

[캡스톤디자인 중간보고서]

■ 연구과제

과 제 명 (작품명)	딥러닝을 이용한 농작물 질병 분류 모델	참여학기	2021년 1학기
----------------	-----------------------	------	-----------

■ 강좌정보

과 목 명	소프트웨어융합캡스톤디자인	학수번호	SWCON401
과제기간	2021년 03월 02일 ~ 2021년 06월 21일	학 점	3

■ 팀구성

팀 명				팀구성 총인원	1명
구 분	성명	학번	소속학과	학년	
대표학생	정환석	2018110660	소프트웨어융합학과	4	
참여학생					

■ 지도교수 확인


지도교수	성 명	이대호	직 급	전임교수
	소속학과	소프트웨어융합학과	지도교수 확인	성명 : (인)

■ 붙임

[첨부1] 과제 중간보고서

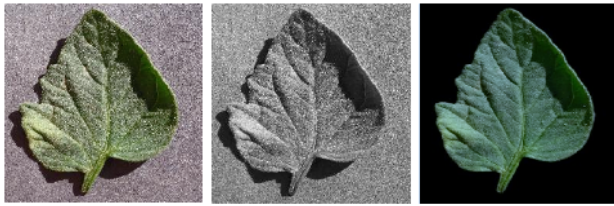
본 팀은 과제를 성실히 수행하고 제반 의무를 이해하여 이에 따른 결과보고서를 제출합니다.

일자 : 2021년 04월 30일

신청자(또는 팀 대표) _____ 정환석 

[캡스톤디자인 과제 중간보고서]

과제명	딥러닝을 이용한 농작물 질병 분류 모델
<p>1. 과제 개요</p> <p>가. 과제 선정 배경 및 필요성</p> <p>한국의 기후 특성상 여름철 긴 장마 후 생육이 불량하고 고온 다습한 환경에서 농작물 병해충으로 인한 피해가 빈번히 발생한다. 병해충에 걸린 농작물은 잎, 줄기, 열매 등에서 증상이 나타나는데 병해충으로 인한 피해를 줄이기 위해서는 감염된 작물의 증상을 통해 초기에 진단하여 신속한 조치를 취해 확산을 막는 것이 가장 중요하다. 하지만 작물별 병해충의 종류는 매우 다양하고 그 진행 양상도 다르므로 인간이 다양한 종류의 작물에 걸린 병을 모두 진단하는 데는 한계가 있다. 따라서 농작물의 증상을 통해 병을 진단하는 연구가 필요하다.</p> <p>나. 과제 주요내용</p> <p>딥러닝 모델 기반 농작물의 잎 데이터를 이용한 질병 분류 모델 구축</p> <p>기존의 이미지 분류에서 높은 성능을 보인 VggNet, GoogLeNet, ResNet 등의 모델의 적용 및 구조적 개선을 통해 분류 모델의 성능을 향상시킨다.</p> <p>다. 최종결과물의 목표</p> <ol style="list-style-type: none">1) 농작물의 질병을 분류하는 딥러닝 모델의 성능을 90% 이상의 정확도를 목표로 한다.2) Data Augmentation 성능을 비교한다.3) 각 모델별 성능을 비교한다.	
<p>2. 과제 수행방법</p> <p>Python, Colab, Pytorch</p> <p>이미지 데이터 수집 → 훈련데이터와 테스트데이터 분할 → 한정적인 데이터를 보완하기 위하여 Data Balancing & Data Augmentation 기법을 이용 → 기존 이미지 분류에서 높은 성능을 보인 CNN Architecture (VggNet, GoogLeNet, ResNet)를 적용 및 구조적 개선 → k-fold Validation → 테스트 데이터셋에 대한 정확도 측정 → 모델별 성능 비교 → 실제 농가의 데이터 수집 및 모델 적용</p>	
<p>3. 진행내용</p> <p>가. 데이터 수집 & 전처리</p> <p>kaggle에서 PlantVillage 데이터셋을 수집하였다. PlantVillage 데이터셋은 질병에 걸린 작물의 잎을 촬영한 이미지로 256x256 크기의 color, grayscale, segmented 데이터로 구성되어있다. 과제에서는 color 이미지를 224x224로 Resize하고 0~255 범위의 픽셀 값을 정규화를 통해 0~1 범위를 가지게 하였다. 전처리된 데이터는 60:20:20으로 train, valid, test 데이터셋으로 나누었다.</p>	



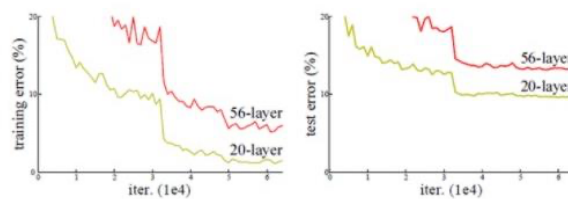
<그림> PlantVillage Data

Class	Healthy & Disease	Count
0	Bacterial_spot	2127
1	Early_blight	1000
2	Healthy	1591
3	Late_blight	1909
4	Leaf_Mold	952
5	Septorial_leaf_spot	1771
6	Spider_mites Two-spotted_spider_mite	1676
7	Target_Spot	1404
8	Tomato_mosaic_virus	5357
9	Tomato_Yellow_Leaf_Curl_Virus	373

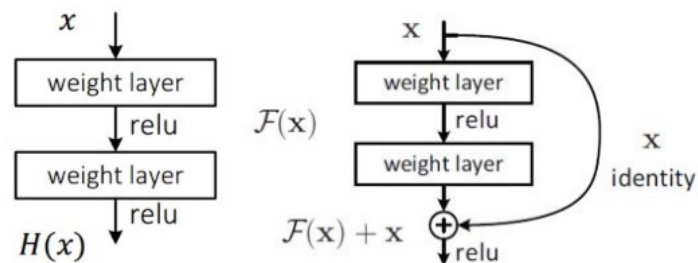
<표> meta data

나. ResNet

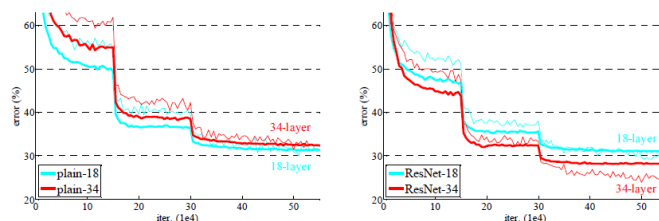
일반적으로 딥러닝에서 신경망을 깊게 설계하는 것만으로도 성능을 올릴 수 있다고 생각할 수 있지만 아래 그림에서 56-layer 보다 20-layer의 신경망의 성능이 더 좋게 나오듯이 무조건 깊게 쌓는다고 성능이 좋아지는 것이 아니다.



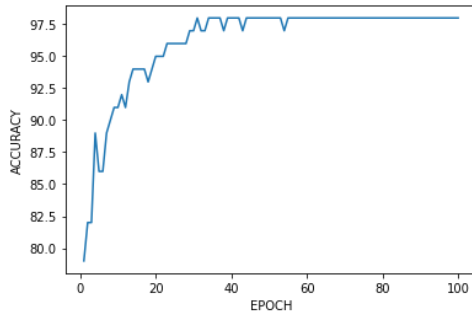
그 이유는 층이 깊어질수록 gradient vanishing 문제가 발생하기 때문이다. 따라서 ResNet 은 이전 Layer 의 Feature Map 을 다음 Layer 의 Feature Map 에 더해주는 Residual Block 을 도입한 모델로 신경망이 깊어지더라도 최소 gradient 로 1 이상의 값을 갖게 하여 gradient vanishing 문제를 해결한 것이다.



아래 Residual Block 이 없는 모델과 있는 모델의 층의 깊이에 따른 성능 차이를 통해서 ResNet 은 신경망의 깊이가 깊어질수록 성능이 좋다는 특징을 찾을 수 있다.



다음은 ResNet 을 이용하여 PlantVillage train 데이터셋을 학습시킨 후 test 한 Epoch 에 따른 정확도 결과이다.



Accuracy of the network on the 3628 test images: 98.24 %

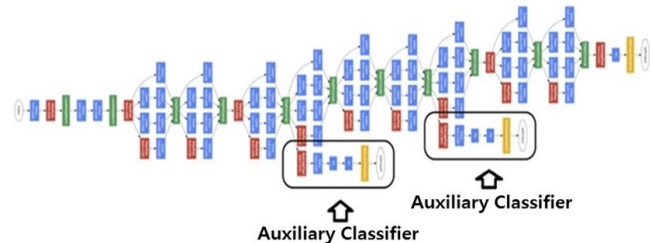
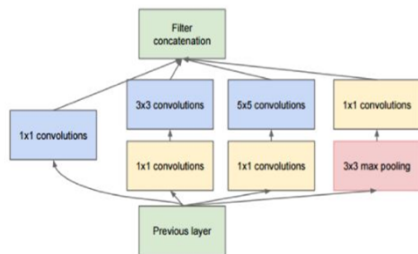
Accuracy of Bacterial_spot : 98 %
 Accuracy of Early_blight : 94 %
 Accuracy of Healthy : 99 %
 Accuracy of Late_blight : 96 %
 Accuracy of Leaf_Mold : 98 %
 Accuracy of Septoria_leaf_spot : 97 %
 Accuracy of Spider_mites Two-spotted_spider_mite : 97 %
 Accuracy of Target_Spot : 95 %
 Accuracy of Tomato_mosaic_virus : 100 %
 Accuracy of Tomato_Yellow_Leaf_Curl_Virus : 99 %

<그림> ResNet - Accuracy

다. GoogLeNet

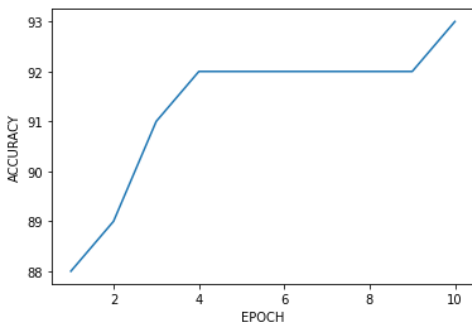
이전까지 CNN 모델에서 컨볼루션 연산을 하면 네트워크가 깊어질수록 학습해야하는 파라미터 수와 연산량이 증가하는 문제점이 있었다.

하지만 GoogLeNet 은 총 22 개의 층을 가지고 Inception module 이라는 block 구조를 통해 연산량을 대폭 줄일 수 있다. Inception module 은 4 가지 서로 다른 연산을 거친 뒤 feature map 을 channel 방향으로 연쇄적으로 합친다. 이 방법을 통해서 연산량을 절반이상으로 줄일 수 있다.



그리고 총 9 개의 inception module 을 쌓는데 3 번째와 6 번째에 보조분류기를 추가하여 총 3 개의 분류기를 사용하였다. 모델이 깊어질수록 마지막에 하나의 분류기만 존재한다면 앞쪽까지 gradient 가 잘 전파되지 않을 수 있는데, 네트워크의 중간부분에 보조분류기를 사용하여 gradient 소멸 문제를 줄일 수 있다고 한다. 보조 분류 기로 구한 loss 는 보조이므로 마지막 분류기 보다 영향력을 적게 주기 위해서 0.3 의 가중치를 주어 total loss 에 더한다.

다음은 GoogLeNet에서 보조분류기를 제거한 상태에서 학습을 진행시킨 후 정확도를 측정한 결과이다.



Accuracy of Bacterial_spot : 50 %
 Accuracy of Early_blight : 49 %
 Accuracy of Healthy : 99 %
 Accuracy of Late_blight : 86 %
 Accuracy of Leaf_Mold : 70 %
 Accuracy of Septoria_leaf_spot : 90 %
 Accuracy of Spider_mites Two-spotted_spider_mite : 44 %
 Accuracy of Target_Spot : 40 %
 Accuracy of Tomato_mosaic_virus : 67 %
 Accuracy of Tomato_Yellow_Leaf_Curl_Virus : 99 %

<그림> GoogLeNet - Accuracy

4. 향후계획

- GoogLeNet에 보조분류기를 추가하고 모델 학습
- 각 모델에 DataAugmentation을 추가하고 모델 학습
- 실제 농가에서 토마토 질병 이미지 데이터 수집
- 모델 성능 평가 기준 정하고 모델 간 성능 비교

팀 학생대표 성명 : 정환석 정환석