Journal of the Korea Academia-Industrial cooperation Society
Vol. 23, No. 2 pp. 22-29, 2022

https://doi.org/10.5762/KAIS.2022.23.2.22
ISSN 1975-4701 / eISSN 2288-4688

전이 학습 기반의 모바일 작물 질병 진단을 위한 PWA

김삼근[1*], 허환[2], 안재근[1]
[1]한경대학교 컴퓨터응용수학부 (컴퓨터시스템연구소)
[2]경기동부원예농협 과수원예지도과

A Progressive Web Application for Mobile Crop Disease Diagnostics Based on Transfer Learning

Sam-Keun Kim[1*], Whan Hur[2], Jae-Geun Ahn[1]
[1]School of Computer Engineering & Applied Mathematics, Hankyong National University
[2]Fruit Horticulture Guidance Department, Kyonggi East Horticultural Agriculture Cooperatives

요 약 작물 질병은 식량 안보에 대한 주요 위협이지만 필요한 기반 시설의 부족으로 인해 세계 여러 지역에서 작물 질병의 빠른 진단은 여전히 어렵다. 최근 전 세계적으로 널리 보급된 스마트 폰과 딥 러닝을 통해 가능해진 최근의 컴퓨터 비전의 발전은 스마트 폰을 이용한 작물 질병 진단의 길을 열었다. 본 논문에서는 PlantVillage 데이터 세트의 70,295개의 이미지를 이용하여 14가지 작물 종의 38종류의 작물 질병을 진단하기 위해 전이 학습과 미세 조정을 통해 심층 MobileNetV2 CNN 모델을 학습시켰다. 학습된 모델은 17,572개의 테스트 이미지에 대해 95.65%의 정확성을 보여주었다. 또한 본 논문에서는 학습된 심층 CNN 모델을 기반으로 작물 질병 진단을 스마트 폰에서 실시간으로 수행할 수 있는 PWA(Progressive Web Application)를 제안한다. 제안한 PWA는 스마트 폰과 모바일 장치에서 개조한 심층 CNN 모델을 오프라인 상태에서도 실행할 수 있는 환경을 제공함으로써 사용자가 언제 어디서나 서비스를 제공받을 수 있도록 하여 사용자 경험을 크게 개선시킬 수 있다.

Abstract Crop diseases are a major threat to food security, but rapid diagnosis of crop diseases remains difficult in many parts of the world due to a lack of necessary infrastructure. The combination of smart phones widely distributed around the world and recent advances in computer vision made possible by deep learning has paved the way for smart phone-assisted crop disease diagnostics. In this paper, a deep MobileNetV2 CNN model is trained through transfer learning and fine tuning to diagnose 38 types of disease in 14 crop species using 70,295 images from a PlantVillage dataset. The trained model shows an accuracy of 95.65% for 17,572 test images. In addition, this paper proposes a Progressive Web Application (PWA) based on a trained deep CNN model, which can perform crop disease diagnosis on a smart phone in real time. The proposed PWA can greatly improve the user experience by providing an environment in which the deep CNN model modified in smart phones and mobile devices can be run, even offline, allowing users to receive services anytime, anywhere.

Keywords : Mobile Plant Disease Diagnostics, Deep Convolutional Neural Networks, Feature Extraction, Deep Learning, Progressive Web Application

*Corresponding Author : Sam-Keun Kim(Hankyong National Univ.)
email: skim@hknu.ac.kr
Received November 9, 2021 Revised December 6, 2021
Accepted February 4, 2022 Published February 28, 2022

1. 서론

전 세계 식량 안보의 상태는 우려할 만하고 작물 해충, 특히 식물 질병으로 인한 손실에 대한 보호는 전 세계적으로 식량 안보를 개선하는 데 중요한 역할을 할 수 있다[1]. 최근 다양한 농업 기술의 발전으로 농부들은 적절한 해결책을 제안받기 위해 작물을 진단 실험실에 보내는 전통적인 방법 대신에 식물 질병 데이터베이스를 이용하거나 전화로 지역 병리학자와 상담하는 방식을 취한다. 또한 스마트 폰이 널리 보급되면서 스마트 폰을 작물 손실을 줄일 수 있는 진단 도구로 활용하는 많은 시도가 있다.

작물 질병 진단과 관련하여 DenseNet[2], SqueezeNet[3], ResNet[4], GoogleNet[5], VGGNet[6], ZFNet[7], AlexNet[8]과 같은 표준 CNN(Convolutional Neural Networks) 아키텍처 중의 하나를 이용하여 작물 손실을 줄이기 위한 많은 애플리케이션들이 제안되었다.

[9]는 PlantVillage 데이터 세트[10]에서 자동 식물 질병 감지를 위해 VGG-16, ResNet-50, GoogleNet CNN 아키텍처를 사용하고, 분류를 위해 k-Nearest Neighbor 및 SVM(Support Vector Machine) 분류기를 사용하여 최종적으로 SVM 분류기와 함께 VGG-16 모델에 대해 97.92%의 우수한 정확도를 얻었다. [11]은 식물 잎 질병의 탐지 및 분류를 위한 딥 러닝 기반 연구를 살펴보고 딥 러닝의 가능성을 평가했다. 대부분의 연구는 PlantVillage 데이터 세트를 사용하고 이 데이터 세트의 모든 식물 질병을 분류하기보다는 특정 식물 또는 여러 식물에 대한 질병을 진단하는 것으로 관찰되었다. [12]는 이러한 연구 중 하나로 CNN을 사용하여 13가지 다른 식물 질병의 분류를 수행했다. [13]은 14가지 작물 종의 26가지 질병을 식별하기 위해 두 개의 CNN 모델 AlexNet과 GoogleNet을 학습시켰다. 병든 식물 잎과 건강한 식물 잎의 54,306개 이미지 데이터 세트가 사용되었다. AlexNet과 Google Net을 사용하는 두 모델이 스마트 폰에서 작동하지 않더라도 스마트 폰 지원 질병 진단을 위한 첫 번째 시도였다.

기존의 모든 식물 질병 탐지 방법은 계산력이 강력한 컴퓨터의 사용에 의존한다. 따라서 식물 질병을 감지하기 위해 심층 CNN을 기반으로 많은 연구가 이루어지고 있지만 모든 스마트 시스템은 저장 용량과 자원 계산력이 제한되어 있어서 심층 CNN 모델을 실행할 수 있는 환경을 제공하지 못했다. 그러나 양자화된 CNN 모델[14]의 개발로 제한된 자원으로 휴대폰에 스마트 애플리케이션을 내장하는 것이 가능해졌다. [15]는 모바일 및 임베디드 비전 애플리케이션을 위한 MobileNet이라는 효율적인 모델을 제안했다. 즉, MobileNet 모델을 깊이별 분리 가능한 컨볼루션으로 구성하여 표준 컨볼루션보다 8~9배 더 적은 계산으로 줄였다. [16]은 계산력과 메모리 용량이 제한적인 모바일 환경에서 실행될 수 있도록 모델 크기를 최적화시킨 MobileNetV2 CNN 모델을 제안하였다.

[17]은 토마토 잎 질병을 인식하기 위해 심층 CNN 기반의 스마트 모바일 애플리케이션 모델을 제안하였다. 애플리케이션을 구축하기 위해 토마토 잎의 7,176개 이미지 데이터 세트에 대해 학습시키고 스마트 모바일 시스템에 사용하여 토마토 질병 진단을 수행하였다. [18]은 식물 질병 진단 프로세스를 자동화하는 머신 러닝 기반의 모바일 시스템을 제시하였다. 개발된 시스템은 CNN을 기본 딥 러닝 엔진으로 사용하여 38개 질병 범주를 분류한다. 사용자 인터페이스는 Android 모바일 앱으로 개발되어 농부들이 감염된 식물 잎의 사진을 캡처할 수 있고 인식 결과를 백분율과 함께 질병 범주를 표시한다. 이 시스템은 농부들이 작물을 건강하게 유지하고 식물에 스트레스를 줄 수 있는 잘못된 비료 사용을 제거할 수 있는 더 나은 기회를 제공할 것으로 기대한다. 제안 시스템은 작물 14종에서 가장 흔한 38개 질병 분류를 인식할 때 94%의 전체 분류 정확도를 얻었다.

본 논문에서는 CNN 기반 전이 학습을 이용하여 PlantVillage 데이터 세트의 작물 14종 38개 식물 질병에 대해 전이 학습과 미세 조정을 통해 심층 MobileNetV2 CNN 모델을 학습시켜 95.65%의 정확성을 얻었다. 이처럼 개조한 CNN 모델을 이용하여 작물 재배자가 오프라인 환경에서도 스마트 폰 및 모바일 장치를 이용하여 작물 질병을 편리하게 진단할 수 있는 PWA(Progressive Web Application)를 제안한다. 제안한 PWA는 스마트 폰과 모바일 장치에서 개조한 심층 CNN 모델을 오프라인 상태에서도 실행할 수 있는 환경을 제공함으로써 사용자가 언제 어디서나 서비스를 제공받을 수 있도록 하여 사용자 경험을 크게 개선시킬 수 있다.

본 논문의 구성은 다음과 같다: 2장에서는 심층 CNN 모델의 전이 학습 및 미세 조정에 대해 기술하고, 3장에서는 학습된 CNN 모델을 기반으로 스마트 폰 상에서 실시간으로 작물 질병을 진단하는 PWA를 제안한다. 4장에서는 데이터 세트 및 실험결과를 제시하고 분석한다. 마지막 5장에서는 결론 및 향후 연구 방향을 기술한다.

2. MobileNetV2 CNN 기반 전이 학습

표준 MobileNet CNN 모델은 스마트 폰에서 허용 가능한 속도로 동작하며 높은 분류 정확도를 얻을 수 있다[15]. 본 논문에서는 MobileNetV2 CNN 모델을 작물 질병 진단 문제에 적용한다.

MobileNetV2는 모바일 장치에서 잘 작동하도록 하는 CNN 아키텍처이다. 잔류 연결이 병목 레이어 사이에 있는 역 잔류 구조를 기반으로 한다. 중간 확장 레이어는 비선형 소스로 기능을 필터링하기 위해 경량의 깊이별 컨볼루션을 사용한다[16].

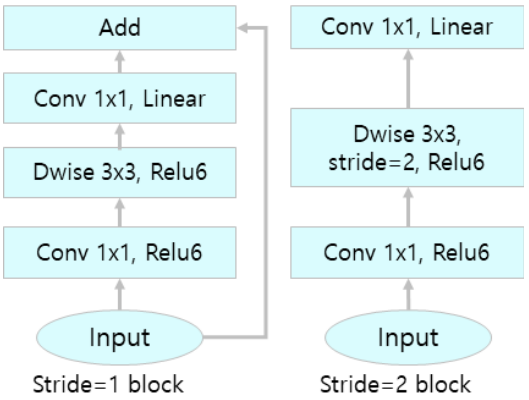

Fig. 1. Convolution blocks of MobileNetV2 architecture [16]

Fig. 1에서처럼 MobileNetV2에는 2가지 종류의 블록이 있다. 첫 번째는 stride가 1인 잔류 블록이고 두 번째는 다운사이징을 위한 stride가 2인 블록이다. 각각의 블록은 3개의 레이어를 가지고 있다. 두 블록 모두 첫 번째 레이어는 Pointwise(1x1) 컨볼루션과 ReLU6으로 구성되어 있다. 두 번째 레이어는 깊이별 컨볼루션이다. 첫 번째 블록에서는 stride가 1로 적용되고 두 번째 블록에서는 stride가 2가 적용되어 다운사이징된다. 두 블록 모두의 세 번째 레이어에서는 다시 Pointwise(1x1) 컨볼루션이 적용된다.

MobileNetV2의 역 잔류 병목 레이어는 메모리 효율적인 구현을 허용하는데, 이는 모바일 애플리케이션에 특히 중요하다. 즉, 표준적인 구현체는 연산을 의미하는 에지와 중간 계산의 텐서를 나타내는 노드로 구성된 방향성 비순환 계산 하이퍼그래프 G를 생성해준다. 또한 메모리에 저장해야 하는 텐서의 총 수를 최소화하기 위한 계산이 수행된다. 가장 일반적인 경우, 전체 타당한 계산 차수 $\Sigma(G)$를 찾아서 식(1)의 우측을 최소화하는 값 ($M(G)$)을 선택한다[16].

$$M(G) = \min_{\pi \in \Sigma(G)} \max_{i \in 1..n} \left[\sum_{A \in R(i,\pi,G)} |A| \right] + size(\pi_i) \quad (1)$$

여기서 $R(i,\pi,G)$는 n개의 노드 $\pi_i \ldots \pi_n$ 중 하나에 연결된 중간 텐서 목록이다. $|A|$는 텐서 A의 크기를 나타내고 $size(\pi_i)$는 작업 i 동안 내부 저장소에 필요한 총 메모리 용량이다.

본 논문에서는 Fig. 2에 보이는 바와 같이 MobileNetV2 CNN 기반 아키텍처에 톱 레벨 분류기를 추가하여 전이 학습과 미세조정을 적용한다. MobileNetV2 CNN 모델은 224x224 크기의 컬러 이미지를 받아들이고, 깊이별/포인트별 레이어와 BN(Batch Normalization) 및 ReLU가 있는 일련의 깊이별로 분리가능한 컨볼루션으로 구성된다. MobileNetV2 베이스 모델 위에 추가한 톱 레벨 분류기는 먼저 MobileNetV2로부터 최종 출력되는 7x7 크기의 1,280개 레이어를 GAP(Global Average Pooling) 연산을 이용하여 FC(Fully Connected) 레이어와 연결한다. 또한 모델이 과적합되는 것을 방지하기 위해 Dropout, BN을 FC 레이어 전후에 추가한다.

3. 제안한 PWA

개조한 심층 CNN 모델은 먼저 애플리케이션에서 사용될 수 있도록 배포되어야 한다. 이를 수행하는 방법은 요구 사항에 따라 다르다. 그러한 요구 사항 중 하나는 인터넷 연결이 원활하지 않은 지역에서 오프라인으로 모델을 실행할 수 있는 기능이다. 이를 위해 한 가지 솔루션은 해당 모델의 압축 버전을 패키징하고 로딩할 모바일 플랫폼용 네이티브앱을 만드는 것이다. 그러나 이 방법은 Android 및 iOS 앱을 개발해야 하는 오버헤드가 있다.

본 논문에서는 React[19] 라이브러리를 이용하고 TFJS 컨버터[20]를 사용하여 개조한 CNN 모델을 이용한 PWA (Progressive Web Application)를 구축함으로써 오프라인 모바일 요구 사항을 충족시켰다. PWA는 네이티브앱과 같은 느낌을 주고 대부분의 최신 웹브라우저에서 실행 가능하다. HTML/ JavaScript에서 애플리케이션을 한 번만 개발하면 되므로 상이한 플랫폼 간 개발이 훨씬 쉽다. 또한 제안한 PWA는 서비스 워커를 사용하여 완전한 오프라인 기능을 제공한다.

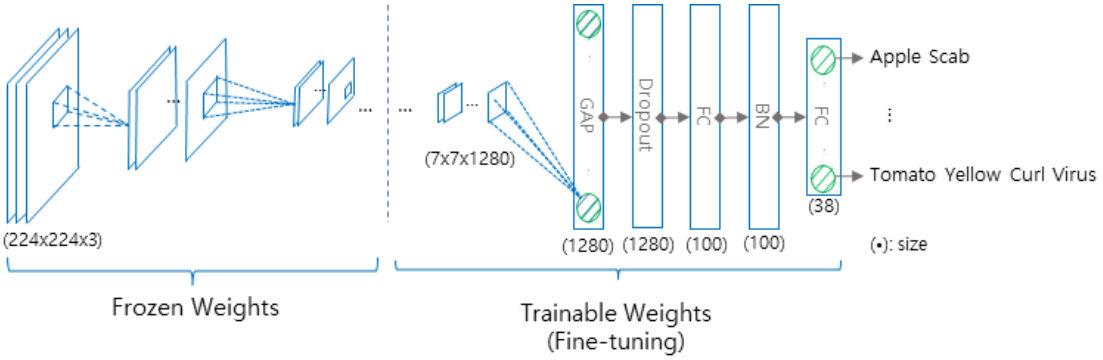

Fig. 2. The structure of deep convolutional neural networks

TFJS 컨버터를 사용하면 사전 학습된 TensorFlow 또는 Keras 모델을 브라우저에서 실행 가능한 TFJS 포맷으로 변환할 수 있다. TFJS 포맷으로 변환된 CNN 모델은 모바일 장치 내에 설치되어 로컬 방식으로도 실행 가능하며, 진단을 위한 추론 역시 장치 내에서 로컬하게 수행될 수 있다. 제안한 CNN 모델은 브라우저 캐시의 최대 용량인 4MB보다 커서, 브라우저 캐시에 모두 저장될 수 없다. 본 논문에서는 이러한 용량 문제를 해결하기 위해 클라이언트 측 저장 공간을 위한 브라우저 API인 IndexedDB[21]에 모델을 저장한다. 사용자의 첫 번째 방문 때 클라우드 환경에서 모델을 다운로드 받을 수 있다. 두 번째 방문부터 모델 로딩은 IndexedDB에서 가져오게 된다. 이렇게 하면 모델을 계속해서 다시 다운로드할 필요가 없다.

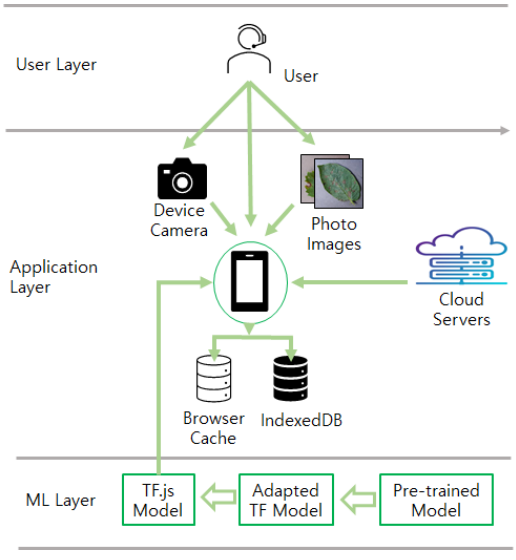

Fig. 3. The proposed PWA architecture

제안한 PWA는 플랫폼 독립적으로 실행될 수 있다. 즉, 사용자가 어떤 플랫폼에 있든 PWA가 최신 브라우저에서 실행되는 한 모든 것이 작동한다. 또한 매니페스트 파일을 사용하면 PWA를 스마트 폰 또는 모바일 장치에 '설치'할 수 있으므로 홈 화면에 자체 앱 아이콘이 있는 네이티브앱처럼 보이게 할 수 있다.

본 논문에서는 개조한 MobileNetV2 CNN 모델을 자체 카메라가 장착된 스마트 폰에서 높은 인식률로 빠르게 동작할 수 있도록 한다. 개조한 CNN 모델 기반으로 구축된 애플리케이션은 스마트 폰에서 오프라인으로 PlantVillage 작물 14종의 38가지 유형의 작물 질병을 식별하고 진단할 수 있다.

4. 데이터 세트 및 실험 결과

본 논문에서는 오프라인 작물 질병 진단 PWA에서 사용하는 사전 학습된 MobileNetV2 CNN 모델을 개조하기 위해 PlantVillage 데이터 세트[22]의 70,295개의 이미지를 이용하여 모델을 학습시켰다. 제안한 PWA는 스마트 모바일 시스템에서 작물 14종 38개의 질병을 진단할 수 있다.

CNN 모델의 과적합을 피하기 위해 불충분한 훈련 세트에 대해 이미지의 개수를 증강시켰다. 여기서는 훈련 데이터에 존재하는 위치적 편향성을 제거하기 위해 기하학적 변환 방법을 사용했다. Fig. 5에서처럼 클래스에 적용된 기하학적 변환 방법으로는 수평 뒤집기, -45° ~ 45° 회전, 가장 가까운 이웃 영역으로 채우기, 범위 0.2로 확대/축소, 상대 배율 0.2로 너비 및 높이 이동을 적용했다.

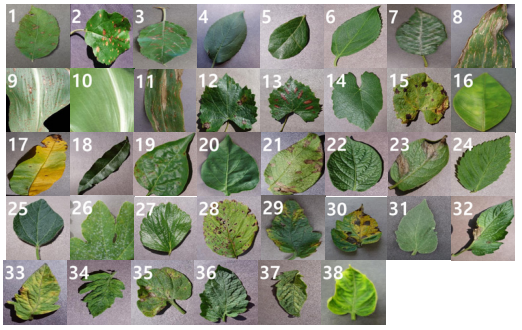

Fig. 4. Example of leaf images from the PlantVillage dataset, representing every crop-disease pair used. (1) Apple Scab (2) Apple Black Rot (3) Apple Cedar Rust (4) Apple healthy (5) Blueberry healthy (6) Cherry healthy (7) Cherry Powdery Mildew (8) Corn Gray Leaf Spot (9) Corn Common Rust (10) Corn healthy (11) Corn Northern Leaf Blight (12) Grape Black Rot (13) Grape Black Measles(Esca) (14) Grape Healthy (15) Grape Leaf Blight (16) Orange Citrus Greening (17) Peach Bacterial Spot (18) Peach healthy (19) Bell Pepper Bacterial Spot (20) Bell Pepper healthy (21) Potato Early Blight (22) Potato healthy (23) Potato Late Blight (24) Raspberry healthy (25) Soybean healthy (26) Squash Powdery Mildew (27) Strawberry Healthy (28) Strawberry Leaf Scorch (29) Tomato Bacterial Spot (30) Tomato Early Blight (31) Tomato healthy (32) Tomato Late Blight (33) Tomato Leaf Mold (34) Tomato Mosaic Virus (35) Tomato Septoria Leaf Spot (36) Tomato Target Spot (37) Tomato Two Spotted Spider Mite (38) Tomato Yellow Leaf Curl Virus.

Table 1. The detailed table of the dataset with diseases, healthy, train and test set

Disease type	# of Diseases	# of Healthy	Train set	Test set
Apple	3	1	7771	1943
Blueberry	0	1	1816	454
Cherry	1	1	3509	877
Corn	3	1	7316	1829
Grape	3	1	7222	1805
Orange	1	0	2010	503
Peach	1	1	3566	891
Bell pepper	1	1	3901	975
Potato	2	1	5702	1426
Raspberry	0	1	1781	445
Soybean	0	1	2022	505
Squash	2	0	1736	434
Strawberry	1	1	3598	900
Tomato	9	1	18345	4585
Total	27	12	70295	17572

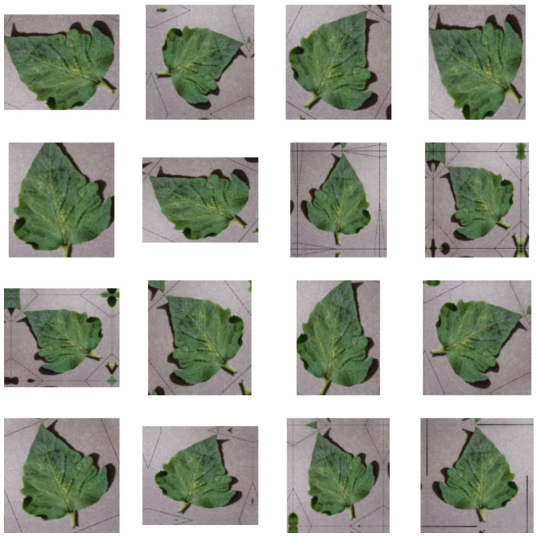

Fig. 5. Dataset augmentation

Fig. 6과 7은 CNN 모델의 훈련 및 검증 세트에 대한 정확도 및 손실 학습 곡선을 보여준다. 학습은 두 단계로 이루어진다. 단계 1은 전이 학습 단계로 먼저 기본 MobileNetV2 CNN 모델의 전체 레이어(155개)를 동결시킨 후 모델을 개조하기 위해 추가한 톱 레벨 분류기만을 학습시킨다. 단계 2에서는 미세 조정 단계로 CNN 모델의 전체 155개의 레이어 중 152개의 레이어만을 동결한 후 나머지 153번째 레이어 이후의 레이어를 재학습시킨다. 이때 적용하는 학습률은 전이 학습 단계에서의 1/10 크기로 낮춘다.

CNN 모델의 전체 성능을 개선할 수 있는 다양한 방법들이 있다. Table 2는 제안한 MobileNetV2 CNN 모델에 SGD(stochastic gradient descent), Adagrad, RMSprop, Adam과 같은 최적화 알고리즘에 다양한 학습률을 적용시켜 테스트한 결과를 보여준다. Adam 최적화 알고리즘을 Table 1의 테스트 세트에 대해 학습률 0.01로 학습시켰을 경우에 95.65%로 가장 좋은 성능을 보여주었다.

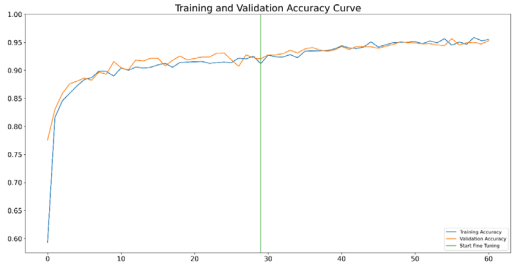

Fig. 6. Training and validation accuracy curve

Fig. 7. Training and validation loss curve

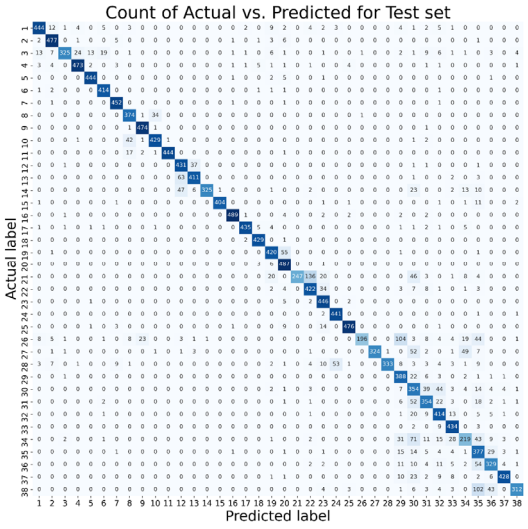

Fig. 8. Sample Confusion Matrix of the Adapted CNN model: Percent Accuracy of Actual vs. Predicted for Test set (The meanings of numbers 1 - 38 of the x and y axis coordinates are the same as the labels in Fig. 4)

Fig. 8은 개조한 MobileNetV2 CNN 모델을 이용하여 Table 1의 테스트 세트 17,572개의 이미지를 해당 클래스 범주로 분류한 결과를 예측과 실제 클래스의 CM(confusion matrix)으로 보여준다.

Table 2. Plant diseases diagnostics performance with respect to optimization algorithm (α: learning rate)

Optimization algorithm	Final test accuracy (%)		
	α=0.01	α=0.005	α=0.001
SGD	79.99	79.93	79.87
Adagrad	82.93	83.14	83.14
RMSprop	95.31	95.55	95.33
Adam	**95.65**	95.41	95.47

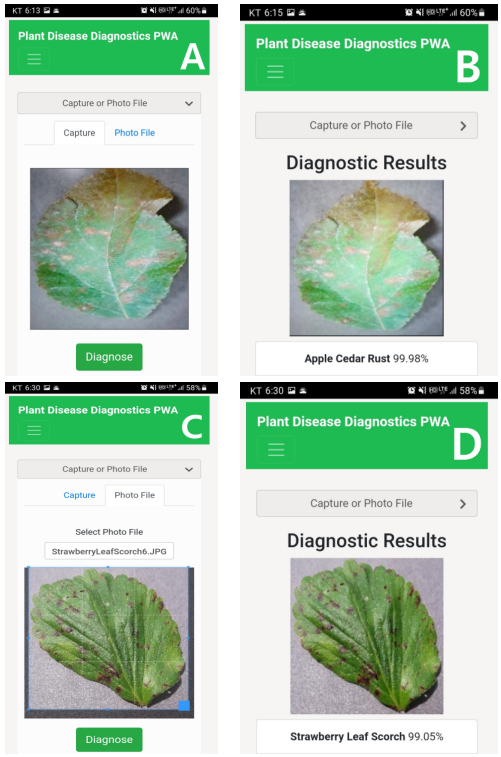

Fig. 9. Screenshots of the proposed PWA for diagnosing plant leaf diseases: A) Camera Capture Screen B) Diagnostic Result of A image by the PWA C) Image Selection Screen D) Diagnostic Result of C image by the PWA

제안한 PWA는 React 라이브러리를 이용하여 카메라가 장착된 스마트 폰 상에서 오프라인 환경에서도 실시간으로 작동할 수 있도록 구현하였다. 작물 질병 진단은 두 가지 방법으로 수행할 수 있다. 첫 번째는 스마트 폰에 기본적으로 장착된 카메라를 이용하여 캡처한 이미지(Fig. 9의 A)를 진단(Fig. 9의 B)하는 방법이다. 두 번째는 이미 캡처된 작물 이미지를 로컬 폴더로부터 선택(Fig. 9의 C)하여 질병을 진단(Fig. 9의 D)하는 방법이다.

5. 결론

본 논문에서는 PlantVillage 데이터 세트를 이용하여 전이 학습과 미세 조정을 통해 심층 MobileNetV2 CNN 모델을 학습시켜 95.65%의 정확성을 얻었다. 이렇게 개조한 심층 CNN 모델에 기반하여 인터넷 연결이

열악한 작물 재배 현장에서 작물 재배자가 오프라인으로 스마트 폰 및 모바일 장치를 이용하여 작물 질병을 편리하게 진단할 수 있는 PWA를 제안하였다. 제안한 PWA는 스마트 폰과 모바일 장치에서 오프라인 상태에서도 실행할 수 있는 환경을 제공함으로써 사용자가 언제 어디서나 서비스를 제공 받을 수 있도록 하여 사용자 경험을 크게 개선시킬 수 있었다.

향후 연구과제로는 작물 재배자들이 현장에서 캡처한 작물 질병 잎 이미지들을 크라우드소싱하여 심층 CNN 모델의 정확성을 개선하는 것과 PlantVillage 데이터 세트에 포함되어 있지 않은 작물 종들도 포함시켜 제안한 PWA의 적용범위를 확대시키는 것이다.

References

[1] Serge Savary, Andrea Ficke, Jean-Noël Aubertot, and Clayton Hollier, "Crop losses due to diseases and their implications for global food production losses and food security," Food Sec. 4, pp.519–537, 2012. DOI: https://doi.org/10.1007/s12571-012-0200-5

[2] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely Connected Convolutional Networks," Computer Vision and Pattern Recognition (CVPR), August 2016. https://arxiv.org/abs/1608.06993v5

[3] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016. https://arxiv.org/abs/1602.07360

[4] 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), June 2016. https://arxiv.org/abs/1512.03385v1

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9, 2015. https://arxiv.org/abs/1409.4842

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014. https://arxiv.org/abs/1409.1556

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in European conference on computer vision. Springer, pp. 818–833, 2014. https://arxiv.org/abs/1311.2901

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, Volume 60, Issue 6, pp 84–90, June 2017. DOI: https://doi.org/10.1145/3065386

[9] Faye Mohameth, Chen Bingcai, and Kane Amath Sada, "Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village," Journal of Computer and Communications, 8, pp.10-22, 2020. DOI: https://doi.org/10.4236/jcc.2020.86002

[10] PlantVillage Dataset, https://www.kaggle.com/abdallahalidev/plantvillage-dataset/activity

[11] Muhammad Hammad Saleem, Johan Potgieter and Khalid Mahmood Arif, "Plant Disease Detection and Classification by Deep Learning," Plants 2019, 8, 468, 2019. DOI: https://doi.org/10.3390/plants8110468

[12] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk,and Darko Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Computational Intelligence and Neuroscience, Vol. 2016(6), pp.1-11, 2016. DOI: http://dx.doi.org/10.1155/2016/3289801

[13] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in plant science, vol. 7, 2016. DOI: https://doi.org/10.3389/fpls.2016.01419

[14] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng, "Quantized Convolutional Neural Networks for Mobile Devices," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol.1, pp. 4820-4828, 2016. DOI: https://doi.org/10.1109/CVPR.2016.521

[15] 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 applications," 2017. https://arxiv.org/abs/1704.04861

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510-4520, 2018. https://arxiv.org/abs/1801.04381v4

[17] Azeddine Elhassouny and Florentin Smarandache, "Smart mobile application to recognize tomato leaf diseases using Convolutional Neural Networks," 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), 2019. DOI: https://doi.org/10.1109/ICCSRE.2019.8807737

[18] Ahmed Abdelmoamen Ahmed and Gopireddy Harshavardhan Reddy, "A Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning," AgriEngineering, 3, pp.478-493, 2021. DOI: https://doi.org/10.3390/agriengineering3030032

[19] React, https://reactjs.org/

[20] TensorFlow.js. https://www.tensorflow.org/js

[21] IndexedDB, https://www.w3.org/TR/IndexedDB/

[22] New Plant Diseases Dataset, https://www.kaggle.com/vipoooool/new-plant-diseases-dataset

김 삼 근(Sam-Keun Kim) [종신회원]

- 1988년 2월 : 숭실대학교 대학원 전자계산학과 (공학석사)
- 1998년 2월 : 숭실대학교 대학원 전자계산학과 (공학박사)
- 1992년 3월 ~ 현재 : 한경대학교 컴퓨터응용수학부 교수

〈관심분야〉
인공지능, 데이터마이닝, IoT, BI시스템

허 환(Whan Hur) [정회원]

- 2007년 2월 : 국립교통대학교 산업대학원 (공학석사)
- 2011년 8월 : 한경대학교 바이오정보통신대학원 컴퓨터공학과 (공학박사)
- 1991년 3월 ~ 현재 : 경기동부원예농협 지도상무

〈관심분야〉
데이터마이닝, GIS

안 재 근(Jae-Geun Ahn) [종신회원]

- 1994년 2월 : 서울대학교 대학원 산업공학과 (공학석사)
- 1997년 8월 : 서울대학교 대학원 산업공학과 (공학박사)
- 1997년 9월 ~ 현재 : 한경대학교 컴퓨터응용수학부 교수

〈관심분야〉
경영정보시스템, 최적화, 데이터베이스, 데이터마이닝