

# 딥러닝 기반 병충해 인식 기술

Deep Learning based Plant Disease Recognition

김가현 세종대학교 지능기전공학과

**Abstract** : Prompt response to plant disease is important to minimize the damage and prevent the spread of disease to other farms. It is possible to early detect plant disease by developing a system based on deep learning, which has recently shown rapid development. In order to develop these systems, various and large amounts of data are required. However, it is difficult in reality to build or obtain such a dataset. Therefore, there are attempts to overcome the problems by utilizing data augmentation methods. In this report, deep learning-based plant disease recognition systems are analyzed.

## I. 머리말

병해충은 작물에 직접적인 영향을 주고 이는 농업 부문의 연간 수확량과 연결되어 경제적 손실을 일으킬 수 있을 뿐 아니라 심하게는 국가의 식량 안보에도 영향을 줄 수 있다[2]. 따라서 농가에서 재배 작물의 병해충을 조기에 탐지할 수 있다면 신속한 대응을 통해 그 피해를 최소화할 수 있게 된다.

기존 컴퓨터 비전 기반 영상 분류 기술은 영상에서 특징을 추출하고 이를 기반으로 특징을 분류하기 때문에 정확도가 떨어지는 단점이 있었다 [1]. 하지만 최근 다양한 분야에서 문제 해결을 위해 적용되는 CNN(Convolutional Neural Network) 등의 딥러닝 기술은 영상에서 물체 혹은 이벤트를 감지 및 인식하기 위해 사용되는데 그 정확도가 높다. 이를 농업에 접목하여 작물의 정상적인 작물과 병든 작물을 분류하고, 더 나아가 해충 혹은 질병을 검출할 수 있는 기술들이 제안되고 있다.

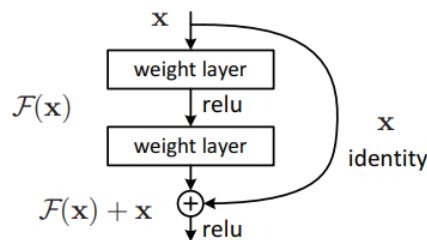
본 보고서에서 참고한 기술들은 물체 검출에서 널리 사용되던 백본 네트워크(backbone network)에 다양한 데이터 증강(data augmentation) 방법을 적용한다는 공통점이 있다. 2장에서는 병충해 인식에 적용된 백본 네트워크의 간략한 소개, 3장에서는 데이터 증강 방법, 4장과 5장에서는 참고 기술의 시스템 설명과 성능에 대해 각각 기술한다.

## II. 백본 네트워크 소개

본 절에서는 병해충 인식 시스템에서 사용된 백본 네트워크에 대해 간략히 소개한다.

### 1. ResNet

Kaming He, et al.에 의해 제안된 ResNet[3]은 인공 신경망의 레이어가 많아지면서 발생하는

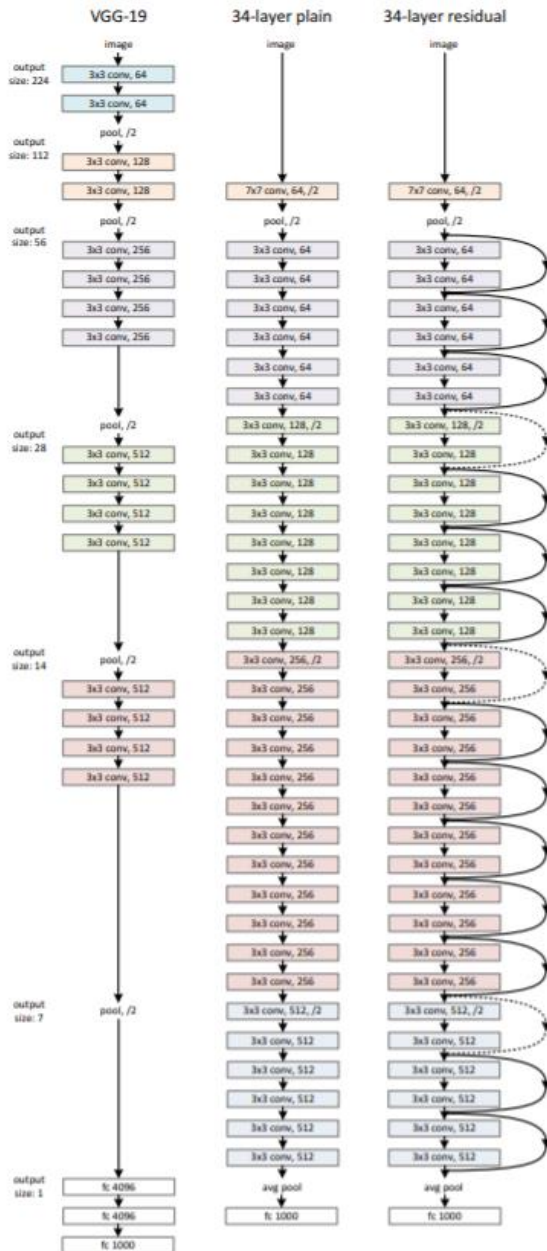


(그림 1) 잔차 학습 [3]

그래디언트(gradient) 소실/폭발 문제를 해결하기 위해 short connection(또는 skip connection)을 추가한 신경망 구조이다. Short connection은 층(layer)을 연속적으로만 연결하는 것이 아니라 중간을 건너뛰으로써 그래디언트를 직접적으로 잘 전달한다는 장점이 있다.

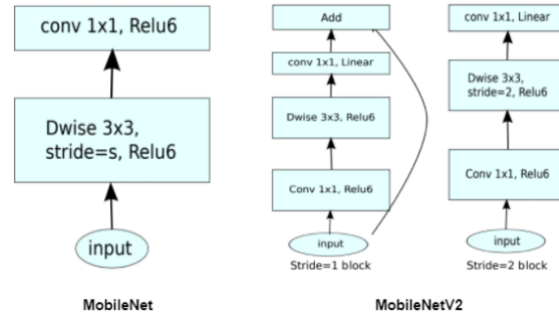
일반적인 CNN 모델은 입력이 여러 층을 거쳐 출력이 생성된다. 하지만 잔차 학습(residual learning) 방식은 이전 층에서 사용되었던 특징을 연결함으로써 새로운 출력을 정의하게 된다. (그림 1)은 잔차 학습을 보여준다. (그림 1)에서 보이는 바와 같이 입력  $x$ 는 CNN 모델 내의 여러 층을 통과하는 동시에 short connection을 통해 해당 층을 거치지 않고 다음 층으로 연결된다. 항등 매핑(identity mapping) 개념이 적용되어 short connection으로 마지막에 입력이 더해짐으로써 기존의 CNN 모델과 달리 입력 값과 출력 값의 차이, 즉 잔차를 최소화하는 방향으로 학습하게 된다. 이는 오차 역전파 시 각 층마다 가질 수 있는 최소 그래디언트가 1이 되어 그래디언트 소실 문제를 해결하였다.

(그림 2)는 층이 34개로 이루어진 ResNet-34의 구조를 보여준다. (그림 2)에서 보이는 바와 같이 ResNet은 대부분의 층이 3X3 convolution을 수행한다. 더욱 세부적으로는 convolution 블록과 identity 블록이 쌓여서 네트워크를 구성한다. 두 블록의 차이는 입력에 1X1 convolution과 배치 정규화의 적용 여부이다. Convolution 블록은 입력에 convolution과 배치 정규화가 모두 적용되는



(그림 2) VGG-19 모델과 층이 34개인 일반적인 CNN 모델, ResNet-34 모델의 구조 비교 [3]

반면, identity 블록은 입력이 그대로 출력부분에 더해진다. 층의 개수에 따라 ResNet-18, 34, 50, 101, 152 등 다양한 모델 구성이 가능하다. 단, 네트워크의 깊이 증가를 위해 ResNet-50, 101, 152에서는 2층 대신 3층으로 구성된 bottleneck 블록으



(그림 3) MobileNet과 MobileNetV2의 convolution 블록 비교 [5]

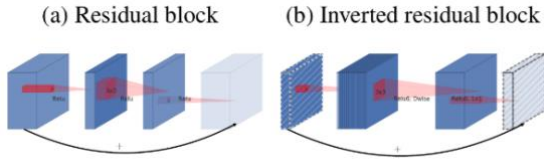
로 잔차 학습을 구성한다. 실험을 통해 층의 깊이가 깊어질수록 성능이 우수해짐을 보인다.

## 2. MobileNet

MobileNet은 Howard, et al.[4]에 의해 제안된 Depthwise separable convolution 구조를 활용하여 경량화된 모델이다. 이는 한정된 자원을 가진 모바일 환경 또는 임베디드 시스템과 같은 환경에 딥러닝을 적용하기에 적합하다. Depthwise separable convolution은 Depthwise convolution 이후에 pointwise convolution을 적용한 것이다. Depthwise convolution은 입력 영상의 각 채널별로 convolution을 실시하고, pointwise convolution은 1x1 convolution을 수행한 것과 동일하다. 따라서 연산량을 크게 감소시킬 수 있다.

MobileNetV2[5]는 MobileNet의 Depthwise separable convolution의 구조를 수정한 것이다. (그림 3)과 같이 MobileNet의 convolution 블록은 2개의 층으로 구성되어 있는 반면, MobileNetV2의 경우 stride가 1인 잔차블록과 stride가 2로 크기 축소를 목적으로 하는 블록 2종류로 되어있다.

MobileNetV2 구조는 linear bottleneck과 inverted residual이 특징이다. Linear Bottleneck은 비선형성을 더해주는 활성화 함수의 적용으로 발생하는 데이터의 손실을 줄이기 위해, 저차원의 데이터를 확장층을 통해 확대시키고 depthwise



(그림 4) 잔차 블록과 inverted residual의 차이 [5]

convolution 연산을 거쳐 최대한 데이터를 보존할 수 있도록 설계되었다. (그림 4 a)와 같이 일반적인 잔차 블록은 1X1 convolution을 이용하여 입력의 채널을 감소시켜 bottleneck을 생성하고, 다시 3X3 convolution과 skip connection을 통해 원래의 크기로 복원되는 구조를 가진다. 하지만 (그림 4b)의 MobileNetV2에서 제안한 inverted residual은 일반적인 잔차 블록과 반대의 형태를 띤다. 즉, 입력 채널의 수를 증가시킨 뒤 다시 감소시키는 형태이다. 이는 입력 데이터와 출력 데이터의 크기가 상대적으로 작기 때문에 메모리 사용을 효율적으로 할 수 있다.

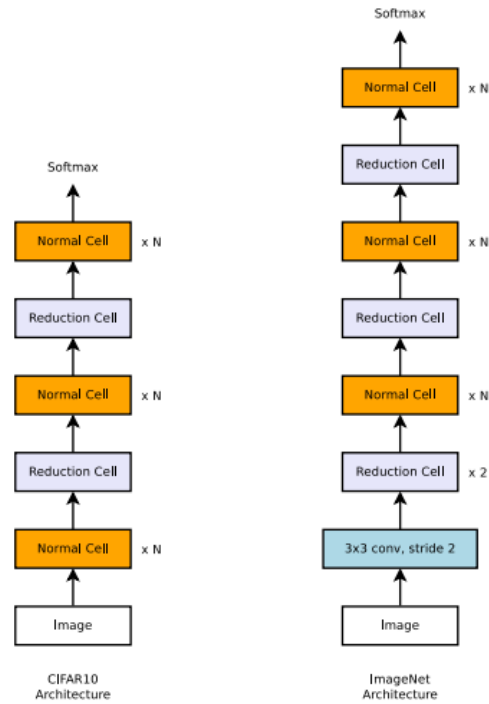
### 3. NasNet

앞서 언급된 ResNet, MobileNet 등은 설계자를 통해 설계 및 모델 구축이 생성된 것과 달리, NasNet은 RNN(recurrent neural network)과 강화학습을 활용하여 생성된 convolution 블록으로 구성된다. 기존 방법론인 NAS(neural Architecture Search)는 모델 구조 전체를 탐색 공간으로 설정하는데 반해, NasNet은 탐색 공간을 축소하여 모델을 구성하는 convolution layer 연산만을 탐색 공간으로 설정한다.

NasNet은 RNN controller로 구성된 다수의 convolution cell을 쌓아 구성된 네트워크이다. 이때 cell은 블록으로 구성되어 있고 블록은 각각 5개의 구성요소로 이루어진다. 구성요소는 hidden state 선택, 연산(operation), 병합에 관한 것으로 hidden state 연산에 대해서는 다음과 같은 연산을 지원한다:

- Identity
- $1 \times 7$  then  $7 \times 1$  convolution
- $3 \times 3$  average pooling
- $5 \times 5$  max pooling
- $1 \times 1$  convolution
- $3 \times 3$  depthwise-separable convolution
- $7 \times 7$  depthwise-separable convolution
- $1 \times 3$  then  $3 \times 1$  convolution
- $3 \times 3$  dilated convolution
- $3 \times 3$  max pooling
- $7 \times 7$  max pooling
- $3 \times 3$  convolution
- $5 \times 5$  depthwise-separable convolution

선택된 연산의 결과물들이 병합 과정에서 합쳐지면서 특징맵이 생성되고, 이러한 과정을 통해 생성된 블록들은 NasNet의 기본 구성 요소인 convolution cell을 구성한다.



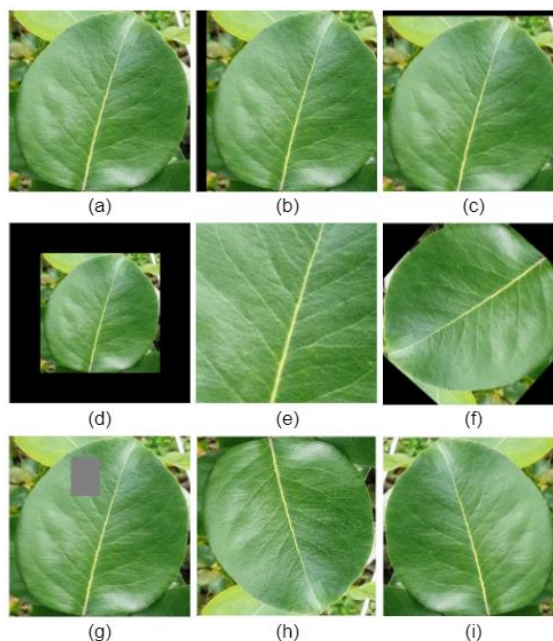
(그림 5) NasNet 구조 [6]

Convolution cell은 reduction cell과 normal cell 2가지 종류가 있다. Reduction cell은 높이와 너비가 1/2로 감소된 특징맵을 반환, normal cell은 동일한 차원의 특징맵을 반환하는 convolution cell이다. (그림 5)는 NasNet을 제안한 [6]에서 발췌한 것으로, CIFAR 데이터 세트로 reduction cell과 normal cell을 찾아 모델을 구성한 뒤, 이를 ImageNet 데이터세트에 적용한 것을 보여준다. 즉, 작은 데이터세트에서 학습한 결과 중 가장 우수한 결과를 더 큰 데이터세트에 적용하는 방식으로 변환하는 것이다.

NasNet은 convolution cell을 쌓아 네트워크를 구성함으로써 데이터의 양에 따라 크기가 변할 수 있게 하였다[2]. 또한 기존의 사람이 설계한 우수한 모델보다 정확도가 높고, FLOPS(Floating point operations per second)는 크게 감소시켰다.

### III. 데이터 증강 방법

심층 신경망은 다수의 은닉층을 쌓아 매개변수를 늘리면서 성능을 향상시켜왔다. 따라서 데이터의 다양성과 수는 성능에 영향을 미치는 중요한 요소이다. 하지만 양질의 데이터를 다수 확보하기 위해서는 시간과 비용의 소모가 크며, 경우에 따라서는 데이터 확보에 어려움이 많다. 이러한 제약 사항들을 극복하기 위해 데이터 증강(data augmentation) 기법이 사용된다. 데이터 증강 기법이란 이미 확보된 데이터를 바탕으로 훈련 데이터에 인위적인 변화를 가해 새로운 훈련 데이터를 확보하는 방법이다. 일반적으로 사용되는 데이터 증강 기법에는 (그림 6)과 같이 영상의 수평 이동, 수직 이동, 축소, 확대, 회전, 잡음 삽입(또는 일부 잘라내기(cutout)), 상하 반전, 좌우 반전 등의 기법이 있다. 해당 기법들의 적용으로 데이터 수의 부족 현상을 극복하고 다양한 데이터를 확보하여 학습 성능을 개선할 수 있다.

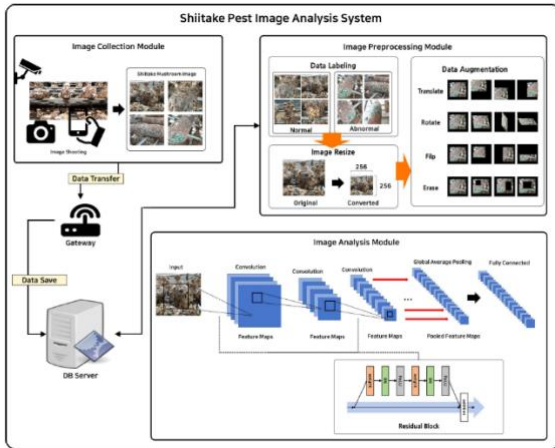


(그림 6) 데이터 증강 기법 적용 예시. (a) 원본 영상, (b) 수평 이동, (c) 수직 이동, (d) 축소, (e) 확대, (f) 회전, (g) 잡음 삽입, (h) 상하 반전, (i) 좌우 반전이 적용된 결과 영상들

### IV. 병충해 인식 시스템

#### 1. 딥러닝 기반 표고버섯 병해충 이미지 분석 시스템[1]

[1]에서 제안된 시스템은 (그림 7)과 같이 이미지 수집 모듈, 이미지 전처리 모듈, 병해충 이미지 분석 모듈로 구성되어 있다. 이미지 수집 모듈은 이미지 분석 모델의 학습을 위해 CCTV, 디지털 카메라, 스마트폰 등 다양한 장치에서 촬영된 영상을 수집하는 부분이다. 해당 모듈을 통해 수집된 영상은 데이터베이스화 되어 이미지 전처리가 적용된다. 앞서 언급된 바와 같이 영상은 다양한 장치를 통해 획득되므로 각각의 화소, 크기, 비율 등이 일정하지 않으며, 병해충 감염 유무에 관계없이 획득되어 분류가 되어있지 않다. 따라서 해당 모듈에서는 데이터를 분류하는 라벨링(labeling) 작업, 모든 영상을 동일한 크기로 변경



(그림 7) [1]에서 제안한 딥러닝 기반 표고버섯 병해충 이미지 분석 시스템의 전체적인 흐름도

하는 크기 변환 작업 그리고 앞절에서 설명한 데이터 증강 작업을 수행한다. 수직/수평 이동, 회전, 상하/좌우 반전, 잡음 삽입 총 4가지 방법의 적용으로 수집 모듈을 통해 확보한 8,000개의 데이터를 24,000개로 확장하여 사용하였다.

이미지 분석 모듈은 ResNet-50을 기반으로 한다. 영상 분류를 위한 출력단의 완전 연결 계층 (fully connected layer) 이전에 전역 평균 풀링 (global average pooling) 계층을 추가하였다. 이미지 분석 모듈을 위해 입력 영상의 크기는 256X256, 학습 데이터는 전체 데이터의 80%인 19,200개 데이터, 나머지 20%의 데이터는 평가를 위해 사용하였다. 제안된 시스템을 이용하여 표고버섯 병해충 감염 여부를 평가한 결과 81%의 성능을 보여주었다.

## 2. 병충해 인식 시스템[2]

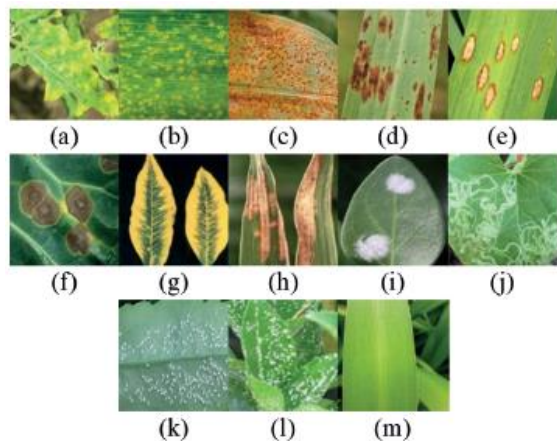
[2]에서 제안한 시스템 역시 딥러닝 모델을 기반으로 데이터 증강 방법을 적용하여 병해충의 종류를 인식한다. 특히 스마트폰 사용이 급격히 증가한 최근의 추세에 맞춰 스마트폰에서 어플리케이션으로서의 사용이 가능하도록 경량화된 모델인 MobileNetV2와 NASNetMobile을 사용하였다.

<표 1> 적용된 데이터 증강 방법의 목록과 해당 매개변수 값 [2]

Data Augmentation Techniques	Parameters
Rotation	$[-170, 170]$
Brightness	$[1, 5]$
Width shift	$[-0.2, +0.2]$
Height shift	$[-0.2, +0.2]$
Zoom	$[0.5, 1.5]$
Fill mode	Reflect
Cutout	0.5
Mixup	0.4

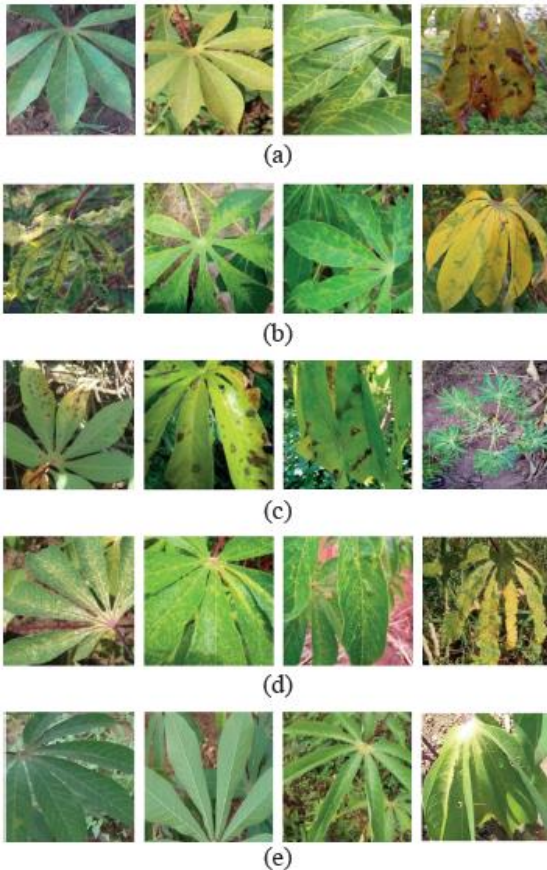
[2]에서는 학습 방식과 데이터 증강 방법의 다양한 조합을 시험 비교하였다. 오프라인 학습, 온라인 학습 그리고 오프라인과 온라인을 혼합하는 학습 방법 3종류를 적용하였다. 데이터 증강 방법으로는 수직/수평 이동, 회전, 확대/축소, 잡음 삽입, 밝기 변화 그리고 언급된 방법들을 조합하여 적용하였다. 적용된 데이터 증강 방법과 각각의 매개변수 값은 <표 1>과 같다.

제안 시스템의 평가를 위해 사용한 데이터는 자체 획득한 leaf disease 데이터세트와 공개 데이터인 iCassava 2019 데이터세트[7]이다. Leaf disease 데이터세트는 (그림 8)과 같이 12 종류의 병해충과 건강한 작물을 포함하여 총 13 클래스(class)의 데이터를 스마트폰을 통해 608개 확보



(그림 8) leaf disease 데이터세트의 예시 영상. (a)~(l)은 서로 다른 병해충 종류, (m)은 건강한 작물의 영상





(그림 9) iCassava 2019 데이터세트의 예시 영상. (a)~(d)는 각각 병해충 종류, (e)는 건강한 카사바 나뭇잎이었다.

iCassava 2019 데이터세트는 카사바 나뭇잎에 대한 영상으로 총 4종류의 병충해 영상과 건강한 카사바 나무의 잎을 포함하여 총 5 클래스의 데이터로 구성되어 있다. 해당 데이터세트는 약 22,000여개의 데이터를 포함하지만 일부 데이터는 라벨링 작업이 되어있지 않다. 따라서 라벨링이 되어 공개된 5,656개의 데이터를 사용하였다. 해당 데이터의 예시 영상은 (그림 9)와 같다.

병해충 인식을 위해 사용한 MobileNetV2와 NASNetMobile을 iCassava 데이터세트에서 동일한 환경으로 학습 후 평가한 결과, 데이터 증강 방법에 따라 성능 차이를 보였다. 데이터 증강없이 진행된 경우나 확대/축소만 적용된 경우

MobileNetV2가 더 우수한 성능을 보였으나, 회전과 이동이 데이터 증강으로 적용된 경우 약 1% 정도의 차이로 NASNetMobile이 우수한 성능을 보였다. 하지만 10-fold cross validation으로 평가를 진행한 결과, MobileNetV2가 약 0.9% 정도 우수한 성능을 보였다. 또한 수행 시간은 모델 크기가 작은 MobileNetV2에서 2.25배 더 빠르게 나타났다.

#### IV. 맺음말

본 보고서에서는 딥러닝 기반 병해충 인식 시스템을 살펴보았다. 병해충으로 인한 작물의 피해는 농가에 미치는 직접적인 영향을 넘어서 국가의 피해로 발전될 수 있다. 따라서 병해충 발생에 대해 신속한 대응이 필요하며 이를 위해 병해충 인식을 위한 기술과 시스템 개발이 조속히 이루어질 필요가 있다. 최근 빠르게 발전하는 딥러닝 기술을 해당 분야에 도입함으로써 병해충을 조기에 발견하는 시스템들이 제안되고 있다. 이러한 시스템이 농가에 적용된다면 농가의 손실을 줄일 수 있을 뿐 아니라, 넓게는 국가의 식량 안보에도 도움이 될 수 있다. 특히, 스마트폰의 어플리케이션으로 사용할 수 있는 시스템의 개발은 농가에 쉽고 빠르게 전달할 수 있는 장점이 있다. 하지만 다양한 작물과 병해충에 대한 데이터 확보가 어렵다는 것은 딥러닝 기반 시스템 개발의 문제이다. 따라서 작물과 병해충의 다양성을 포함하여 다량의 데이터 세트 확보 또한 시스템 개발과 함께 조속히 이루어져야 할 것으로 보인다.

#### 참고문헌

- [1] K. H. Jo, S. H. Jung and C. B. Sim, "A Study of Shiitake Disease and Pest Image Analysis based on Deep Learning," Journal of Korea Multimedia Society Vol. 23, No. 1, pp. 50-57, 2020
- [2] P. Enkvetchakul and O. Surinta, "Effective Data

Augmentation and Training Techniques for Improving Deep Learning in Plant Leaf Disease Recognition," *Applied Science and Engineering Progress*, 2021

- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Application," *arXiv*, pp. 1–9, 2017
- [5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018
- [6] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition," *arXiv:1707.07012v4*
- [7] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tusubira, "iCassava 2019 fine-grained visual categorization challenge," *arXiv*, pp. 4321–4326, 2019, doi: arXiv:1908.02900.