

심층 CNN 기반 구조를 이용한 토마토 작물 병해충 분류 모델

김삼근*, 안재근

한경대학교 컴퓨터응용수학부(컴퓨터시스템연구소)

Tomato Crop Diseases Classification Models Using Deep CNN-based Architectures

Sam-Keun Kim*, Jae-Geun Ahn

School of Computer Engineering & Applied Mathematics, Hankyong National University

요약 토마토 작물은 병해충의 영향을 많이 받기 때문에 이를 예방하지 않으면 농업 경제에 막대한 손실을 초래할 수 있다. 따라서 토마토의 다양한 병해충의 진단을 빠르고 정확하게 진단하는 시스템이 요구된다. 본 논문에서는 ImageNet 데이터 셋 상에서 다양하게 사전 학습된 딥러닝 기반 CNN 모델을 적용하여 토마토의 9가지 병해충 및 정상인 경우의 클래스를 분류하는 시스템을 제안한다. PlantVillage 데이터 셋으로부터 발췌한 토마토 잎의 이미지 셋을 3가지 딥러닝 기반 CNN 구조를 갖는 ResNet, Xception, DenseNet의 입력으로 사용한다. 기본 CNN 모델 위에 톱-레벨 분류기를 추가하여 제안 모델을 구성하였으며, 훈련 데이터 셋에 대해 5-fold 교차검증 기법을 적용하여 학습시켰다. 3가지 제안 모델의 학습은 모두 기본 CNN 모델의 계층을 동결하여 학습시키는 전이 학습과 동결을 해제한 후 학습률을 매우 작은 수로 설정하여 학습시키는 미세 조정 학습 두 단계로 진행하였다. 모델 최적화 알고리즘으로는 SGD, RMSprop, Adam을 적용하였다. 실험 결과는 RMSprop 알고리즘이 적용된 DenseNet CNN 모델이 98.63%의 정확도로 가장 우수한 결과를 보였다.

Abstract Tomato crops are highly affected by tomato diseases, and if not prevented, a disease can cause severe losses for the agricultural economy. Therefore, there is a need for a system that quickly and accurately diagnoses various tomato diseases. In this paper, we propose a system that classifies nine diseases as well as healthy tomato plants by applying various pretrained deep learning-based CNN models trained on an ImageNet dataset. The tomato leaf image dataset obtained from PlantVillage is provided as input to ResNet, Xception, and DenseNet, which have deep learning-based CNN architectures. The proposed models were constructed by adding a top-level classifier to the basic CNN model, and they were trained by applying a 5-fold cross-validation strategy. All three of the proposed models were trained in two stages: transfer learning (which freezes the layers of the basic CNN model and then trains only the top-level classifiers), and fine-tuned learning (which sets the learning rate to a very small number and trains after unfreezing basic CNN layers). SGD, RMSprop, and Adam were applied as optimization algorithms. The experimental results show that the DenseNet CNN model to which the RMSprop algorithm was applied output the best results, with 98.63% accuracy.

Keywords : Convolutional Neural Networks, Deep Learning, Transfer Learning, Fine Tuning, Plant Diseases Classification

*Corresponding Author : Sam-Keun Kim(Hankyong National Univ.)

email: skim@hknu.ac.kr

Received February 1, 2021

Accepted May 7, 2021

Revised March 15, 2021

Published May 31, 2021

1. 서론

토마토 작물은 병해충의 영향을 많이 받아 농업 경제에 막대한 손실을 초래한다[1,2]. 따라서 토마토 병해충의 정확한 식별 및 진단은 높은 작물 수확량을 위해 매우 중요하다. 토마토 병해충을 발견하기 위해 일반적으로 식물 병리학자가 선호하는 육안 접근 방식에서는 결정이 전문가의 경험과 지식에 기반하고 있기 때문에 주관적인 편견이 발생할 수 있다[3]. 한편, 정확한 진단 결과를 얻기 위해 여러 연구들에서 디지털 이미지 처리[3], 컴퓨터 비전[4], 패턴 인식[5]을 기반으로 하여 자동화된 식물 작물 병해충 진단을 시도하였다. 또한 이러한 고급 기술은 오이[6], 석류[7], 옥수수[8], 콩[9]과 같은 다양한 과일 및 작물 종을 분석하는데 사용되었다.

식물 병해충 분류를 위해 다양한 기계학습 알고리즘을 적용하는 최근의 추세는 몇 가지 선택된 병해충과 작물에서 성공적인 결과를 보여주었다[1]. 더구나, 딥러닝 기반의 CNN(Convolutional Neural Network) 아키텍처의 진화는 분류의 정확성을 크게 향상시켰다[10].

딥러닝은 기계학습 응용분야에서 활발하게 활용되는 새로운 추세로 컴퓨터 비전, 바이오 인포매틱스와 같은 많은 분야에서 성공적인 결과를 얻었다[11]. 특히, 컴퓨터 비전 분야에서 많은 딥러닝 기반 접근 방법들이 활발하게 제안되었다[11-13]. [14]는 2개의 딥러닝 기반 CNN인 AlexNet과 VGG16을 이용하여 6가지 종류의 토마토 병해충 및 정상인 경우를 분류하였다. 결과로는 AlexNet이 97.49%, VGG16이 97.23%의 정확도를 얻었다. [15]는 사전 학습된 VGG19 CNN을 3,242개의 이미지(병해충/정상)에 적용하여 98.6%의 정확도를 얻었다. [16]은 정상인 경우를 포함한 10가지 종류의 토마토 병해충을 인식하기 위한 5 종류의 CNN 기반 시스템을 제안하였다. 제안 모델 중에서도 Densenet121_Xception 모델이 97.10%의 가장 우수한 정확성을 보여 주었다.

본 논문에서는 9가지 토마토 병해충 및 정상인 경우를 분류하기 위한 접근 방법으로 CNN 기반의 전이 학습을 도입한다. 즉, 대규모 데이터 상에서 사전 학습된 3가지 딥러닝 모델들인, ResNet, Xception, DenseNet에 튜레벨 분류기를 추가하여 전이 학습 모델을 구현한다. 제안 모델은 98.63%의 정확도로 토마토 병해충을 식별할 수 있으며, 이는 식물 병리학자가 병해충을 진단하는데 도움을 줄 수 있다.

본 논문의 구성은 다음과 같다: 2장에서는 데이터 셋에 대해 기술하고, 3장에서는 심층 CNN 모델을 제시하

고, 전이 학습과 미세 조정에 대해 기술한다. 4장에서는 실험 결과를 제시하고 분석한다. 마지막 5장에서는 결론을 기술한다.

2. 데이터 셋

본 논문에서는 PlantVillage 데이터 셋[17]으로부터 얻은 토마토 작물의 9가지 병해충 및 정상적인 샘플 이미지 데이터 셋을 분석한다. 실험에 사용된 토마토 작물의 잎 이미지는 16,011개이며, 각 이미지는 할당된 레이블(병해충 이름과 정상)을 갖는다. 즉, 각 클래스 레이블은 토마토 병해충 이름 및 정상인 경우로 구분된다. 여기서의 목적은 일단 식물 잎의 이미지가 주어지면 해당 병해충 이름 또는 정상인 클래스로 분류하는 것이다. Fig. 1은 PlantVillage 데이터 셋으로부터 얻은 토마토 병해충 데이터 셋에서 무작위로 뽑은 몇 가지 샘플 이미지를 보여준다. PlantVillage 데이터 셋에서의 이미지 크기는 256x256x3이고, 본 논문에서는 이를 128x128x3으로 조정하여 적용하였다.

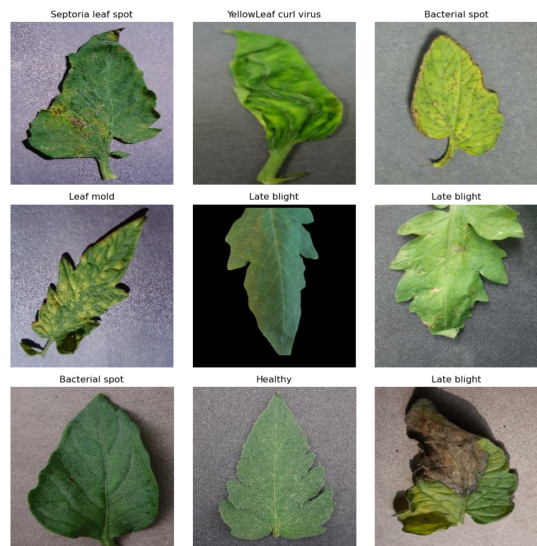


Fig. 1. Random samples output of testing CNN models

전체 이미지 중 80%는 학습에 사용하고, 나머지 20%는 테스트용으로 남겨둔다. Fig. 2는 토마토 병해충 타입의 빈도 히스토그램을 보여준다. Table 1은 Fig. 2의 전체 데이터를 훈련 데이터 셋과 테스트 데이터 셋으로 나눈 각 병해충 데이터의 개수를 보여준다. 학습에 사용된

이미지 개수는 12,808개이고 5-fold 교차검증(cross-validation: CV) 방식으로 학습시켰다. 테스트용으로는 3,203개의 이미지가 사용되었다.

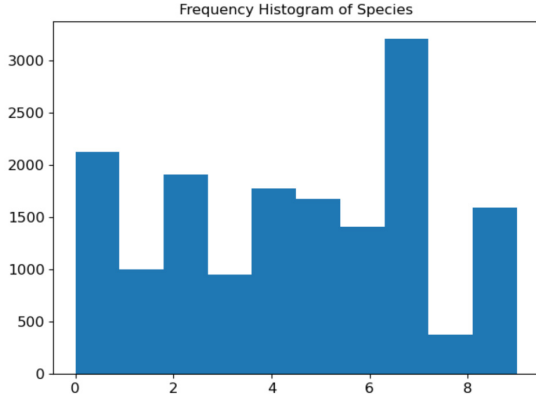


Fig. 2. Frequency histogram of diseases type

Table 1. Training, validation, and test data set of CNN models

Sr. no.	Disease class	Entire set	Train set	Val set	Test set
1	Bacterial spot	2,127	1,351	337	439
2	Early blight	1,000	632	165	203
3	Late blight	1,909	1,221	303	385
4	Leaf mold	952	618	147	187
5	Septoria leaf spot	1,771	1,136	291	344
6	Tetranychus urticae	1,676	1,061	272	343
7	Target spot	1,404	904	233	267
8	Yellow leaf curl virus	3,208	2,056	514	638
9	Mosaic virus	373	246	57	70
10	Healthy	1,591	1,021	243	327
Total	—	16,011	10,246	2,562	3,203

3. 심층 CNN 모델

본 논문에서는 PlantVillage 데이터 셋으로부터 얻은 토마토 병해충 데이터 셋을 분류하기 위해 전이 학습을 이용하여 사전 학습된 CNN을 다시 학습시키는 방법을 적용한다. 대부분의 사전 학습된 CNN은 ILSVRC18에서 사용되는 ImageNet 데이터 셋[19]의 데이터로 학습되었으며 이미지를 1,000가지 사물 범주로 분류할 수 있다. 이러한 다양한 사전 학습된 CNN 모델 중에서

ResNet, Xception, DenseNet 모델을 기본 모델로 하여 전이 학습 및 미세 조정을 통해 Table 1에서 제시한 토마토 병해충 데이터 셋을 분류한다. Fig. 3은 제안한 사전 학습된 ResNet, Xception, DenseNet의 일반 CNN 모델 구조를 보여준다.

모델 구축 단계는 다음과 같다. 먼저 특징 추출을 위해 ImageNet에 대해 사전 학습된 ResNet, Xception, 또는 DenseNet의 마지막 상위 분류 계층을 제외한 기본 CNN 모델 네트워크를 둔다. 기본 CNN 모델은 컨볼루션과 풀링 계층들로 이루어져 있다. 컨볼루션 계층은 이미지에 필터링 기법을 적용한다. 필터링은 행렬의 형태로 표현된 이미지에 대해 행렬로 표현된 필터를 동일하게 적용함으로써 수행된다. 일반적으로 행렬로 표현된 필터링된 이미지의 i 번째 행, j 번째 열의 픽셀 P_{ij} 는 아래 식 (1)과 같이 원본 이미지 X 와 F 의 합성곱(convolution)으로 계산된다:

$$P_{ij} = (F \times X)(i, j) = \sum_{m=0}^{F_H-1} \sum_{n=0}^{F_W-1} F_{m,n} X_{(i-m), (j-n)} \quad (1)$$

여기서 F_H 와 F_W 는 각각 필터의 높이와 너비이다. CNN의 각 컨볼루션 계층에는 입력되는 이미지의 채널 개수 만큼 필터가 존재하며, 각 채널에 할당된 필터를 적용함으로써 컨볼루션 계층의 출력 이미지가 생성된다. 일반적으로 컨볼루션 계층의 출력은 아래 식 (2)처럼 표현될 수 있다:

$$C_j^p = f\left(\sum_{i \in M_j} M_i^{p-1} \times k_{ij}^p + N_j^p\right) \quad (2)$$

여기서 p 는 p 번째 계층, k_{ij} 는 컨볼루션 커널, N_j 는 바이어스, M_j 는 입력 맵을 의미한다. 풀링 계층에는 CNN에서 컨볼루션 계층과 ReLU와 같은 비선형 활성화 함수를 거쳐서 생성된 이미지를 축소 샘플링함으로써 출력을 단순화시켜 준다. 즉, CNN이 학습해야 하는 파라미터의 수를 크게 줄여준다.

다음 단계는 앞 단계에서 생성된 기본 CNN 모델 상위 계층에 톱-레벨 분류기를 추가한다. 톱-레벨 분류기의 첫 번째 계층으로는 기본 CNN 모델로부터 출력되는 이미지의 크기를 $224 \times 224 \times 3$ (또는 $299 \times 299 \times 3$)에서 $4 \times 4 \times 2048$ (또는 $4 \times 4 \times 1920$)로 변환시켜 주는 글로벌 평균 풀링(GAP) 계층을 둔다. 이 계층은 특징들의 블록으로부터 예측을 생성하기 위해 각 이미지의 4×4 공간 위치로부터 평균값을 구해 2048(또는 1920) 벡터로 변환한다. 다음은 이러한 특징들을 이미지마다 하나의 예

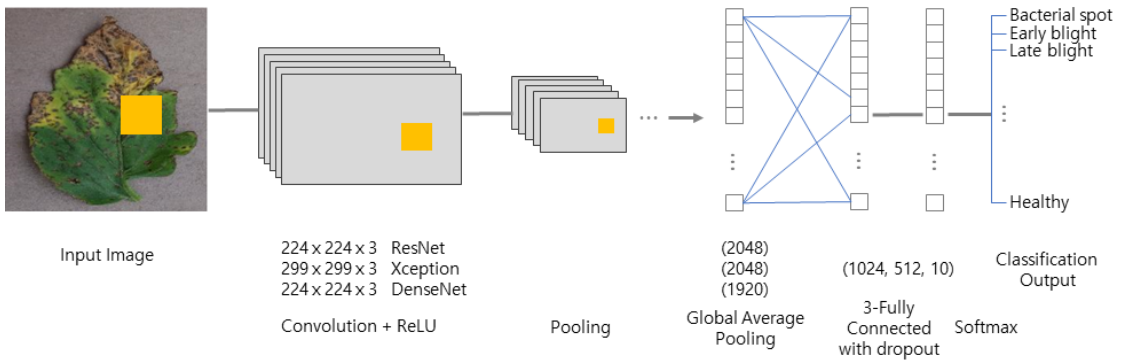


Fig. 3. Proposed ResNet, Xception, and DenseNet CNN general architecture

측으로 변환하기 위해 ReLU 활성화함수를 가진 Dense 계층을 적용한다. 마지막으로, 데이터 증강, 기본 CNN 모델, 특징 추출기를 체인으로 엮음으로써 모델 구축을 완성한다.

제안한 CNN 모델은 네트워크 학습을 두 단계로 나누어서 진행한다(Fig. 4): 전이 학습(Phase 1: transfer-learning)과 미세 조정(Phase 2: fine-tuning). 전이 학습은 딥러닝 응용 분야에서 널리 사용되는 학습 방법으로 사전 학습된 CNN을 새로운 작업을 학습하기 위한 출발점으로 사용한다. 일반적으로 전이 학습으로 CNN을 학습시키는 방법이 처음부터 무작위로 초기화된 가중치를 사용하여 CNN을 학습시키는 것보다 훨씬 쉽고 빠르다. 즉, 전이 학습은 특징을 보다 적은 개수의 훈련 이미지를 사용하여 새로운 데이터 셋에 빠르게 전이시킬 수 있다.

전이 학습으로 학습된 CNN을 미세 조정하면 훨씬 더 높은 정확도를 얻을 수 있다. CNN을 미세 조정하는 것은 간단한 특징 추출에 비해 상대적으로 속도가 느리고

더 많은 노력이 필요하지만, CNN에서 여러 다른 특징이 추출되도록 학습시킬 수 있으므로 최종으로 얻은 CNN의 정확도가 대부분의 경우 더 높다. Fig. 4는 원시 입력 이미지에서 각 병해충의 예측된 출력 확률을 사용하여 ResNet, Xception, DenseNet 모델을 재학습시키는 프로세스를 보여준다. 네트워크의 입력 이미지는 128 × 128 픽셀로 조정된다. 네트워크의 출력 결과는 각 병해충의 확률을 의미한다. 본 논문에서는 Table 1에서의 데이터 셋으로부터 이미지 분류 모델을 개발하기 위해 심층 CNN 모델을 재학습시키는 방법을 제안한다.

4. 실험 결과

본 논문에서는 제안한 CNN 모델을 학습시키기 위한 방법으로 토마토의 9가지 병해충 및 정상 이미지를 분류하기 위해 사전 학습된 심층 CNN을 이용한다. CNN 모델은 ImageNet 데이터 셋으로부터 1,000가지 종류의

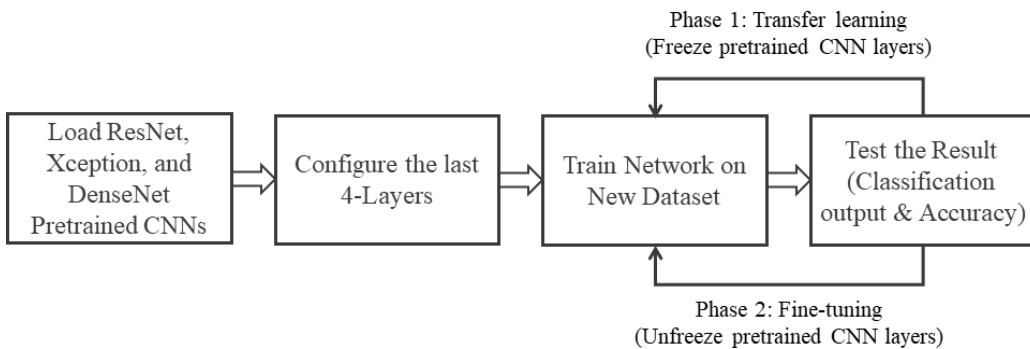


Fig. 4. Retraining process of ResNet, Xception, and DenseNet CNNs model

사물을 분류하도록 사전에 학습되어 있는 상태이다. 따라서 기본 CNN 모델의 마지막 상위 계층을 제거하여 새로운 데이터 셋의 클래스 개수와 동일하게 맞춰야 한다. 본 논문에서는 9가지의 토마토 병해충 종류와 정상인 경우의 10가지 클래스로 분류하도록 만든다. 한편, 상단이 없는 기본 CNN 모델은 마지막 컨볼루션 또는 풀링 계층에서 직접 활성 값을 출력한다. 본 논문에서는 이러한 활성 값을 요약하기 위해 GAP 계층을 기본 CNN 모델 바로 위에 추가하였다. 그 결과는 입력에 대한 기능 설명자로 사용할 수 있는 벡터가 된다. 또한 GAP 계층 위에는 2개의 ReLU 활성화 함수를 가진 FC(fully connected) 계층을 추가하고, 마지막 최상위 FC 계층은 Softmax 계층으로 추가하였다. 또한 FC 계층 학습 시 발생할 수 있는 과적합을 방지하기 위해 BatchNormalization 계층과 Dropout 계층을 FC 계층 사이에 추가하였다. 모델의 초기 학습률은 0.002로 설정하였다. 또한 일정 시간 학습이 개선되지 않을 경우 Keras의 콜백함수인 ReduceLROnPlateau 함수를 적용하여 학습률을 자동으로 조정함으로써 국부적 최적해에서 빠져나올 기회를 개선시켰다. 각 모델의 가중치 갱신을 위해 3가지 최적화 함수를 적용하였다: SGD, RMSprop, Adam(Table 2). Loss 함수는 'categorical_crossentropy'를 적용하였고, 미니배치 크기는 64로 설정하였다. Epoch 수는 단계 1의 전이 학습 단계에서는 30으로 설정하였고, 단계 2의 미세 조정 단계에서는 20으로 설정하였다.

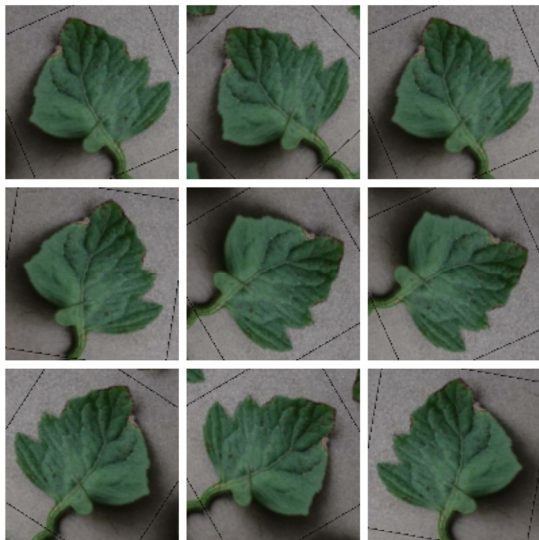


Fig. 5. Augmented sample images for performance

4.1 데이터 증강

본 논문에서는 기본 CNN 모델에 사용된 데이터 셋에 비해 새로 도입하는 토마토 병해충 데이터 셋이 충분하지 않다. 그러므로 훈련 이미지 데이터 셋의 이미지를 무작위로 수평 뒤집기와 회전시키는 방식으로 데이터를 증강시킴으로써 훈련 데이터 셋에 인위적으로 다양성을 도입하였다. Fig. 5는 증강된 이미지 샘플을 보여준다.

4.2 모델 평가 방법

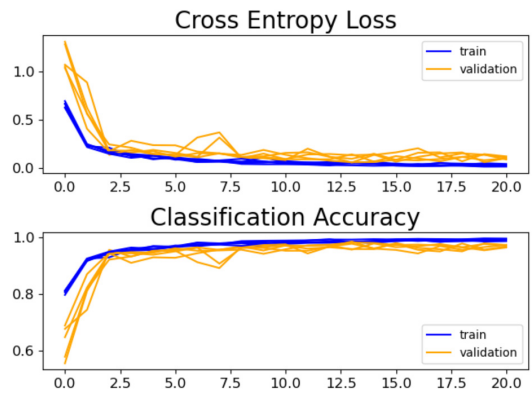


Fig. 6. Loss and Accuracy Learning Curves for the Model Fine-tuning on the PlantVillage(Tomato) Dataset Evaluated Using k-Fold Cross-Validation.

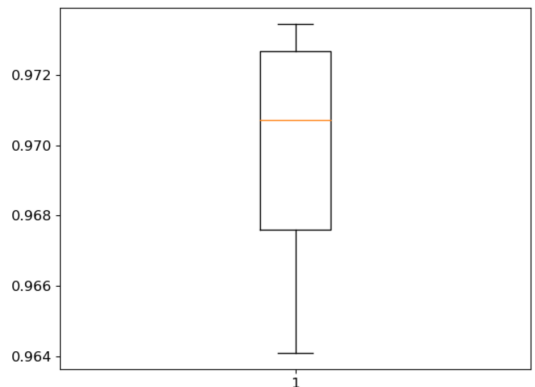


Fig. 7. Box and Whisker Plot of Accuracy Scores for the Model Fine-tuning on the PlantVillage (Tomato) Dataset Evaluated Using k-Fold Cross- Validation.

본 논문에서는 제안한 모델의 성능을 평가하기 위해 먼저 훈련 데이터 셋을 훈련 셋과 검증 셋으로 나누고 5-fold 교차검증 방식을 학습에 적용하였다. 이 방법은 모델이 과적합 하는지 등을 알아보는 데 유용하다. Fig. 6은 훈련 셋과 검증 셋에 대해 제안한 CNN 모델을 미세 조정할 때의 손실과 정확도 학습곡선을 도식화한 것이다. 학습곡선의 결과는 적절히 학습되었음을 보여준다. Fig. 7은 정확도 점수의 분포를 요약해 주는 box-and-whisker 도표이다.

Fig. 8은 모델이 2단계로 학습된 결과를 보여준다. 단계 1은 전이 학습 단계로 기본 CNN 모델의 전체 계층을 동결(freezing)시킨 후 톱-레벨 분류기만을 학습시킨다. 즉, 이 단계에서의 학습은 기본 CNN 모델 계층의 가중치 갱신이 일어나지 않는다. 단지 톱-레벨 분류기의 계층에서만 학습이 이루어진다. 단계 2에서는 미세 조정 단계로 단계 1에서 동결시켰던 기본 CNN 모델의 전 계층에서도 학습이 이루어질 수 있도록 동결을 해제하고 학습률을 매우 작은 수로 설정하여 학습한다.

Fig. 9는 제안한 DenseNet CNN 모델을 사용하여 3,203개의 테스트 셋 이미지를 해당 클래스 카테고리별로 분류한 결과를 예측과 실제 클래스의 CM(confusion matrix)으로 보여준다. Fig. 9(a)는 예측과 실제 클래스의 카운트이고, Fig. 9(b)는 정확도 백분율이다.

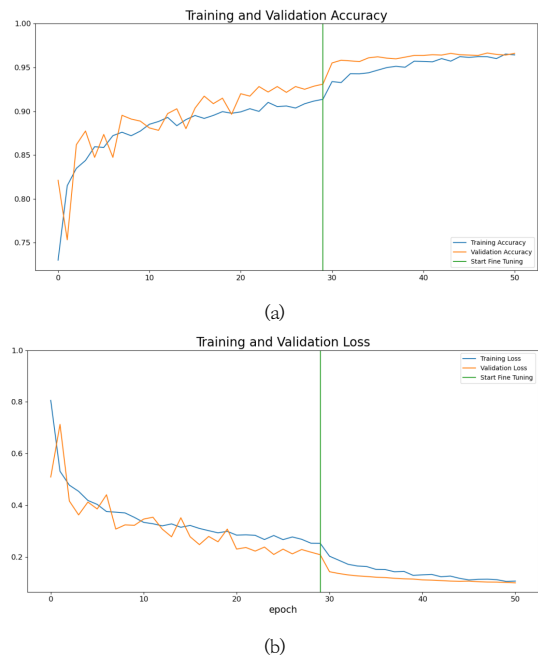
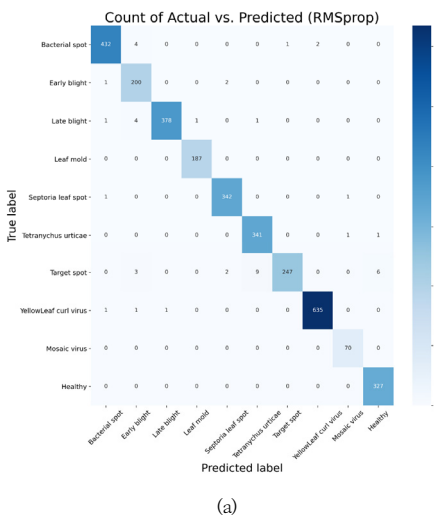


Fig. 8. Plot of retraining progress for tomato diseases classification using proposed DenseNet CNN model (a) Model loss (b) Model accuracy

4.3 제안한 CNN 모델 비교 분석

Table 2는 제안한 CNN 모델들의 성능을 최상의 결과 값, 평균과 t-student 95% 신뢰구간($\bar{x} \pm 1.96\hat{\sigma}_{\bar{x}}$) 방식으로 보여준다[20]. 벤치마킹의 목적을 위해 기계학습

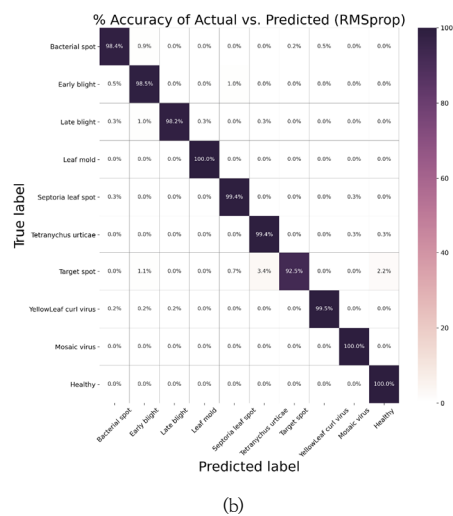


Fig. 9. Confusion matrix of DenseNet CNN model (a) Count of Actual vs. Predicted (b) Percent Accuracy of Actual vs. Predicted

Table 2. Analysis of classification accuracy from 64 mini-batch size during k-fold cross-validation (best $(x \pm 1.96\hat{\sigma}_x)$ values in parentheses; underline- best model; **bold** - best within accuracy & F1 score)

CNN Models	Optimizer	Model performance (%) (train set)	Accuracy (%) (test set)	F1 score (%) (test set)		
				macro avg	weighted avg	support
ResNet	SGD	69.13(65.72±10.08)	67.72	60.47	65.86	3,203
	RMSprop	98.40(97.35±2.51)	98.47	98.35	98.47	3,203
	Adam	97.54(96.49±2.26)	98.19	97.77	98.18	3,203
Xception	SGD	51.31(49.31±2.26)	52.76	42.28	50.86	3,203
	RMSprop	98.28(96.90±1.86)	97.81	97.56	97.82	3,203
	Adam	96.80(95.98±1.44)	96.66	95.88	96.66	3,203
DenseNet	SGD	78.52(75.70±3.37)	75.02	70.19	74.34	3,203
	RMSprop	99.18 (97.82±2.28)	98.63	98.43	98.62	3,203
	Adam	98.21(97.63±1.03)	98.60	98.03	98.60	3,203

SGD 알고리즘의 예측결과(각 모델의 첫 번째 줄)를 추가 하였다. 훈련 셋에 대해 RMSprop 알고리즘을 적용한 DenseNet 모델이 99.18%의 정확도로 가장 좋은 학습 결과를 보였고, 3,203개의 테스트 셋에 대해서도 DenseNet 모델 성능이 98.63%의 정확도로 가장 우수 했다. 마찬가지로 DenseNet 모델의 매크로 평균의 F1 점수는 98.43%이고, 가중치 평균의 F1 점수는 98.62%로 가장 우수했다.

5. 결론

본 논문에서는 PlantVillage 데이터 셋에서 토마토 데이터 셋을 얻어서 딥러닝 기반 CNN 모델에 톱-레벨 분류기를 추가하여 토마토의 9가지 병해충 및 정상인 경우의 클래스로 분류하는 시스템을 제안하였다. ResNet, Xception, DenseNet에 적용한 최적화 함수 중에서 3 가지 모델 모두 RMSprop 함수가 가장 좋은 정확도를 보였으며, DenseNet 모델의 분류 정확도가 98.62%로 가장 높았다.

향후 연구과제로는 실시간 모바일 이미지를 이용하여 토마토 병해충 있을 발견하여 병해충 종류를 인식하는 시스템을 개발하는 것이다.

References

[1] Dheeb Al Bashish, Malik Braik and Sulieman Bani-Ahmad, "Detection and classification of leaf diseases using K-means-based segmentation and

neural networks based classification," Inform Technol J 10, pp.267-275, 2011.

DOI: <https://doi.org/10.3923/iti.2011.267.275>

[2] Jayme Garcia Arnal Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," Biosystems Engineering, Vol. 144, pp. 52-60, 2016.
DOI: <https://doi.org/10.1016/j.biosystemseng.2016.01.017>.

[3] Alireza Khoshroo, Arman Arefi, and Jalal Khodaei, "Detection of Red Tomato on Plants using Image Processing Techniques," Agricultural Communications, 2, pp.9-15, 2014.
https://www.researchgate.net/publication/267624601_Detection_of_Red_Tomato_on_Plants_using_Image_Processing_Techniques

[4] David Ireri, Eisa Belal, Cedric Okinda, Nelson Makange, and Changying Ji, "A computer vision system for defect discrimination and grading in tomatoes using machine learning and image processing," Artificial Intelligence in Agriculture, Volume 2, pp.28-37, 2019.
DOI: <https://doi.org/10.1016/j.aiaa.2019.06.001>

[5] B. Luna-Benoso, J. C. Martinez-Perales, and J. Cortes-Galicia, "Tomato disease detection by means of pattern recognition," International Journal of Computing and Optimization, Vol. 7, no. 1, pp.35-45, 2020.
DOI: <https://doi.org/10.12988/ijco.2020.9823>

[6] Dong Pixia, Wang Xiangdong, "Recognition of Greenhouse Cucumber Disease Based on Image Processing Technology," Open Journal of Applied Sciences, 3, pp.27-31, 2013.
DOI: <https://doi.org/10.4236/ojapps.2013.31B006>

[7] Tejal Deshpande, Sharmila Sengupta and K.S.Raghuvanshi, "Grading and Identification of Disease in Pomegranate Leaf and Fruit", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), pp. 4638-4645, 2014.

- <http://www.ijcsit.com/docs/Volume%205/vol5issue03/ijcsit20140503429.pdf>
- [8] Khairnar K, Dagade R, "Disease detection and diagnosis on plant using image processing—a review," *Int J Comput Appl* 108(13), pp. 36–39, 2014.
DOI: <https://doi.org/10.5120/18973-0445>
- [9] Arpan Singh Rajput, Shailja Shukla, and S. S. Thakur, "Soybean Leaf Diseases Detection and Classification Using Recent Image Processing Techniques," *International Journal of Students' Research in Technology & Management* eISSN: 2321-2543, Vol 8, No 3, pp 01-08, 2020.
DOI: <https://doi.org/10.18510/ijstrtm.2020.831>
- [10] Bin Liu, Yun Zhang, Dongjian He, and Yuxiang Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, 10, 2018.
DOI: <https://doi.org/10.3390/sym10010011>
- [11] Mohammed Brahimi, Boukhalfa Kamel, and Abdelouahab Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Applied Artificial Intelligence* 31(4), pp. 299–315, 2017.
DOI: <http://dx.doi.org/10.1080/08839514.2017.1315516>
- [12] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, pp.1-10, 2016.
DOI: <https://doi.org/10.3389/fpls.2016.01419>
- [13] Jose A. Jimenez-Berni, Amanda Ramcharan, and Jayme G. Barbedo, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases," *Frontiers in Plant Science*, vol. 10: 941, 2019.
DOI: <https://doi.org/10.3389/fpls.2019.00941>
- [14] Aravind Krishnaswamy Rangarajan, Raja Purushothaman, and Anirudh Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Computer Science*, Volume 133, pp. 1040-1047, 2018.
DOI: <https://doi.org/10.1016/i.procs.2018.07.070>
- [15] Jiten Khurana, Anurag Sharma, Harshit Singh Chhabra, and Rahul Nijhawan, "An Integrated Deep Learning Framework of Tomato Leaf Disease Detection," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Volume-8, Issue-11S, 2019.
Available at: <https://www.ijitee.org/wp-content/uploads/papers/v8i11S/K101009811S19.pdf>
- [16] H. Hong, J. Lin and F. Huang, "Tomato Disease Detection and Classification by Deep Learning," 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Fuzhou, China, pp. 25-29, 2020.
DOI: <https://doi.org/10.1109/ICBAIE49996.2020.00012>
- [17] PlantVillage Dataset, <https://www.kaggle.com/emmarex/plantdisease>
- [18] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei (* = equal contribution), "ImageNet Large Scale Visual Recognition Challenge", *International Journal of Computer Vision(IJCV)*, Vol 115, Issue 3, pp. 211–252, 2015. <http://www.image-net.org/challenges/LSVRC/>
- [19] ImageNet. <http://www.image-net.org>
- [20] Arthur Flexer, "Statistical evaluation of neural networks experiments: Minimum requirements and current practice", In *Proceedings of the 13th European Meeting on Cybernetics and Systems Research*, volume 2, pp.1005–1008, Vienna, Austria, 1996.
https://researchgate.net/publication/2627930_Statistical_Evaluation_of_Neural_Network_Experiments_Minimum_Requirements_and_Current_Practice

김 삼 근(Sam-Keun Kim)

[중신회원]



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

<관심분야>

인공지능, 데이터마ining, IoT, BI시스템

안 재 근(Jae-Geun Ahn)

[중신회원]



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

<관심분야>

경영정보시스템, 최적화, 데이터베이스, 데이터마ining