**AI기반 농작물 잎사귀 영상을 통한 질병진단**

**1. 프로젝트 개요**

작물 잎을 촬영한 약 40000장의 사진 데이터를 총 33개의 클래스로 분류하였습니다. 성능개선과 클래스간 데이터 수 불균형 문제를 해결하기 위해 pre trained 모델, 오버샘플링을 이용한 결과 여러 가지 성능 측정 방법에서 모두 성능 개선이 있었습니다.

**2. 프로젝트 목적**

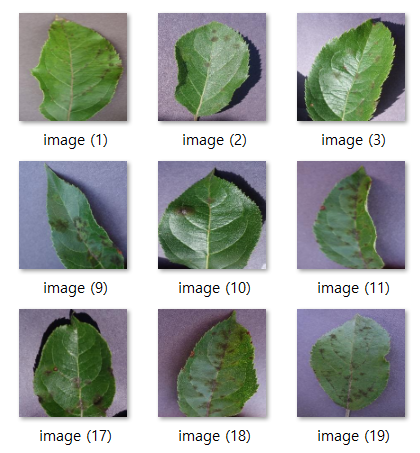
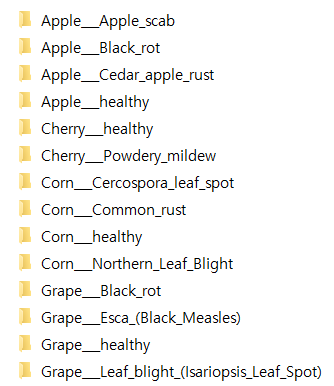
이 분류 모델을 통해서 작물 잎 사진만으로 작물의 질병을 쉽게 진단할 수 있습니다.

작물 잎 사진을 촬영하면 해당 작물의 질병을 진단하는 어플리케이션 개발로 응용될 수 있을 것이라 기대합니다.

**3. 프로젝트 내용**

3-1. Train Validation Test 데이터 분리

- 원본 데이터는 각 분류 클래스에 해당하는 폴더마다 해당 클래스의 이미지 데이터가 train, validation, test가 구별되지 않은 상태로 저장되어 있습니다.

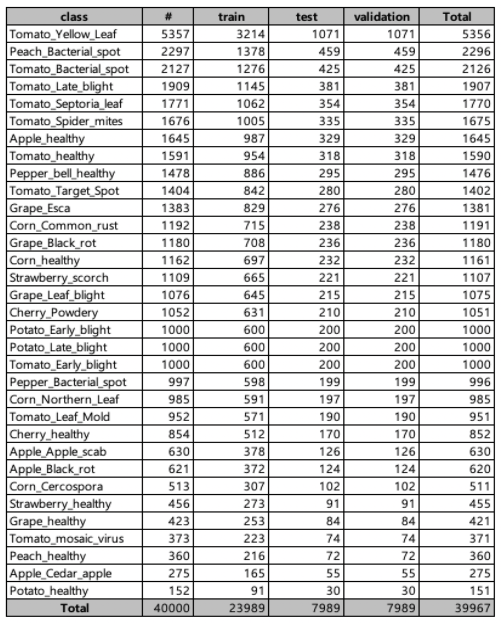


[사진 1] 데이터셋 저장 형태

- 따라서 학습을 시작하기에 앞서 이 데이터들을 train, validation, test 데이터로 나누고, 각각의 클래스에 해당하는 폴더에 저장하는 작업을 시행했습니다.

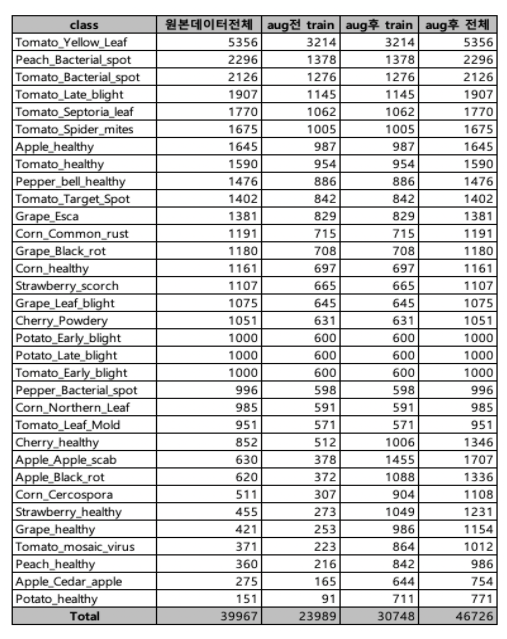
3-2. Oversampling

- Train 데이터를 살펴보면 다음과 같이 클래스별로 데이터의 수 차이가 많이 나는 것을 알 수 있습니다.



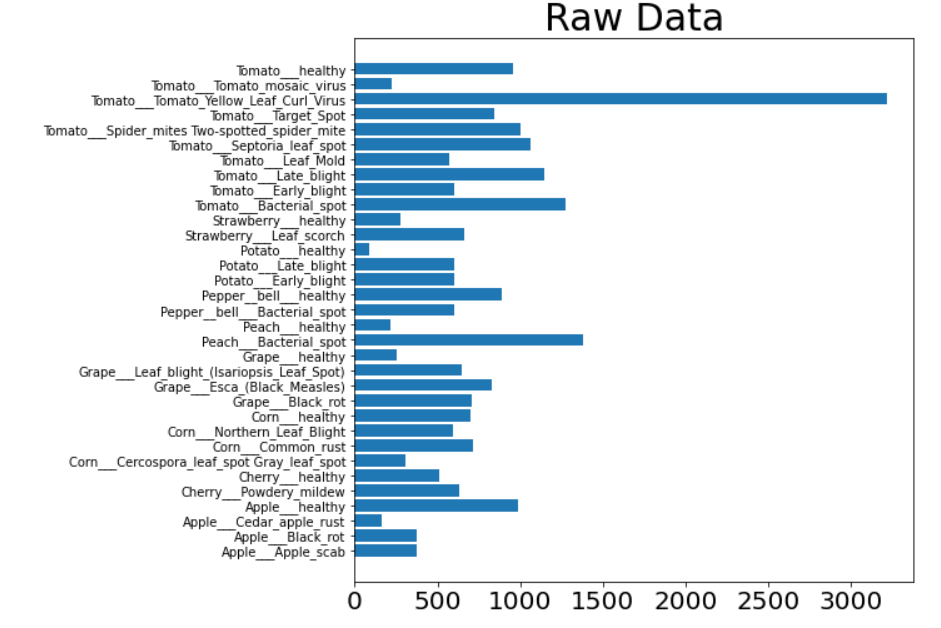
[사진 2] oversampling 이전의 클래스별 데이터 수

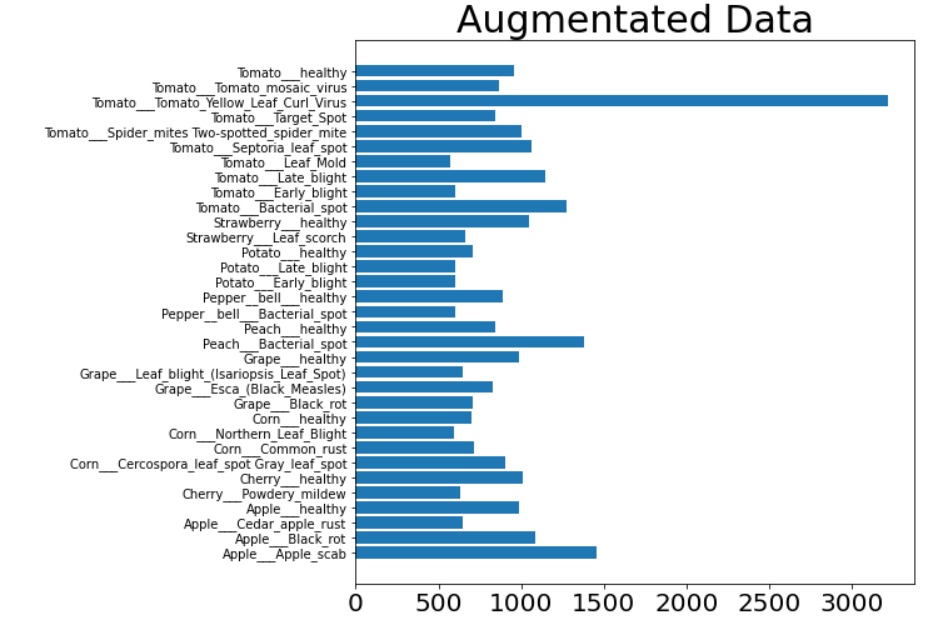
- 이러한 클래스별 데이터 불균형 문제를 해결하기 위해 데이터가 900개 미만인 클래스에 대해 Oversampling을 시행했습니다.



[사진 3] oversampling 이후의 클래스별 데이터 수

- Oversampling으로 인해 클래스 간의 데이터 불균형이 보완된 것을 확인할 수 있습니다.

* 

[사진 4] Oversampling 이전의 클래스별 데이터 분포 시각화

[사진 5] Oversampling 이후의 클래스별 데이터 분포 시각화

3-3. Train

1) Baseline 모델

- Convolutional Neural Network 모델을 구성하였습니다.

- 클래스간의 데이터 수 불균형이 해결되지 않은 raw 데이터를 사용한 모델과 oversampling을 시행한 데이터를 사용한 모델을 각각 구성하였습니다.

2) Pre- trained 모델

- Pre-trained model로 ResNet50 모델을 불러와 fine-tuning을 진행하였습니다. Fine-tuning을 할 때, 모든 레이어의 가중치를 업데이트 가능하도록 설정하였습니다.

- Oversampling을 시행한 데이터를 이용하여 학습하였습니다.

3-4. Evaluation

세 가지 모델을 accuracy, 2종오류, F1-score 측면에서 비교하였습니다.

1) Accuracy

- Baseline + 원본 데이터 : 약 86%

- Baseline + oversampling된 데이터: 약 89%

- ResNet50: 약 97%

2) False Negative

- 질병 분류 문제이기 떄문에 질병이 있는데 없다고 예측한FN(False Negative)의 값, 즉 2종 오류에 해당하는 값 또한 중요한 성능 평가 metric이라 할 수 있습니다.

- 예시로 질병의 종류가 가장 많은 토마토에 대해서 FN 값을 비교해보았다. 각 모델의 Confusion Matrix에서 "Tomato\_\_healthy"라고 예측하였는데 실제 질병이 있는 경우의 개수를 세어본 결과는 다음과 같습니다.

- Baseline + 원본 데이터: 79개

- Baseline + oversampling된 데이터: 24개

- ResNet50: 12개

- 따라서 데이터의 불균형 해소를 한 것이 더 낮은 FN값을 가지고 Baseline보다 Pretrained 모델이 2종 오류를 적게 범함을 알 수 있습니다.

3) F1-score

- 4가지 class의 f1 score를 비교해보았습니다. 4가지 클래스는 다음과 같습니다.

|  |
| --- |
| 데이터가 가장 많은 Tomato yellow leaf(oversampling 미적용 클래스) |
| 데이터가 가장 적었던 Potato healthy(oversampling 적용 클래스) |
| oversampled data기준으로 중간정도의 데이터를 가지고 있는  Apple healthy oversampling 미적용 클래스) |
| oversampled data기준으로 중간정도의 데이터를 가지고 있는  Cherry healthy(oversampling 적용 클래스) |

- 4가지 클래스의 f1 score는 다음과 같습니다.

|  |
| --- |
| Tomato yellow leaf: raw: 0.96, over: 0.97 |
| Potato healthy: raw: 0.63, over: 0.74 |
| Apple healthy: raw: 0.86, over: 0.89 |
| Cherry healthy : raw: 0.89, over: 0.96 |

- Baseline 모델을 보면 oversampling을 적용한 클래스의 f1 score가 증가 폭이 더 큼을 알 수 있습니다. 따라서 데이터가 적은 클래스에 oversampling을 적용한 것이 분류 성능을 높이는 효과를 가져왔음을 알 수 있습니다.

- 데이터 수가 가장 많은 "Tomato yellow leaf"는f1 score도 0.97로 가장 높은 반면, 가장 적은 데이터를 가진 "Potato healthy"의 f1-score는 0.74로 비교적 낮은 값을 가지고 있음을 알 수 있습니다.

**4. 주요 사용기술**

- 언어 : Python

- 모델 : CNN

- 프레임워크 : Keras

기본적으로 Convolutional Neural Network를 사용하였습니다. 성능 향상을 위해 Pre-trained model인 ResNet50을 불러온 후, 모든 레이어의 가중치를 학습 과정에서 업데이트하였습니다. Keras 프레임워크로 구현하였습니다.

**데이터 출처**

원본 데이터 : <https://data.mendeley.com/datasets/tywbtsjrjv/1>

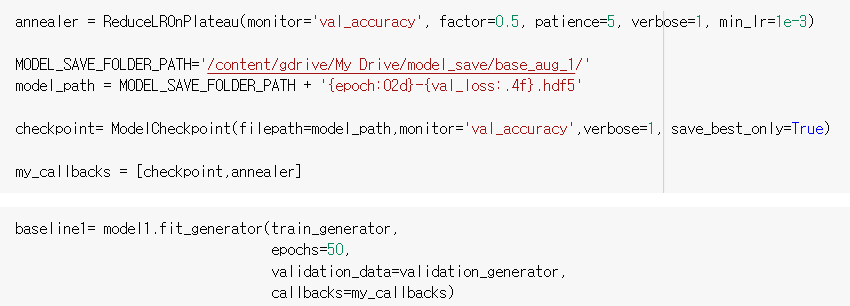
작물 잎 사진을 통해 해당 작물이 질병을 가지고 있는지의 여부를 판단하는 것이 프로젝트의 목표이기에 원본 데이터에서 한 종류의 작물에 대해 ‘질병’에 해당하는 클래스가 존재하지 않고 ‘건강한’에 해당하는 클래스만 존재하는 경우는 제외하였습니다.

**5. 주요 소스코드**

- 베이스라인 모델 구축 코드 (오버샘플링 코드와 구조 동일)



- 베이스라인 모델 학습 코드



- ResNet50 학습 코드

