

스몰 딥러닝을 이용한 아스팔트 도로 포장의 균열 탐지에 관한 연구

A Study on Crack Detection in Asphalt Road Pavement Using Small Deep Learning

지 봉 준[†]
Bongjun Ji

Received: August 11th, 2021; Revised: August 13th, 2021; Accepted: September 24th, 2021

ABSTRACT : Cracks in asphalt pavement occur due to changes in weather or impact from vehicles, and if cracks are left unattended, the life of the pavement may be shortened, and various accidents may occur. Therefore, studies have been conducted to detect cracks through images in order to quickly detect cracks in the asphalt pavement automatically and perform maintenance activity. Recent studies adopt machine-learning models for detecting cracks in asphalt road pavement using a Convolutional Neural Network. However, their practical use is limited because they require high-performance computing power. Therefore, this paper proposes a framework for detecting cracks in asphalt road pavement by applying a small deep learning model applicable to mobile devices. The small deep learning model proposed through the case study was compared with general deep learning models, and although it was a model with relatively few parameters, it showed similar performance to general deep learning models. The developed model is expected to be embedded and used in mobile devices or IoT for crack detection in asphalt pavement.

Keywords : Deep learning, Crack detection, Asphalt, Road maintenance

요 지 : 아스팔트 포장의 균열은 날씨의 변화나 차량에 의한 충격으로 발생하며, 균열을 방치할 경우 포장 수명이 단축되고 각종 사고를 불러 일으킬 수 있다. 따라서 아스팔트 도로 포장의 균열을 빠르게 감지하여 보수조치를 취하기 위하여 이미지를 통해 균열을 자동으로 탐지하기 위한 연구들이 지속되어 왔다. 특히 최근들어 Convolutional Neural Network를 사용하여 아스팔트 도로 포장의 균열을 탐지하려는 모델들이 많이 연구되고 있으나, 고성능의 컴퓨팅 파워를 요구하기 때문에 실제 활용에는 한계가 있다. 이에 본 논문에서는 모바일 기기에 적용 가능한 스몰 딥러닝 모델을 적용하여 아스팔트 도로 포장의 균열을 탐지하는 모델의 개발을 위한 프레임워크를 제안한다. 사례연구를 통해 제안한 스몰 딥러닝 모델은 일반적인 딥러닝 모델들과 비교 연구되었으며, 상대적으로 적은 파라미터를 가지는 모델임에도 일반적인 딥러닝 모델들과 유사한 성능을 보였다. 개발된 모델은 모바일 기기나 IoT에 임베디드 되어 사용될 수 있을 것으로 기대된다.

주요어 : 경량 딥러닝, 균열 탐지, 아스팔트, 도로 관리

1. 서 론

아스팔트로 포장된 도로는 지형이나 기후, 교통 조건 등을 고려하여 설계, 포장되지만 기후변화와 같은 자연적 요인이나, 이동 차량의 하중 등으로 노화되고 균열이 발생한다(Weng et al., 2019). 균열이 발생한 시점에 적절한 대처 없이 방치된 아스팔트 포장은 우수의 침투 등에 의해 파손이 가속화되어 더욱 확대될 수 있어 적절한 시점에 유지보수를 하는 것이 중요하다. 이에 따라 포장 상태를 진단하는 고가의 특수 장비를 투입하거나 전문가가 도로를 육안 점검하여 사전 정의된 기준에 의해 균열을 평가하는 방법이 주로 쓰인다(Kim et al., 2008). 하지만 육안 점검은 많은

시간과 비용이 발생되고 사람에 의존하기 때문에 사람의 역량이나 컨디션에 따라 신뢰성에 문제가 발생할 수 있다는 단점이 있다(Koch & Brilakis, 2011; Hoang et al., 2018). 도로라는 특성 상 검사자는 운행 중인 자동차에 노출될 수 있어 검사의 과정 또한 위험할 수 있으므로 육안검사를 자동화하려는 연구들이 수행되어왔다. 기존의 영상 처리를 통한 표면균열 검사는 특정 결함을 규격화하고 Feature로 추출하여, 해당 Feature를 포함하고 있는지와 같은 규칙 기반의 검사들이 주를 이뤘다. 하지만 이와 같은 규칙 기반의 검사들은 다양한 변형이 존재할 수 밖에 없는 현실에서 한정적으로만 쓰일 수 밖에 없었다. 특히 실험환경이 아닌, 외부에 노출되어있는 도로에서 수집된 영상은 기온, 기상

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

상태, 조도 등 기후에 영향을 받기 때문에 이에 영향을 받지 않는 강건한 방법이 필요했다. 이러한 한계를 극복하기 위해 CNN(Convolutional Neural Network)과 같은 딥러닝 기법을 적용하기 시작했다(Dorafshan et al., 2018; Cha et al., 2017; Choi et al., 2019; Kumar et al., 2020, Chun et al., 2021). CNN은 이미지를 처리하는데 우수한 성능을 보인 딥러닝의 한 종류로 이미지의 특성을 자동으로 추출하고, 추출된 특성들로부터 예측해야하는 값을 산출하는데 높은 성능을 보였다. 하지만 아스팔트의 균열 감지는 상대적으로 낮은 정확도를 보였으며, 대부분의 연구가 초기 CNN 구조를 답습하는 것과 적은 데이터의 한계를 극복하지 못하였다. 또한 일부 연구에선 보다 많은 양의 파라미터를 필요로 하는 딥러닝 아키텍처들을 적용하였지만, 이들 연구는 실제 현장에서 활용 가능한 수준을 넘어선다. 따라서 학습하기도, 활용하기도 쉽지 않았다. 예를 들어 ResNet-50의 파라미터 수는 2,300만 개 이상이고, VGG-16은 1억 3,800만개 이상의 파라미터의 학습을 필요로 한다. 이에 따라 실험실 수준의 연구를 벗어나 상업화에는 한계가 있었다. 이러한 한계를 극복하기 위해 본 연구에서는 스몰 딥러닝이라 불리는 MobileNet(Howard et al., 2017)을 적용하고자 한다. MobileNet은 성능상으로는 최선(State-of-the-art)인 딥러닝 기법이 아니지만 최신인 딥러닝 기법과 유사한 성능을 보여주며, 타 딥러닝 아키텍처에 비해 적은 수의 파라미터 학습을 요구한다. 이에 따라 모바일 기기에서도 작동 가능한 장점이 있다. 따라서 MobileNet 아키텍처를 적용하여 높은 정확도의 도로 포장 균열 감지가 가능하다면 아스팔트 도로의 균열 감지를 위한 학습이 모바일 단말기에서 이루어질 수 있으며, 균열 감지 또한 실시간으로 이루어질 수 있다. 이를 통해 각 상황에 맞는 아스팔트 균열 감지 모델 개발이 가능할 것이며, 감지된 결과를 바탕으로 전문가가 육안으로 정밀 검진할 수 있는 기회를 제공해줄 것으로 기대된다. 특히 현재 사용되고 있는 고가의 레이저 장비나 센서 등을 대체하여 모바일 기기로도 균열을 감지할 수 있다는 점은 본 연구의 현장 적용 및 확산에 이점으로 작용할 것이다. 본 논문은 다음과 같이 구성되어 있다. 두 번째 챕터는 이론적 배경에 대한 설명으로 CNN과 MobileNet 아키텍처에 관해 기술하고자 한다. 세 번째 챕터에서는 프레임워크에 대해 제안하고, 네 번째 챕터에서는 실험을 통해 제안된 프레임워크에 대해 검증하고자 한다. 마지막으로 결론 섹션에서는 연구 결과를 요약하고, 앞으로의 연구 방향을 제시한다.

2. 이론적 배경

이 챕터에서는 이미지 딥러닝의 기반이 되는 CNN과 스몰 딥러닝 모델인 MobileNet에 대해 논의하고자 한다.

2.1 CNN

CNN은 이미지 내의 텍스트를 분류하기 위해 개발되었으며 여러 분류 문제에 널리 사용되고 있다. 이미지 인식 분야에서 기존 Neural Network 대비 매우 좋은 성능을 보였는데, 이는 Convolutional layer, Pooling layer, Fully-connected layer 라고 불리는 세 가지의 레이어로 구성되는 CNN의 특징 덕분이다. Convolutional layer에서는 convolution 연산이 이루어지는데 이는 이미지에 Filter 값을 곱한 후 합을 구하는 연산이다. Convolution 연산은 필터를 이동해가며 수행되며 이를 통해 이미지 내 Feature를 추출한다(Fig. 1).

Pooling layer는 Convolutional layer를 통해 추출된 Feature를 다시 한 번 필터링 하는 Layer로 대푯값을 선정하는 연산을 수행한다. 최대값을 선정하는 경우를 Max pooling이라 부르며, 평균값을 선정하는 경우 Average pooling이라고 불린다(Fig. 2).

LeCun et al.(1989)에 의해 제안된 CNN은 32×32 픽셀의 Grayscale 손글씨를 인식하기 위해 개발되었으며, 높은 성능을 보였다. 이미지는 Convolutional layer와 Pooling layer를 차례로 2번 거치고, Fully connected layer를 3번 거친 후, 손글씨를 예측하는 구조를 가진다(Fig. 3). 이후 AlexNet

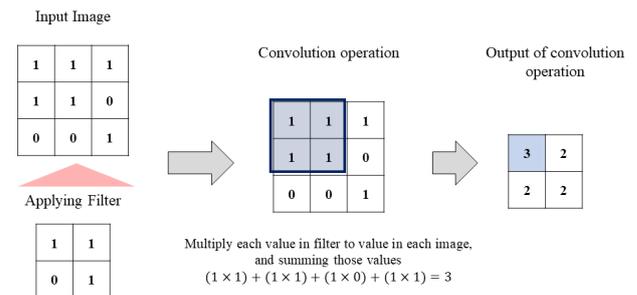


Fig. 1. Convolution operation in Convolution layer

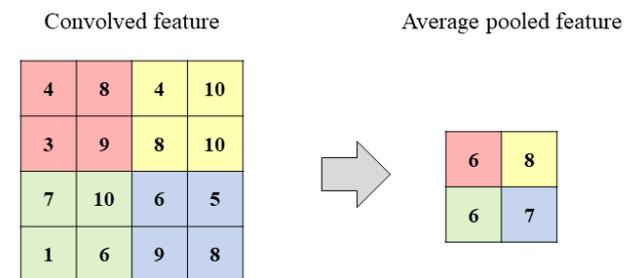


Fig. 2. Average pooling operation

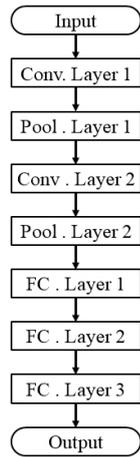


Fig. 3. Architecture of CNN

(Krizhevsky et al., 2012), VGG-16(Simonyan & Zisserman, 2014), ResNet(He et al., 2016) 등과 같은 많은 CNN 기반의 이미지 인식 딥러닝 모델들이 개발·활용되었다. Convolutional layer에서는 모서리와 같은 고수준 특징을 추출하는 역할을 하며, Pooling layer는 네트워크에서 계산의 양을 줄이기 위해 사용된다. Fully Connected layer는 다차원의 출력을 평면화(Flatten)하고 Output에 연결하며, Convolutional layer와 Pooling layer를 거쳐 추출된 Feature와 Output간의 비선형적 관계를 학습한다.

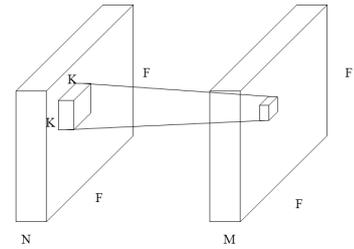
2.2 MobileNet

MobileNet 또한 많은 CNN의 파생 아키텍처 중 하나다. MobileNet은 모바일 기기에서 작동 가능한 딥러닝 모델을 목표로 개발되어 모바일 기기나 임베디드 시스템 또는 GPU가 없는 컴퓨터에서도 효율적으로 작동하도록 설계되었다. 기존 CNN 아키텍처들과 비교했을 때, 가장 큰 차이점은 MobileNet의 Depthwise separable convolution이다. MobileNet은 Depthwise separable convolution을 통해 학습에 필요한 파라미터의 수를 줄이고, 성능을 최대화할 수 있었다.

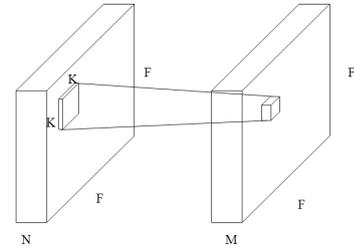
Depthwise separable convolution은 Depthwise convolution과 Pointwise convolution을 연속적으로 연결한 형태로 기존 Convolution 연산(Fig. 4(a))을 공간 방향의 Depthwise convolution(Fig. 4(b))과 채널 방향의 Pointwise convolution (Fig. 4(c))로 대체한다. Depthwise convolution은 채널마다 다른 필터를 적용하고, Pointwise convolution 시에는 반대로 채널방향으로만 연산을 수행한다.

필터의 크기를 K, 이미지의 높이, 너비를 F, 이미지의 입력 채널의 수를 N, 이미지의 출력 채널의 수를 M이라고 한다면 일반적인 Convolution 연산은 $F^2 \times K^2 \times N \times M$ 의 연산량을 가진다.

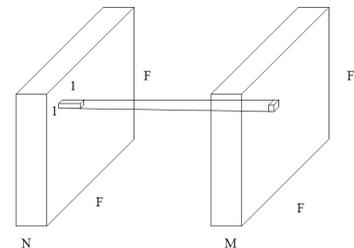
반면, Depthwise separable convolution의 연산량은 ($F^2 \times$



(a) Traditional convolution operation



(b) Depthwise convolution



(c) Pointwise convolution

Fig. 4. Convolution operation

$K^2 \times N) + (F^2 \times 1^2 \times N \times M)$ 이다. 만약 필터 사이즈 K가 2, 입력 채널의 수 N이 64인 경우 연산량이 기존 Convolution 연산 대비 1/18이 되는 것이다. 이를 통해 적은 수의 파라미터로 Convolution 연산을 대체하였다.

3. 프레임워크

본 연구가 제안하고자 하는 프레임워크는 다음과 같이 크게 두 가지의 단계로 구성된다. 첫 번째 단계는 딥러닝 모델을 학습하는 Offline 학습 단계, 두 번째 단계는 학습된 모델을 사용하는 Online 단계이다(Fig. 5).

3.1 Offline 학습 단계

Offline 학습 단계에서는 수집된 아스팔트 이미지를 이용하여 스몰 딥러닝 모델을 학습한다. 이를 위한 첫 번째 단계로 아스팔트 이미지에 라벨을 할당한다. 즉, 수집된 아스팔트 이미지가 균열이 있는 이미지인지 아닌지 판별한다. 두 번째 단계로는 데이터를 학습 데이터와 검증 데이터로 나눈다. 학습 데이터와 검증 데이터 모두 학습에 활용되지만 그

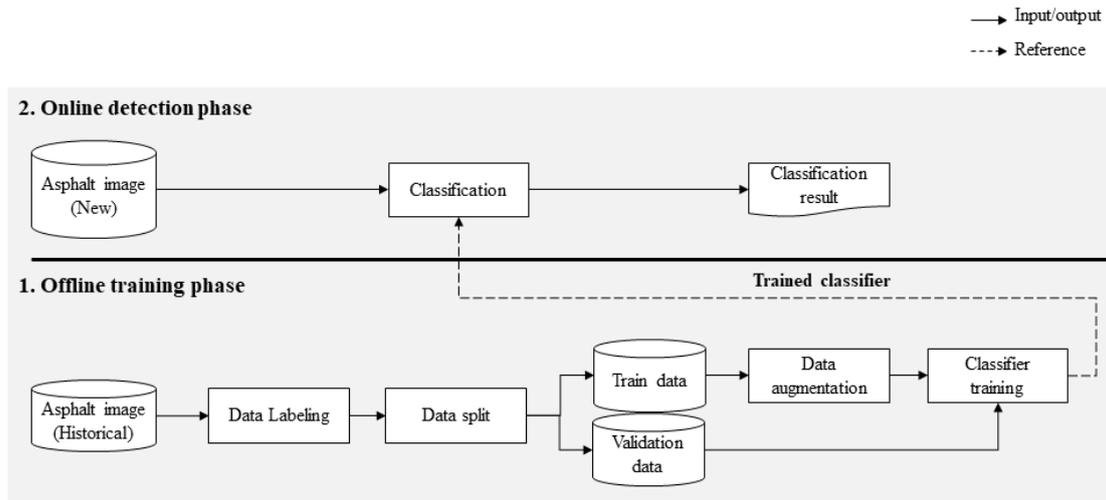


Fig. 5. Proposed framework for asphalt crack detection model development

목적이 다르다. 딥러닝 모델의 학습이란 예측값과 실제값의 차이가 작아지도록 딥러닝 모델을 구성하는 가중치(Weight)와 편향(Bias)를 찾아내는 것이다. 이를 위해 학습 데이터는 직접적인 학습에 활용된다. 랜덤으로 주어진 초기 가중치와 편향은 학습데이터를 통해 예측한 값(균열 또는 균열 아님)과 실제 값(균열 또는 균열 아님)과 실제값의 차이를 계산하고 이 차이가 줄어들도록 Weight와 편향이 조정된다. 검증 데이터는 학습 데이터로 학습된 모델이 학습데이터에 편향되거나 과적합(Overfitting)되는지 여부를 판단하기 위해 사용된다. 과적합은 일반화된 모델이 아닌 주어진 학습데이터에만 정확한 결과를 산출하는 것을 말한다. 학습 데이터로 학습된 스몰 딥러닝 모델이 과적합되지 않은지 판단하기 위해 검증 데이터가 사용된다. 학습 데이터에서는 예측값과 실제값의 차이가 적어지더라도, 검증 데이터에서는 예측값과 실제값의 차이가 커진다면 과적합을 의심할 수 있으므로 더 이상의 학습을 멈추는 것이다.

학습 데이터는 학습 전에 데이터 Augmentation이라는 단계를 거친다. 데이터 Augmentation이란 학습 데이터에 조금의 변형을 가해 학습 데이터의 양과 다양성을 증가시킨다. 예를 들어 학습 데이터를 정해진 범위 내에서 회전하거나, 옆으로 이동시키거나, 상하 또는 좌우 반전을 통해 학습 데이터와는 다르지만, 학습데이터의 특징을 유지한 데이터를 만드는 것이다. 이와 같은 데이터 Augmentation을 통해 학습 데이터의 양과 다양성을 증가시키며 이를 통해 학습 성능을 향상시킨다.

마지막으로 스몰 딥러닝 모델을 학습한다. 스몰 딥러닝 모델은 MobileNet의 아키텍처를 수정하여 결과값으로 두 가지의 값을 산출하도록 구성하였다. 스몰 딥러닝 모델은 이미지를 Red, Green, Blue의 3가지 채널의 값으로 입력받아, 해당 값으로부터 실제값, 균열이면 1, 균열이 아니라면

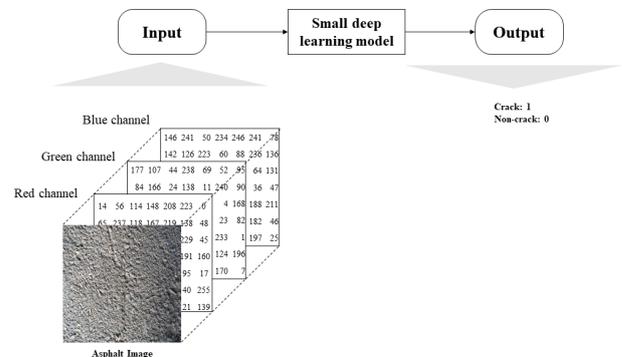


Fig. 6. Concept of classifier training

0을 예측하도록 학습한다(Fig. 6). 모든 색은 Red, Green, Blue의 삼원색으로 표현 가능하므로 이 세 가지 채널의 값을 통해 색을 표현 가능하다.

Loss는 잘못된 예측에 대한 패널티를 뜻한다. 따라서 학습은 Loss를 줄이는 방향으로 학습되며 Loss는 다음과 같이 정의된다.

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

N은 데이터 포인트 수를, y는 라벨을, p(y)는 해당 라벨의 예측된 확률을 나타내며 예측된 값과 실제값의 차이가 클수록 Loss가 크며, 반대의 경우 Loss가 작음을 알 수 있다.

학습을 위해 하이퍼 파라미터(Hyper parameter)를 설정해야 하는데, 하이퍼 파라미터란 데이터로부터 추론된 값이 아닌, 사용자가 입력하는 값을 뜻한다. 대표적인 하이퍼파라미터로 Dropout 비율, Batch 사이즈 등이 그 예시이다. Dropout은 과적합을 방지하기 위한 Regularization 방법 중

하나로, 무작위로 노드들을 제거하는 테크닉이다. 또한 Batch 사이즈는 학습을 위해 데이터 중 몇 개의 데이터를 학습에 활용할지 선택하는 것을 말한다.

3.2 Online 감지 단계

Online 감지 단계에서는 학습에는 활용되지 않은 아스팔트 이미지를 학습된 스몰 딥러닝 모델에 입력하여 균열인지 여부를 판별한다. 본 단계에서는 신규 데이터들이 사용되는데 학습된 모델을 활용하여 새로운 이미지가 입력되었을 때 균열 여부를 판별하는 점에서 실제 활용 시나리오를 가정한다. 감지 단계의 성능 평가는 Accuracy 지표를 사용한다. Table 1과 같이 예측된 값과 실제 값에 따라 결과를 네 가지로 분류 가능한데, 전체 예측 개수 중, 정확히 예측한 개수의 비율을 Accuracy라고 한다.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

Table 1. Performance metric

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

4. 실험

본 챗터에서는 챗터 3에서 제안된 프레임워크의 검증을 위하여 실제 아스팔트 이미지를 활용하여 스몰 딥러닝 모델을 학습, 성능을 평가하였다. 실험을 위한 데이터에 대한 설명은 챗터 4.1에서, Offline 학습 단계와 Online 예측 단계에 대한 서술은 각기 챗터 4.2와 챗터 4.3에서 다룬다. 본 실험에 활용된 컴퓨팅 환경은 다음과 같다. 하드웨어 환경의 경우, Central Processing Unit(CPU)는 Ryzen 7 5800X, Graphics Processing Unit(GPU)로는 NVIDIA RTX 3070 Ti가 사용되었으며 32GB의 램(RAM)에서 실험이 이루어졌다. 소프트웨어 환경으로는 Python 3.6.9, Tensorflow-gpu 2.4.0, cuDNN 8.0, CUDA 11.0에서 모든 실험이 이루어졌다.

4.1 데이터 Description

본 연구에서 사용한 아스팔트 이미지는 총 400개로 구성 되어있다(Balaji et al., 2019). 이 중 50%인 200개의 이미지는 균열이 포함된 아스팔트 이미지이며, 나머지 200개의 이미지는 균열이 포함되지 않은 이미지이다. Fig. 7은 본 실험

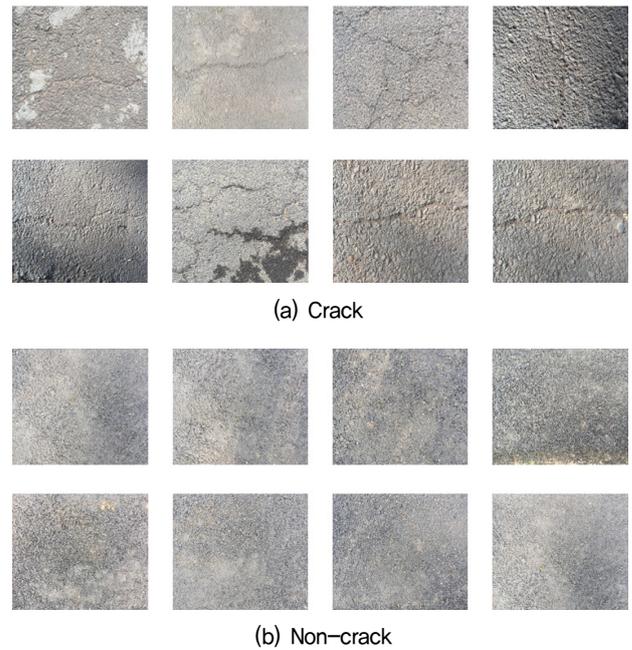


Fig. 7. Example of data used for experiments

에서 사용한 아스팔트 이미지의 예시이다. Fig. 7(a)는 균열이 포함된 아스팔트 도로의 이미지 예시이며, Fig. 7(b)는 균열이 포함되지 않은 정상상태의 아스팔트 도로 이미지이다. 균열 데이터와 정상 상태의 아스팔트 도로 이미지 모두 448×448픽셀로 이루어져 있다.

전체 데이터의 80%인 320개의 이미지는 Historical 이미지로 가정하여 Offline 학습에 사용하고, 나머지 20%인 80개의 이미지는 새로운 이미지로 가정하여 Online 감지 단계에서 활용하였다. 두 데이터는 동일한 비율의 정상/균열 데이터를 가지도록 무작위로 선정하였다.

4.2 Offline 학습 단계

준비된 데이터셋 중 80%를 학습 데이터(256개)로, 나머지 20%(64개)를 검증 데이터로 나누었다. 학습 데이터는 직접적인 모델의 학습(기준치와 편향 값 산출)에 활용되며, 검

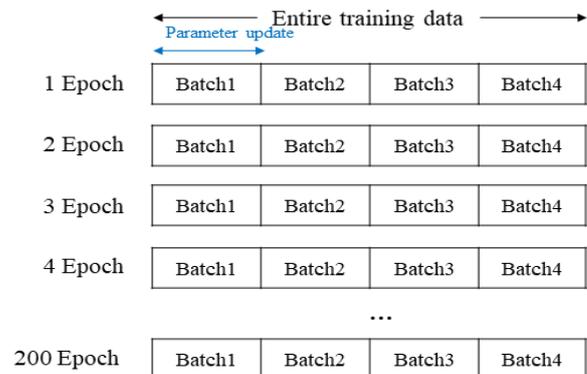
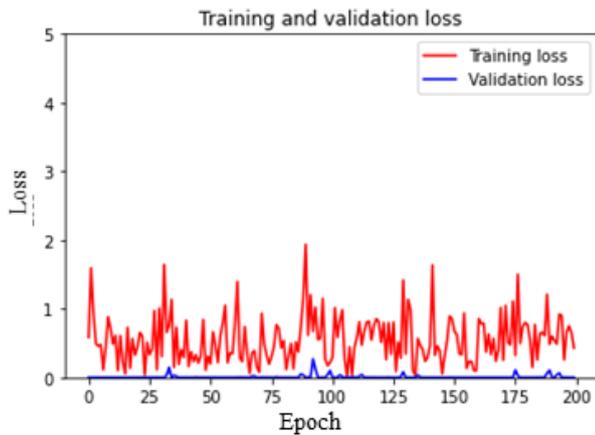


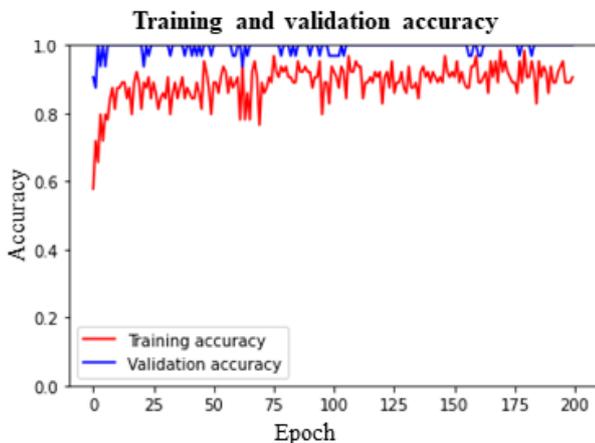
Fig. 8. Batch and epoch in training

증 데이터는 학습된 모델의 성능을 중간 평가하고, 더 학습을 진행할지, 멈출 지에 대한 판단을 하거나 혹은 Dropout이나 Batch size 선정과 같은 하이퍼 파라미터 선정에 활용하였다. 학습은 Fig. 8과 같이 진행하였다. 전체 학습 데이터를 4개의 Batch로 나눈 후, 각 Batch 마다 학습을 진행, 가중치와 편향을 업데이트 한 후, 다음 Batch로 학습을 진행하고 총 4번의 Batch 학습을 통해 전체 데이터를 스몰 딥러닝 모델이 학습하도록 구성하였다(1 Epoch). 그리고 전체 데이터는 200번 학습하도록 구성하여, 총 200 Epoch, 800 Batch의 학습이 이루어졌다.

이상적인 학습이라면 각 Epoch마다 Loss는 감소하고, Accuracy는 증가하는 모습을 보여야 한다. 실제 본 연구의 실험에서는 Epoch가 증가함에 따라 학습 데이터와 검증 데이터 모두에서 Accuracy가 증가하는 모습을 보였다. 다만, Loss의 경우, Epoch가 증가함에도 일정 수준 이하로 떨어지지 않음을 알 수 있었는데, 이는 Loss의 절대적인 값이 작기 때문이다(Fig. 9). Training loss보다 Validation loss가 작은 원인은 다양할 수 있다. Training loss의 측정시점이



(a) Loss over epochs



(b) Accuracy over epochs

Fig. 9. Training result

Validation loss보다 앞선 시점인 것이 원인일 수도 있지만, 본 연구에서는 Validation 데이터의 수가 적고, Validation 데이터에 분류가 쉬운 데이터가 할당되어 예측값이 정확한 것으로 추정된다.

4.3 Online 감지 단계

4.3에서 학습된 모델의 성능을 학습에 사용되지 않은 테스트 데이터셋을 통해 평가하였다. 테스트 데이터셋의 균열 데이터는 40개이며, 정상 데이터는 40개였다. 학습된 스몰 딥러닝 모델은 균열데이터 40개 중 40개를 균열로 분류하였으며, 정상 데이터 40개 중 37개를 정상으로 분류하였다(Fig. 9). 균열 데이터의 경우 정확히 균열로 예측하였고 결론적으로 0.93의 Accuracy를 보였다. 이는 실제 아스팔트 도로의 균열 판정을 위해 사용되어도 손색없을 만큼 충분한 성능이다. 특히, 균열은 100%의 Accuracy로 감지해 냈기에 균열임에도 놓치는 경우는 드물 것으로 예상된다.

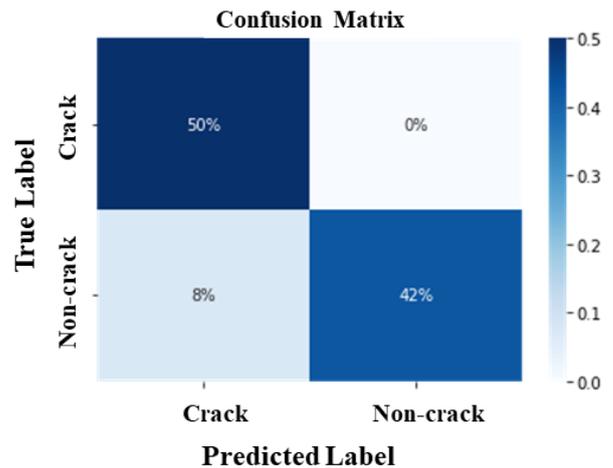


Fig. 10. Confusion matrix

4.4 고찰

본 연구는 스몰 딥러닝 모델인 MobileNet 아키텍처를 도입하여 아스팔트의 균열을 감지하고자 했다. 그 결과, 균열 이미지와 그렇지 않은 이미지의 차이를 높은 성능으로 추출하고 균열을 감지할 수 있음을 알 수 있다. 제한한 연구의 상대적인 성능 평가를 위해 VGG-16과 ResNet-50 아키텍처를 같은 프레임워크를 활용하여 학습·평가하였다. VGG-16은 Test accuracy가 0.95, ResNet은 Test accuracy가 1.00을 나타냈다. 즉, 파라미터가 많고 복잡한 아키텍처일수록 학습과 예측 시간은 오래 소요되지만, 높은 성능을 보임을 알 수 있었다(Fig. 11). 하지만, MobileNet은 VGG-16대비 약 1/3의 파라미터, ResNet에 비해서는 약 1/5의 파라미터로

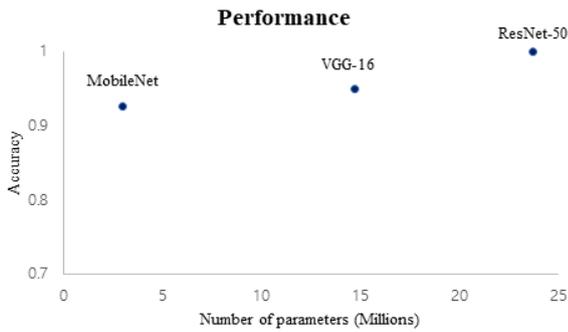


Fig. 11. Performance comparison

유사한 성능을 나타내므로 충분히 효과적으로 아스팔트의 균열을 감지해낼 수 있음을 알 수 있다.

5. 결 론

본 논문에서는 모바일 기기에서 작동가능한 스펴 딥러닝 아키텍처인 MobileNet을 활용한 아스팔트 균열 감지 모델을 개발하고 성능을 평가하였다. VGG-16, ResNet-50 등 타 딥러닝 모델이 각각 0.95, 1.00의 Accuracy를 보였고 MobileNet 기반의 아스팔트 균열 감지모델의 성능은 0.93으로 나타났다. 파라미터의 수가 월등히 적음에도 불구하고 감지 성능에 큰 차이가 없었다. 특히 균열인 경우는 모두 균열로 판단했기 때문에 이와 같은 결과는 현재 전문가에 의존하거나 고가의 특수 장비에 의존하여 아스팔트 도로의 포장 상태를 평가함에 따라 발생하는 비용과 시간, 신뢰성 문제 등을 해결할 수 있을 것으로 기대된다. 실제 균열 감지 시스템의 구현 및 적용을 위해서는 한정된 자원을 활용할 수 밖에 없다. 하지만 스마트폰과 같은 모바일 기기는 카메라와 AP (Application Processor)를 탑재하고 있다. 이러한 모바일 기기의 보급이 보편화된 상황에서 모바일 기기 내에서 학습과 예측이 가능한 모델을 실험, 검증했다는 점에서 본 논문은 의의를 가진다. 본 논문에서 제안된 프레임워크를 통해 개발된 아스팔트 균열 감지 모델은 널리 적용, 사용될 수 있을 것으로 기대된다.

1. Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S. and Büyüköztürk, O. (2017), Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, *Computer-Aided Civil and Infrastructure Engineering*, Vol. 33, No.9, pp. 731~747.
2. Choi, S. H., Do, M. S., You, S. H. and Cho, C. S. (2018), Determination of visual based asphalt pavement crack condition using deep learning, *International Journal of Highway Engineering*, Vol. 20, No. 5, pp. 75~83 (In Korean).
3. Chun, P. J., Yamane, T. and Tsuzuki, Y. (2021), Automatic detection of cracks in asphalt pavement using deep learning to overcome weaknesses in Images and GIS Visualization, *Applied Sciences*, Vol. 11, No. 3, pp. 892.
4. Dorafshan, S., Thomas, R. J. and Maguire, M. (2018), Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete, *Construction and Building Materials*, Vol. 186, pp. 1031~1045.
5. He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770~778.
6. Balaji, A. J., Balaji, G. T., Dinesh, M. S., Binoy, N. and Ram, D. S. (2019), Asphalt crack dataset, *Mendeley Data*, V2.
7. Kim, J. H., Kim, J. R. and Moon, H. C. (2008), Development of pavement distress survey system, *Journal of Korean Society of Road Engineers*, pp. 475~484 (In Korean).
8. Koch, C. and Brilakis, I. (2011), Pothole detection in asphalt pavement images, *Advanced Engineering Informatics*, Vol. 25, No. 3, pp. 507~515.
9. Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 25, 1097~1105.
10. Kumar, A., Kumar, A., Jha, A. K. and Trivedi, A. (2020), Crack detection of structures using deep learning framework, In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* pp. 526~533.
11. 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.
12. Nhat-Duc, Hoang, Quoc-Lam Nguyen and Van-Duc Tran. (2018), Automatic recognition of asphalt pavement cracks using meta-heuristic optimized edge detection algorithms and convolution neural network, *Automation in Construction* Vol. 94, pp. 203~213.
13. Simonyan, K. and Zisserman, A. (2014), Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*.
14. Weng, X., Huang, Y. and Wang, W. (2019), Segment-based pavement crack quantification, *Automation in Construction*, Vol. 105, pp. 102819.