서 다양한 연구들이 존재했다. 객관성과 신뢰성을 확보하기 위해 촬영된 이미지를 활용하는 방법이 70년대부터 개발되기 시작했다(Luxmoore, 1973). 초기

[†] PhD, Industrial and Management Engineering, Pohang University of Science and Technology (Corresponding Author : bongjun_ji@postech.ac.kr)

연구는 열악한 화질의 사진과 컴퓨팅 파워의 한계로 진전을 이루지 못하였다. 하지만 2000년대 들어 하드웨어와 소프트웨어 모두의 발전과 더불어 이미지 가공 분야에 급격한 발전이 이루어졌다. Frequency 기반의 필터(Abdel-Qader et al., 2003)나 Wavelet transform과 같은 변환 기법(Hutchinson & Chen, 2006), 기하학적 특징을 찾아내는 기법(Liu et al., 2019), Neuro-fuzzy hybrid computing structures 등이 소개되었고 좋은 성능을 보였다. 이와 같은 규칙 기반의 기법들은(Zhao & Chen, 2002) 일반화하기 어렵고, 특정 조명과 각도를 요구하는 등 조건이 까다로운 한계가 있었다. 이러한 한계를 극복하기 위하여 기계학습 기반의 기법들이 소개되기 시작했다. 기계학습 기반의 기법은 사람이 규칙을 찾을 필요가 없고 환경에 영향을 적게 받는 강건한 모델을 만들 수 있어 이 전의 규칙 기반의 기법들보다 좋은 성능을 보였다. 기계학습 기반의 기법들 중 Support vector machine (Prasanna et al., 2016), k-nearest neighbors(Zhang et al., 2014), Neural network(Choudhary et al., 2012) 등 일반적인 기계학습 기법부터 여러 기계학습 모델을 동시에 사용하는 앙상블 기반의 방법인 Random Forest(Shi et al., 2016)나 AdaBoost(Prasanna et al., 2014) 등 다양한 방법론이 있었다. 최근 들어 대량의 이미지를 처리할 수 있는 기계학습 기법이 소개되며, 다층의 Layer로 구성된 기계학습 기법들도 적용되었는데 픽셀간 종속성을 잘 학습하여 이미지 처리에 널리 사용되는 CNN(Convolutional Neural Network) 기반의 방법들이 널리 적용되었다(Kim et al., 2018). 다만 기존 관련 연구들은 널리 적용된 아키텍처와 가중치(Weight)를 그대로 사용하거나, Feature를 선정하는데 사용하였고 예측과정의 해석이나 아키텍처의 수정, 보완을 통한 접근은 거의 이루어지지 않았다. 기계학습에서 Feature란 결과값에 영향을 미치는 변수, 특성으로 입력값 자체가 Feature일 수도 있고, 연산을 통해 변환된 입력값들 또한 Feature가 될 수 있다.

본 논문에서는 이미지 프로세싱에 널리 사용되는 기계학습 아키텍처에 앙상블 기법을 적용하여 콘크리트 균열을 감지하고자 한다. 또한, 균열을 감지하는 과정을 분석하여, 기계학습이 균열의 어떤 점을 주로 감지하는지 분석하여, 기계학습 결과를 해석하고자 한다. 즉, 앙상블 기법을 적용한다는 점과, 그 해석을 시도한다는 점에서 기존 연구들과 차별점이 있다. 본 논문은 다음과 같이 구성된다. 챕터 2에서는 기계학습의 기초가 되는 CNN의 구조와 여러 모델들을 결합하여 값을 예측하는 앙상블 방법에 대해 소개한다. 챕터 3에서는 콘크리트 균열을 감지하기 위한 프레임워크를 소개한다. 프레임워크는 개념적인 내용을 포함하고 있다. 챕터 4에서는 챕터 3에서 제안된 프레임워크를 실제 데이터 셋에 적용하여 기계학습 모델을 학습하고 성능을 평가한다.

챕터 4는 실험 결과에 대한 고찰을 포함한다. 챕터 5에서는 연구결과를 요약하고, 본 논문이 어떤 점에서 공헌하고 있는지 서술하며, 앞으로의 연구 방향을 제안하고자 한다.

2. 이론적 배경

이 챕터에서는 이미지 기계학습의 기반이 되는 CNN과, 여러 기계학습 모델을 사용하여 시너지를 얻을 수 있는 앙상블 기법에 대해 논의하고자 한다.

2.1 CNN

LeCun et al.(1989)이 제안한 CNN은 이미지 인식 관련 어플리케이션에서 좋은 성능을 보였다. CNN은 기본적으로 Convolutional layer, Pooling layer, Fully-connected layer 등 세 가지의 레이어로 구성된다. Convolutional layer는 합성곱을 실행하는 layer로 정해진 필터를 이미지에 합성곱시켜 Feature를 추출한다(Fig. 1).

Pooling layer는 대체로 Convolutional layer를 통해 추출된 Feature에 적용된다(Fig. 2). Max pooling과 Average pooling 등이 널리 사용되는 방법으로 Max pooling은 Feature 중 가장 큰 값을, Average pooling은 평균값을 Feature로 추출한다.

이와 같은 Convolutional layer와 Pooling layer를 겹겹이 쌓아 추출된 Feature를 통해 값을 예측하는 것이 CNN의 원리이다. Layer의 구성 외에도 기존 널리 사용되던 MLP (Multi-Layer Perceptron)와의 차이점이 존재하는데, MLP는

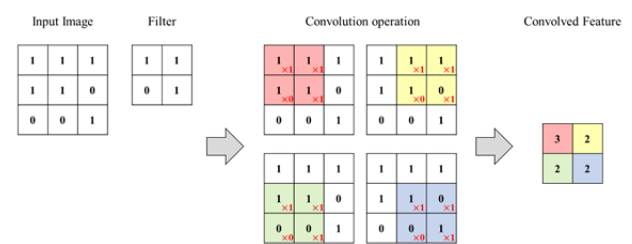


Fig. 1. Convolution operation in convolution layer

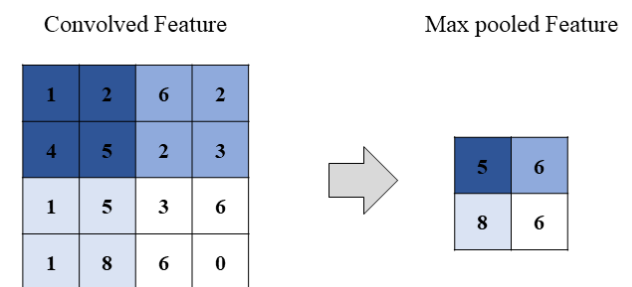


Fig. 2. Max pooling operation

모든 노드들이 연결되어 있는 Fully connected 상태라면, CNN은 일부만 연결되어 있다. VGG-16(Simonyan & Zisserman, 2014)은 CNN의 파생 아키텍처로 CNN과 마찬가지로 Convolutional layer, Pooling layer, Fully-connected layer로 구성되어 있다(Fig. 3). 해당 아키텍처는 1,400만개 이상의 이미

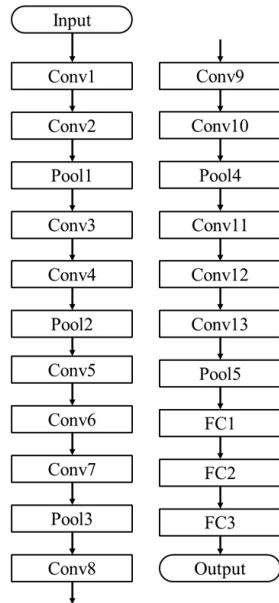


Fig. 3. Architecture of VGG-16

지에서 1,000여개의 클래스를 분류하는 컴페디션인 ImageNet Large Scale Visual Recognition Challenge(ILSVRC)에 발표되어 널리 알려진 모델로 아스팔트 바인더의 물성 예측(Ji et al., 2020), 불량 농작물의 감지(Xie et al., 2021), 제품의 불량 감지(Chen et al., 2020), 회로의 결함 감지(Cheong et al., 2019) 등 다양한 응용분야에서 높은 성능을 보였다. 따라서 본 논문에서는 VGG-16을 기초 모델로 선정하였다.

2.2 앙상블 기법

앙상블 기법은 여러 예측 모델을 결합하여 더 좋은 성능의 예측 모델을 만드는 기법을 뜻한다. 앙상블 기법은 결합하는 방식에 따라 Bagging과 Boosting으로 나뉜다. Bagging은 결합하는 모델 각각을 독립적으로 학습하고, 각 모델을 병렬로 결합하여, 출력된 예측들을 모아 투표를 하는 개념이다(Fig. 4). 대표적인 기법으로는 Random Forest가 존재하는데, 각각의 Decision tree를 학습, 예측값을 산출한 후, 다수의 답을 예측값으로 산출해낸다. 두 번째 방법인 Boosting 기법은 순차적인 방법으로, 다수의 예측 모델을 각각 학습하는 Bagging과 달리, Boosting은 순차적으로 학습을 진행한다. 이 전 모델에서 예측에 실패한 값을 잘 예측할 수 있도록 가중치를 주어 학습시킨다. Fig. 4는 이들의 차이를 개념적으로 설명한다.

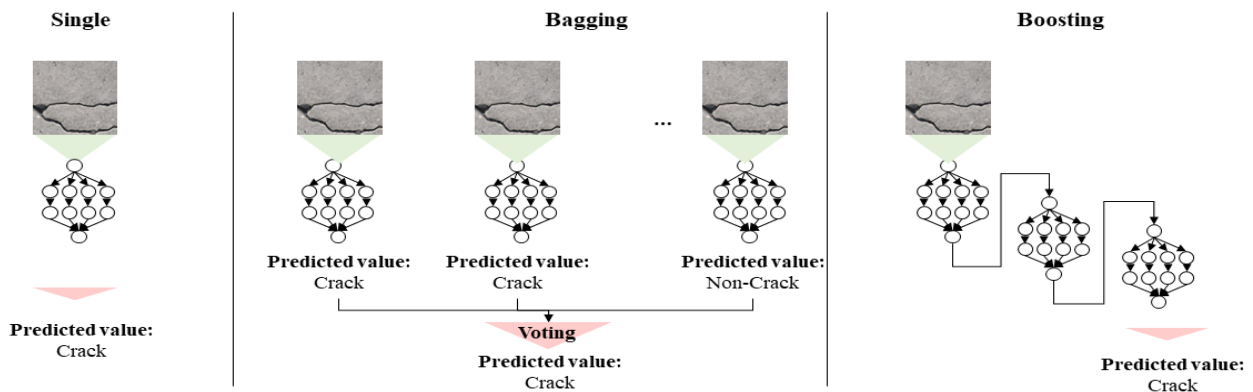


Fig. 4. Concept of Boosting and Bagging

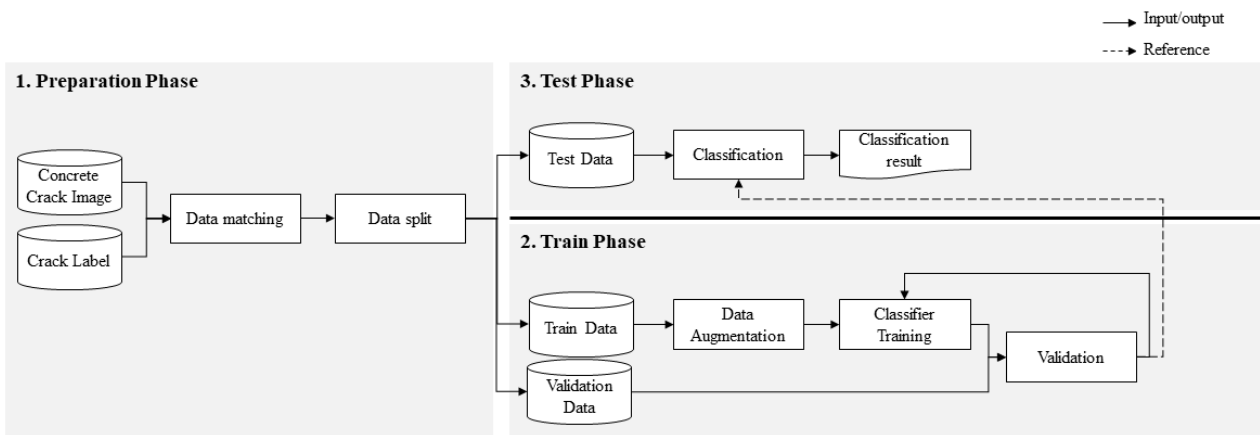


Fig. 5. Proposed framework

3. 프레임워크

본 연구에서 제안하는 프레임워크는 크게 세 단계로 구성된다. 첫 번째는 준비 단계, 두 번째 단계는 학습 단계, 세 번째 단계는 테스트 단계이다(Fig. 5).

3.1 준비 단계

본 섹션에서는 균열 감지 모델을 학습하기 전에 데이터셋을 준비하는 단계를 다룬다. 데이터셋 준비를 위해 각 이미지는 라벨과 매칭이 된다. 즉, 주어진 이미지가 균열 이미지인지, 그렇지 않은 정상 상태의 이미지인지 매칭하는 작업이다. 이 작업을 통해 기계학습 모델은 이미지가 주어졌을 때 정답(라벨)이 무엇인지 학습하게 된다. 데이터 매칭 작업이 끝난 후에는 데이터를 나누는 작업을 진행한다. 본 논문에서 제안하는 기계학습을 위한 데이터셋은 세 가지로 구분된다. 첫 번째는 학습 데이터셋으로, 기계학습 모델의 학습에 사용되는 데이터셋이다. 학습 데이터셋은 직접적으로 기계학습의 학습(파라미터 산출)에 사용되는 데이터셋으로, 랜덤으로 선택된 데이터의 70%가 학습 데이터로 선정된다.

두 번째 데이터는 검증(Validation) 데이터셋이다. 검증 데이터셋은 모델의 학습에 사용되지 않고, 학습된 모델의 성능 평가에 사용된다. 해당 데이터셋은 학습에 사용되지 않는 데이터이므로 과적합(Ovefitting)을 방지하고, 입력되지 않은 데이터에 대한 정확도를 가늠할 수 있다. 마지막으로, 테스트 데이터셋은 개발된 모델의 성능을 최종적으로 평가하기 위해 사용되는 데이터셋이다. 검증 데이터셋은 모델의 학습 방향에 대한 가이드를 제공하는 것이라면, 테스트 데이터셋은 최종적인 성능 평가를 위해 사용되는 점에서 차이가 있다.

3.2 학습 단계

본 단계에서는 기계학습 모델을 학습한다. 기계학습 모델의 학습이란 기계학습 모델을 구성하는 가중치와 편향(Bias)의 값을 찾는 것으로, 이들은 아키텍처 내의 노드와 노드 사이에 위치해있다. 가중치와 편향은 데이터가 입력되었을 때 학습된 가중치와 편향의 연산을 통해 예측값(균열/정상)을 산출할 수 있도록 한다. 본 연구에서는 학습 단계를 데이터 증대(Augmentation)과 기계학습 모델인 Classifier를 학습하는 것으로 구성한다.

3.2.1 데이터 증대

데이터 증대는 학습 데이터의 양을 늘리기 위한 기술로 학습 데이터에 조금의 수정을 가하는 방식으로 학습 데이터

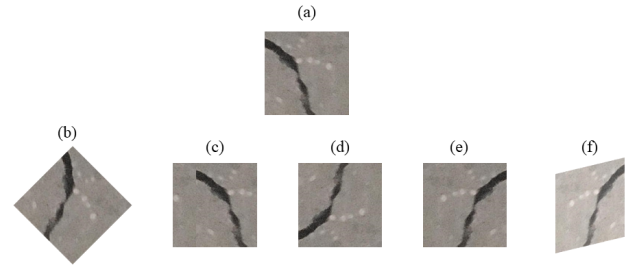


Fig. 6. Example of data augmentation

의 양을 증가시킨다. 예를 들어 이미지를 회전하거나, 상하나 좌우반전을 하거나, 뒤틀거나, 노이즈를 추가하는 것이 그 예시이다. 이를 통해 학습 데이터에 과적합되는 것을 막을 수 있을 뿐더러 기계학습 모델 자체의 성능을 증가시킬 수 있기에 널리 사용되는 기술이다.

Fig. 6은 원본 학습데이터에 데이터 증대 기술을 적용한 예시이다.

Fig. 6의 (a)는 원본 이미지, (b)는 45도 회전한 이미지, (c)는 오른쪽으로 이동한 이미지, (d)는 상하반전한 이미지, (e)는 좌우 반전을 한 이미지, (f)는 뒤틀기를 적용한 이미지이다. 이와 같이 하나의 원본 이미지로부터 데이터를 증대하여 많은 양의 데이터를 만들어 낼 수 있으며, 원본데이터에 약간의 수정을 가한 이미지기 때문에, 실제로도 기계학습모델의 성능 향상에도 중요한 역할을 한다.

3.2.2 기계학습 모델 학습

본 연구에서 기계학습 모델이 예측하는 값은 정해진 카테고리 내의 값이다(균열/정상). 따라서 본 문제는 분류 문제로 연속된 숫자를 예측하는 Regression 문제와 구분된다. 기계학습 모델은 예측된 값과 실제값의 차이가 최소화 되도록 기계학습 모델 내의 가중치와 편향이 학습된다. 본 연구에서는 앙상블 기법 중 Bagging을 적용하여 3개의 VGG-16 기반의 독립적인 모델을 학습하고 Voting을 통해 최종 값을 예측하는 기계학습 모델을 구성하였다.

3.2.3 검증

학습 데이터셋으로 학습된 기계학습 모델의 평가를 위해서는 학습에 사용되지 않은 검증 데이터셋이 활용된다. 학습 데이터셋에만 한정적으로 높은 성능을 보이는 과적합을 방지하기 위한 방법으로, 학습에 사용하지 않은 데이터를 예측한 후, 실제 결과값과의 비교를 통해 객관적인 성능 평가가 가능하다.

3.3 테스트 단계

테스트 단계에서는 실제 모델의 성능을 평가하는 것으로

Table 1. Performance metric

Actual class	Predicted class		
		Class=Crack	Class=Non-crack
	Class=Crack	True-positive (TP)	False Negative (FN)
Class=Non-crack	False-positive (FP)	True Negative (TN)	

검증과는 다른 데이터셋인 테스트 데이터셋을 사용한다. 검증에서 활용된 검증 데이터셋이 모델의 하이퍼 파라미터 튜닝에 활용되어 모델 성능에 간접적인 영향을 주었다면, 테스트셋은 학습과 완벽히 분리된 데이터셋으로, 순전히 모델의 성능 평가를 위해 활용된다. 하이퍼파라미터란 학습을 통해 도출되지 않고, 사용자가 직접 입력해주는 값으로 아키텍처의 구조와 학습 결과에 영향을 미친다. 하이퍼 파라미터의 예시로는 Batch 사이즈, 학습률, Dropout 비율 등이 있다. 모델의 성능 평가를 위한 성능평가지표로는 Accuracy, Precision Recall, F-1 점수 등이 널리 사용된다. Table 1은 널리 사용되는 성능 평가 테이블을 나타내며 다음과 같은 식은 테이블의 결과를 요약해서 나타내는 지표를 나타낸다. 특정 Class가 타 Class에 비해 크거나 작은 중요도를 가진다면 Precision 또는 Recall을 사용할 필요가 있다. 또한, 특정 예측값이 특정 Class로 편향되는 것을 방지하기 위해서는 F-1 점수가 사용되기도 한다. 본 연구에서는 특정 클래스를 예측하는 것에 대한 중요성이 편향되어있지 않기 때문에 Accuracy를 평가지표로 삼았다.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

4. 실험

챕터 3에서 제안된 프레임워크가 잘 작동하는지 확인하기 위해 오픈데이터셋에 프레임워크를 적용하였다. 기계학습이 충분한 성능을 발휘할 정도로 충분한 양의 데이터가 존재하며, 누구나 접근가능한 데이터를 통해 본 연구에서 제안한 프레임워크를 검증하고자 한다. 본 연구에서 사용된 컴퓨팅 환경의 Central Processing Unit(CPU)는 Ryzen 7 5800X, Graphics Processing Unit(GPU)로는 NVIDIA RTX 3070 Ti가 사용되었으며 32GB의 메모리를 활용하여 수행되었다. 구현은 Python 3.6.9로 Tensorflow-gpu 2.4.0를 이용하고 cuDNN 8.0, CUDA 11.0 환경에서 수행되었다.

4.1 데이터 설명

본 연구에서는 Mendeley Data 중 하나인 Concrete crack images for classification version 2(Çağlar Fırat, 2019)를 사

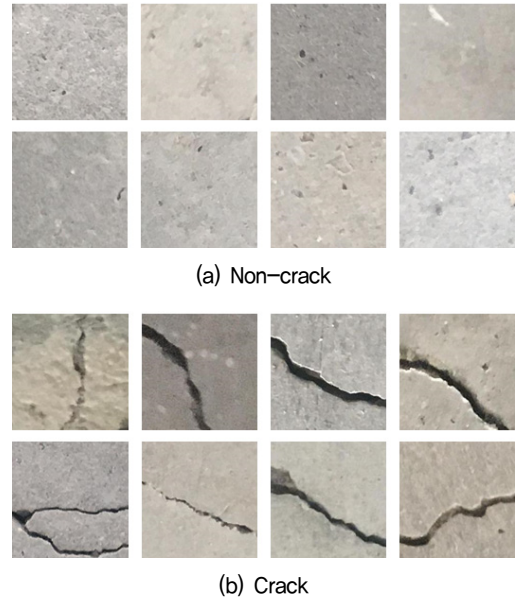


Fig. 7. Example of data used for experiments

용하였다. 해당 데이터 셋은 Middle East Technical University, Turkey(METU)의 콘크리트 빌딩에서 벽면과 바닥면을 촬영한 40,000개의 이미지들로 구성된다. 모든 이미지들은 표면으로부터 1m 떨어진 카메라로 촬영되었으며, 같은 날 비슷한 조명 환경에서 촬영되었다. 균열이 있는 20,000개의 이미지와 균열이 없는 20,000개의 이미지들로, 데이터의 원본은 4,032 X 3,024 픽셀의 해상도로 촬영되었고, Çağlar Fırat (2019)에 의해 227 X 227 픽셀로 추출된 이미지를 활용하였다. Fig. 7은 Case study에 활용된 사진들의 예시이다. Fig. 7(a)는 균열이 없는 정상 상태의 이미지 예시이고, Fig. 7(b)는 균열이 포함된 이미지의 예시이다. 표면 마감이나 조명 환경, 페인트나 이물질 존재 여부 등으로 완벽히 같은 조건은 아니며, 이는 학습된 균열 감지 모델이 일반화될 수 있는지 평가하기에 좋은 데이터셋이 될 수 있다.

4.2 준비 단계

준비된 데이터셋을 학습 데이터셋, 검증 데이터셋, 테스트 데이터셋으로 나누었다. 각각 학습데이터와 테스트데이터는 각 전체 데이터의 80%와 20%로 나누었고, 학습데

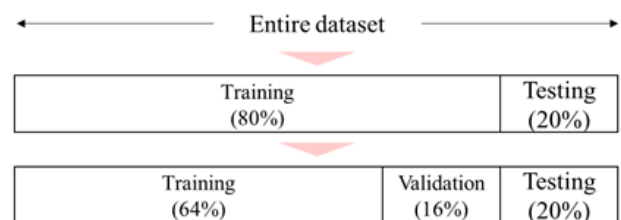


Fig. 8. Data split

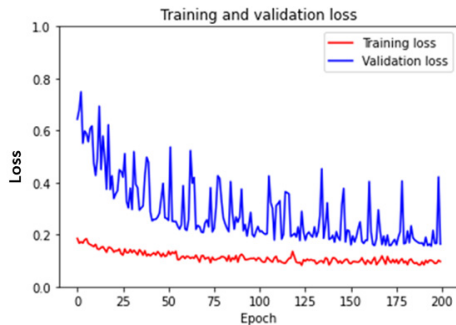
이터의 80%는 학습데이터로, 20%는 검증데이터로 다시 나누었다(Fig. 8). 각 데이터셋은 동일한 비율의 클래스(균열/정상)를 가지도록 지정되었으며, 이미지가 어떤 데이터셋에 할당될 지는 무작위로 선택되었다.

최종적으로 총 40,000개의 데이터 중, 25,600개의 데이터가 학습의 활용되었으며, 6,400개의 데이터는 검증에, 나머지 8,000개의 데이터는 테스트에 활용되었다.

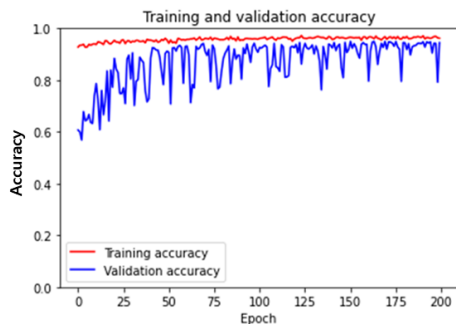
4.3 학습 단계

데이터는 섹션 3.2.1에서 서술했던 것과 마찬가지로 증대를 거쳐 수가 증가되었다. 회전, 뒤틀기, 확대, 상하좌우 반전, 이동 등이 적용되었다. Kears의 데이터 증대 함수를 활용하여 데이터 증대를 수행하였고, 빈 공간의 색은 가장 주변의 색으로 채워넣었다. 섹션 3.2.1의 예시에서 알 수 있듯이 데이터를 증대하더라도, 균열인지 아닌지에 대한 판단은 변함이 없다. 전체 데이터를 한 번 학습할 때마다 Loss와 성능(Accuracy)을 측정하였고, 그 결과는 Fig. 9와 같다. Fig. 9의 가로축은 Epoch로 전체 학습 데이터를 한 번 학습하면 1 Epoch이다. 또한 Loss는 Binary cross entropy로 아래와 같은 식을 통해 산출되었다.

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (2)$$



(a) Loss over epochs



(b) Accuracy over epochs

Fig. 9. Training result

N , y , $p(y)$ 각각 데이터 포인트 수와, 라벨, 해당 라벨의 예측된 확률을 나타낸다. 예측값과 실제값의 차이가 클수록 Loss가 커지며, 예측값과 실제값의 차이가 없다면 Loss값은 작아지게 된다.

학습을 진행할 때 마다 Loss는 감소하고 성능은 증가하는 것으로 보아 기계학습 모델이 균열 이미지와 그렇지 않은 이미지를 분류하기 위한 Feature를 높은 정확도로 추출하였고 이에 따라 성능이 향상됨을 알 수 있다. 학습 데이터셋과 검증 데이터셋 모두에서 같은 경향을 보이는 것으로 보아, 과적합 없이 일반적으로 높은 성능을 보일 것을 기대할 수 있다. Validation loss와 Accuracy에 편차가 발생하는 이유는 Batch를 사용하여 데이터가 무작위로 선택되는데 선택된 데이터가 편향되었을 때 일부 높은 Loss, 낮은 Accuracy를 보이는 것으로 추정된다.

4.4 테스트 단계

4.3에서 학습된 모델의 성능을 학습에 사용되지 않은 테스트 데이터셋을 통해 평가하였다. 테스트 데이터셋의 균열 데이터는 4,000개이며, 정상 데이터는 4,000개였다. 독립적으로 학습된 3개의 VGG-16모델이 테스트 데이터를 균열 또는 정상으로 분류한다. 최종 결과는 3개의 모델 중 다수의 결과를 선택하였다. 모델은 균열데이터 4,000개 중 3,867개를 균열로 분류하였으며, 정상 데이터 4,000개 중 3,783개를 정상으로 분류하였다(Fig. 10). 상대적으로 균열 데이터 경우에 더 높은 성능을 보였으나 큰 차이는 없었다. 결론적으로 전체 테스트 데이터에 대해 95.6%의 정확도를 보였다.

4.5 고찰

본 연구에서 기계학습 모델이 균열 이미지의 어떤 점을 부각시켜 분류를 했는지 분석하기 위해 기계학습 모델의 중간 과정을 시각화하였다. 결과는 Fig. 11과 같다. 앙상블을 이루는 기계학습 모델 중 하나의 모델에서 2번째, 9번째, 17번째 Layer를 시각화 한 것으로 입력된 데이터가 각 Layer

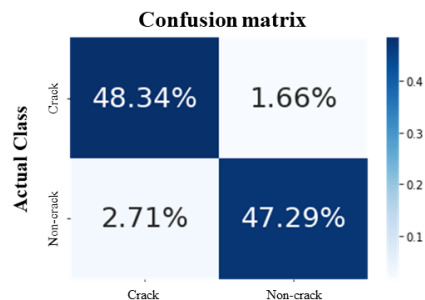


Fig. 10. Confusion matrix

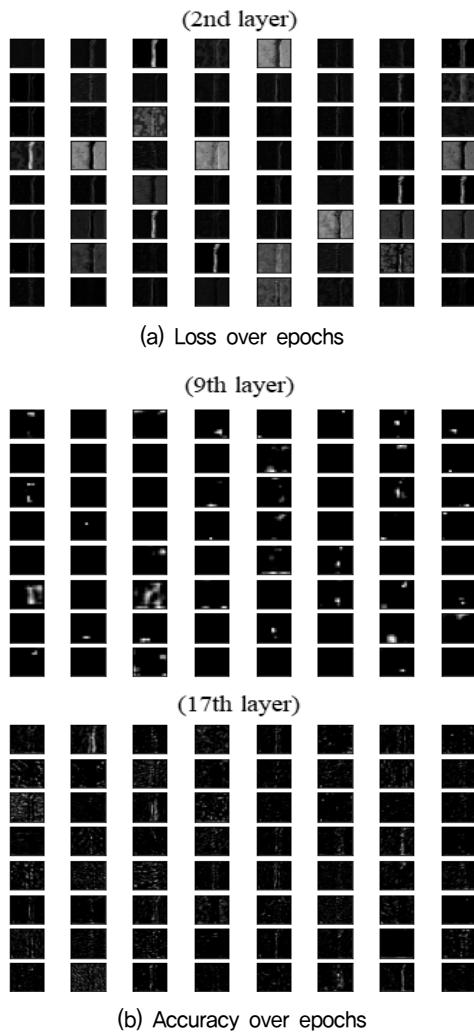


Fig. 11. Visualization of the Extracted Feature from layers

를 거치며 특정 Feature들이 균열이 발생한 부분을 강조하는 것을 알 수 있었다.

이를 통해 우리는 학습된 기계학습 모델이 균열과 균열이 아닌 부분의 차이를 Feature로 추출하여 분류에 활용하고 있음을 알 수 있다.

5. 결 론

본 논문에서는 자동화된 콘크리트 균열 감지를 위해 기계학습 기반의 균열 감지 프레임워크를 제안했다. 제안된 프레임워크를 통해 개발된 기계학습 모델은 높은 정확도로 시설물의 균열을 감지해 내었다. 특히, 기계학습 모델의 성능을 높이고자 앙상블 기법 중 하나인 Bagging을 적용하였고 이를 통해 95.6%의 정확도로 균열 감지가 가능했다. 앙상블 기법 외에도 성능을 향상시키기 위하여 Data 증대 기법을 적용하여 기계학습 모델이 학습되지 않은 데이터에서도 높은 성능

을 보이는 일반적인 모델이 될 수 있도록 하였다. 또한 해석이 불가능하여 Blackbox 모델로 불리는 기계학습 모델의 단점을 극복하고자 기계학습 모델이 어떤 Feature를 추출하는지 중간 Layer의 아웃풋을 시각화 함으로써 분석하였고, 균열과 주변 부분을 과장하여 대비되게 만들고, 이를 바탕으로 결정을 내리는 것을 확인할 수 있었다. 이와 같은 결과는 현재 전문가에 의존하여 빌딩의 상태를 평가하고, 관리함에 따라 발생하는 비용, 시간, 신뢰성 문제 등을 해결할 수 있을 것으로 기대된다. 향후 과제는 실험에 사용된 학습 모델을 다른 콘크리트 이미지들에 적용하여 본 기계학습 모델이 여러 데이터셋에 적용될 수 있음을 검증할 수 있을 것이다. 특히, 드론을 통해 수집된 데이터에 적용하여 검증할 예정이다.

References

1. Abdel-Qader, I., Abudayyeh, O. and Kelly, M. (2003), Analysis of edge-detection techniques for crack identification in bridges. *Journal of Computing in Civil Engineering*, Vol. 17, No. 4, pp. 255~263.
2. Chen, X., Chen, J., Han, X., Zhao, C., Zhang, D., Zhu, K. and Su, Y. (2020), A light-weighted cnn model for wafer structural defect detection, *IEEE Access*, No. 8, pp. 24006~24018.
3. Cheong, L. K., Suandi, S. A. and Rahman, S. (2019), Defects and components recognition in printed circuit boards using convolutional neural network, In *10th International Conference on Robotics, Vision, Signal Processing and Power Applications*, pp. 75~81.
4. Choudhary, G. K. and Dey, S. (2012), Crack detection in concrete surfaces using image processing, fuzzy logic, and neural networks, In *2012 IEEE fifth international conference on advanced computational intelligence (ICACI)*, pp. 404~411.
5. Chun, P. J., Izumi, S. and Yamane, T. (2021), Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 36, No. 1, pp. 61~72.
6. Çağlar Fırat, O. (2019), *Concrete Crack Images for Classification*, Mendeley Data.
7. Hutchinson, T. C. and Chen, Z. (2006), Improved image analysis for evaluating concrete damage, *Journal of Computing in Civil Engineering*, Vol. 20, No. 3, pp. 210~216.
8. Ji, B., Lee, S. J., Mazumder, M., Lee, M. S. and Kim, H. H. (2020), Deep Regression Prediction of Rheological Properties of SIS-Modified Asphalt Binders, *Materials*, Vol. 13, No. 24, pp. 5738.
9. Kawamura, K. and Miyamoto, A. (2003), Condition state evaluation of existing reinforced concrete bridges using neuro-fuzzy hybrid system, *Computers & structures*, Vol. 81, No. 18~19, pp. 1931~1940.
10. Kim, A. R., Kim, D. H., Byun, Y. S. and Lee, S. W. (2018), Crack detection of concrete structure using deep learning and image processing method in geotechnical engineering, *Journal of the Korean Geotechnical Society*, Vol. 34, No. 12, pp.

- 145~154 (In Korean).
11. Kim, Y. S. (2020), Current status of old buildings and future tasks, National Assmebly Research Service, pp. 01~13 (In Korean).
 12. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D. (1989), Backpropagation applied to handwritten zip code recognition, *Neural Computation*, Vol. 1, No. 4, pp. 541~551.
 13. Liu, Y., Yao, J., Lu, X., Xie, R. and Li, L. (2019), Deepcrack: A deep hierarchical feature learning architecture for crack segmentation, *Neurocomputing*, Vol. 338, pp. 139~153.
 14. Luxmoore, A. (1973), Holographic detection of cracks in concrete, *Non-Destructive Testing*, Vol. 6, No. 5, pp. 258~263.
 15. Prasanna, P., Dana, K. J., Gucunski, N., Basily, B. B., La, H. M., Lim, R. S. and Parvardeh, H. (2014), Automated crack detection on concrete bridges, *IEEE Transactions on automation science and engineering*, Vol. 13, No. 2, pp. 591~599.
 16. Ministry of Land, Infrastructure, Transport and Tourism. (2019), Tunnel periodic inspection guideline.
 17. Shi, Y., Cui, L., Qi, Z., Meng, F. and Chen, Z. (2016), Automatic road crack detection using random structured forests, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 12, pp. 3434~3445.
 18. Simonyan, K. and Zisserman, A. (2014), Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*.
 19. Xie, W., Wei, S., Zheng, Z., Jiang, Y. and Yang, D. (2021), Recognition of defective carrots based on deep learning and transfer learning, *Food and Bioprocess Technology*, Vol. 14, No. 7, pp. 1~14.
 20. Zhang, W., Zhang, Z., Qi, D. and Liu, Y. (2014), Automatic crack detection and classification method for subway tunnel safety monitoring, *Sensors*, Vol. 14, No. 10, pp. 19307~19328.
 21. Zhao, Z. and Chen, C. (2002), A fuzzy system for concrete bridge damage diagnosis, *Computers & structures*, Vol. 80, No.7~8, pp. 629~641.