

# 전이학습을 활용한 매실 병충해 진단 어플리케이션 개발

정찬혁, 이상철, 서현근, 박동호, 신창선  
순천대학교 정보통신공학과

jmocho@naver.com, lsch9040@naver.com, j6532283@naver.com, ehdgh4875@naver.com  
cssshin@sunchon.ac.kr

## Development of Plum-Diseases Diagnosis Application Using Transfer Learning

Chan-Hyeok Jeong, Sang-Cheol Lee, Hyeon-Keun Seo, Dong-Ho Park,  
Changsun Shin

Dept. of Information and Communication Engineering, Sunchon National  
University

### 요 약

매실의 병충해 이미지를 Tensorflow hub에서 제공하는 Resnet50모델에 Transfer Learning기법을 이용하여 학습시키고, 학습된 모델을 Flask를 이용하여 연동시킨다. 이렇게 완성된 웹앱은 사용자가 매실의 이미지를 업로드 하면, 어떤 병충해를 가지고 있는 지 알려주며, 사용자는 얻은 결과를 통해 육안으로 구분하기 어려운 병충해의 정보를 얻어 매실이 손상이 가는 것을 예방할 수 있다.

### 1. 서론

쌀이 주식인 우리나라는 예로부터 농업이 발달해왔다. 신석기 시대에 발견된 유물을 보고 농업이 언제 시작되었는지 알 수 있고, 가축을 기르고 정착생활을 하게 된 이유도 농업에 있다. 현대 사회에서 농업은 정유, 화학, 반도체, 조선업과 같은 주요 산업에 밀리고 도시화 현상으로 인해 후퇴하는 듯 했으나, 4차 산업의 핵심인 IT기술과 결합된 스마트 농업이 각광받으면서 다시 농업이 주목을 받았다.



(그림 1) 검은별무늬병과 세균성구멍병

농업은 주변 환경의 영향을 많이 받으므로 작물이 손상을 입지 않고 잘 자랄 수 있도록 많은 노력과 시간이 필요하다.

작물이 손상이 가는 원인에는 가뭄, 홍수, 우박과 같은 자연환경의 요인도 있지만,

병충해로 인한 작물의 손상도 있다.

매실을 예로 들면, <그림 1>은 각각 검은별무늬병과 세균성구멍병으로 매실의 대표적인 병해증상이다. 하지만 둘은 발생하는 시기, 증상, 예방법의 차이가 있는 완전히 다른 병해이다.[1]

이 둘은 자칫 혼동하기 쉬워 맞지 않는 예방법을 사용한다면 병해가 완화되지 않고 오히려 악화될 수 있다.

본 논문에서는 앞서 언급한 육안으로 구분하기 어려운 병충해의 정보를 제대로 알고 예방하거나 올바르게 대처할 수 있도록 예측하는 딥러닝 모델을 구현하였다. 또한 사용자가 어디서든 어떤 기기로는 쉽게 접속하여 사용할 수 있도록 웹앱을 만든다.

### 2. 시스템 구조

머신러닝의 Classification은 지도학습의 일종으로 데이터의 관계를 파악하여 같은 Category로 나누는 작업이다.[2] 본 논문에서 구현된 웹앱은 매실 병충해 이미지를 Classification 알고리즘으로 학습시킨 모델과 웹을 연동시킨 구조이다.

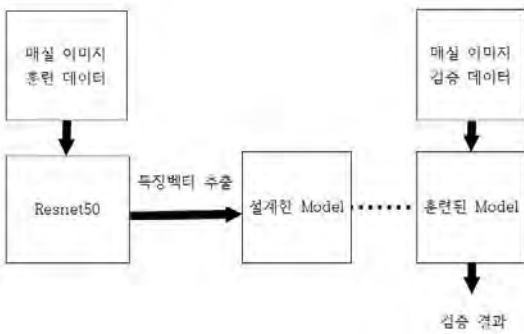
#### 2-1. 전이학습

매실 병충해이미지를 학습시키기 위해서 Python의 라이브러리인 Tensorflow를 사용하였다.

Tensorflow는 구글에서 제공하는 오픈소스 플랫폼으로써 Tensorflow hub를 통해 학습된 모델을 재사용하는 전이학습을 할 수 있다.[3]

전이학습은 미리 훈련된 모델을 재학습 시키거나 데이터의 특징(Feature)을 추출하는 기법이다.[4] 본 논문에서는 Resnet50[5]모델을 이용하여 매실병충해 이미지의 특징을 추출하였다.

### 2-2 학습 및 모델 추출



(그림 2) 학습 과정 도식화

Data Set은 12개의 범주로 분류되어 있으며, 1267장의 매실 병충해 이미지가 있다.

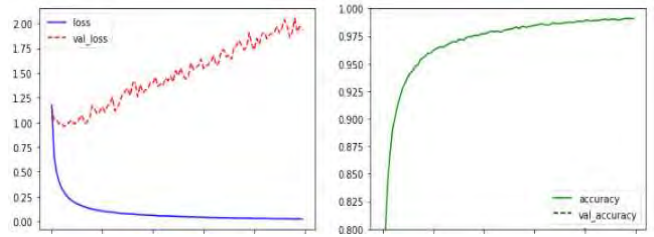
- Aphid
- Arge pagana
- Bacterial shot hole
- Brown Felt
- Canker
- Lichen
- Mite
- Powdery mildew
- Resin
- Scab
- Sclerotial rot
- Sessidae

(그림 3) 12개의 범주

75:25의 비율로 Train Data와 Validation Data로 나눈 뒤, Resnet50모델에 통과시켜, 특징 값을 벡터 형식으로 추출한다. 이 추출한 벡터를 직접 설계한 모델에 통과시켜 50번 학습시켰다. 모델은 Input에 Dense 레이어와 Dropout 레이어, Output에 Dense 레이어로 구성되었고 Optimizer는 Adam, loss function은 Mean Squared Error를 사용하였다.

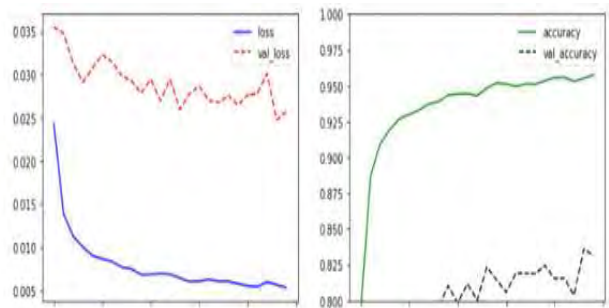
<그림 4>를 보면 검증손실률이 계속해서 증가하는 것을 보이고 이는 모델이 학습과정에서 과적합된 것

을 의미한다.



(그림 4) 과적합 발생

이와 같은 문제가 발생한 이유는 매실 병충해 이미지가 부족한 문제와 이미지에 사람의 손이나 나뭇가지 같은 노이즈가 섞여있기 때문이다



(그림 5) 검증 손실률과 검증 정확도

이미지를 10배 증대하여 다시 학습을 시킨 결과, 검증 손실률은 손실률과 차이는 있지만 감소하는 추이를 나타내며 과적합이 나타나지 않음을 알 수 있었다.

검증 정확도도 다소 차이가 있지만 기존 대비 10% 상승된 약 83%의 정확도를 나타내며 성능 개선이 이루어진 것을 확인했다.

### 2-3. Python Web Framework

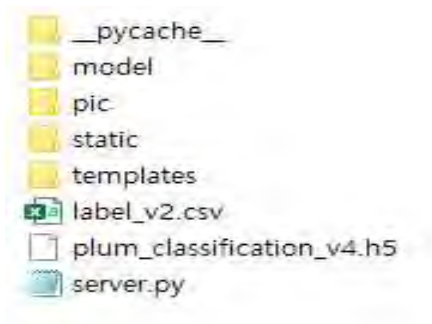
파이썬의 웹 워크프레임 중 대표적인 것은 Flask와 Django인데, 주요 목적은 앞서 학습했던 모델을 추출하여 웹과 연동하는 것이 주목적이기 때문에 상대적으로 가벼운 Flask를 사용하였다.[6]

### 2-4. 웹페이지 구현

웹페이지는 CSS를 이용하여 유지 보수가 용이하도록 구축하였고 업로드, 제출, 분석 버튼을 만들어서 직관성을 확보하였다.



(그림 6) 웹페이지 템플릿



(그림 7) 서버와 모델의 연동

<그림 7>의 server.py 파일은 Flask를 구동하기 위해 사용되는 파일로 사용자가 업로드한 이미지를 추출한 모델에 전달하여 결과값을 웹페이지로 반환해 준다.



(그림 8) 이미지 분석 결과

<그림 8>은 훈련에 사용되지 않은 매실 병충해 이미지를 넣어 앱을 실행시킨 모습이다.

분석 결과값을 출력하는 것으로 서버가 웹과 모델을 제대로 연동하고 있음을 나타낸다.

웹앱은 100번의 성능 테스트결과 87번의 정답을 맞추며 높은 정확도를 나타냈다.

### 3. 결론

딥러닝의 전이학습을 이용하여 모델을 구현하였고 Flask를 이용하여 웹과 연동하였다. 구축된 웹에 현장 매실 이미지를 업로드하여 분석한 결과 매실이 가지고 있는 병충해를 진단하였다.

현재는 업로드된 이미지와 모델이 서버가 구축되어 있는 폴더 안에 저장된다. 그래서 이미지를 관리하기가 쉽지 않고 이미지를 업로드하여 분석을 실행하면 자신이 업로드하지 않은 이미지에 대한 결과가 나타나는 단점이 있다.

이를 해결하기 위해 데이터베이스를 구축하여 데이터의 중복성을 줄이고, 자료를 구조화하여 저장함으로써 자료의 검색과 갱신의 효율을 높일 예정이다. [7]

<그림 8>의 결과 값은 매실 병충해 이름만 보여지고 있지만 결과 창에 병충해 이미지와 증상, 발생원인, 발생 시기, 예방법 등을 보여주는 테이블을 구현하여 사용자가 다양한 정보를 얻을 수 있게 할 예정이다.

또한 웹앱 프레임워크를 사용하여 모바일 친화적인 웹을 구현하여 네트워크만 연결이 되어있다면 어디서든 스마트폰으로 매실 분류기를 사용할 수 있도록 하는 것이 목표이다.

### 참고문헌

- [1] 작목기술정보, 농사로  
[https://www.nongsaro.go.kr/portal/ps/psb/psbk/kidofcomdtyDtl.ps;jsessionid=sqajXa1WVGPLYF4TiZ1K1aJqNOFcsQEqD0RgRUXvJWdoVMtc04JxJC9EvVrfg8jx.nongsaro-web\\_servlet\\_engine1?menuId=PS00067&kidofcomdtyNo=24118](https://www.nongsaro.go.kr/portal/ps/psb/psbk/kidofcomdtyDtl.ps;jsessionid=sqajXa1WVGPLYF4TiZ1K1aJqNOFcsQEqD0RgRUXvJWdoVMtc04JxJC9EvVrfg8jx.nongsaro-web_servlet_engine1?menuId=PS00067&kidofcomdtyNo=24118)
- [2] Classification  
<http://www.incodom.kr/%EA%B8%B0%EA%B3%84%ED%95%99%EC%8A%B5/Classification>
- [3] Tensorflow  
<https://www.tensorflow.org/about?hl=ko>
- [4] Hwan-Hee Kim, 시작하세요! 텐서플로 2.0 프로그래밍, 위키북스, 2020
- [5] Resnet50

<https://keras.io/api/applications/resnet/>

[6] <https://wendys.tistory.com/172>

[7] 데이터베이스

[https://terms.naver.com/entry.nhn?docId=1082446  
&cid=40942&categoryId=32840](https://terms.naver.com/entry.nhn?docId=1082446&cid=40942&categoryId=32840)