

Fruit Froot Image Classification

Year-dream school, Bopy CV project

최윤석, 안희수, 허재호

요약

사과 이미지 37024개와 배 이미지 22991개의 이미지 데이터들을 각각의 과수에서 발생하는 질병으로 구분하였다. 배의 질병에는 배검은병무늬병, 배과수화상병이 있고, 사과에는 사과갈색무늬병, 사과과수화상병, 사과부란병, 사과점무늬낙엽병, 사과탄저병이 있다. 배에서는 3개의 다중 클래스들을 MobileNet 모델로 학습을 시켰고, 사과는 6개의 다중 클래스들을 VGG19 모델로 학습을 시켰다. 또한 Flask를 이용한 API 구축에서는 사과와 배, 두개의 선택지를 적용시켜서 이미지를 업로드 할 때 따로 업로드 시킬 수 있도록 적용하였다.

1. 서론

본 학습에서는 사과와 배의 국내 과원들에서 지금까지 발병한 사진들의 데이터들을 이용해 차후 발생하는 사과와 배의 질병 증상들을 발견하였을 때 초기에 질병명을 구분을 하여 맞는 치료법을 제공할 수 있게 하기 위한 자료를 만드는 것을 목표로 하였다.

2. 데이터의 설명

본 데이터는 아래의 주소에서 제공된 데이터를 사용하였다

<https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=146>

설명: 사과와 배의 화상병 및 유사 병종의 이미지 데이터

질병명: 배의 질병에는 배검은병무늬병, 배과수화상병이 있고, 사과에는 사과갈색무늬병, 사과과수화상병, 사과부란병, 사과점무늬낙엽병, 사과탄저병이 있다.

아래 그림 1은 원본데이터에서 제공된 총 이미지들의 개수이다. 이 데이터에서는 증강 이미지들도 포함이 되었기 때문에 원본 데이터의 크기가 총 100기가 넘었고, 데이터의 구분이 정상과 질병 두개의 폴더로만 나뉘어져 있었다.

| 구분 | 과수화상병 | | |
|-----|-------|----------|---------|
| | 작물명 | 질병명 | 구축수량 |
| 1-0 | 배 | 정상 | 25,617 |
| 1-1 | | 배검은별무늬병 | 20,422 |
| 1-2 | | 배과수화상병 | 19,578 |
| 2-0 | 사과 | 정상 | 35,938 |
| 2-3 | | 사과갈색무늬병 | 22,313 |
| 2-4 | | 사과과수화상병 | 21,930 |
| 2-5 | | 사과부란병 | 22,851 |
| 2-6 | | 사과점무늬낙엽병 | 21,454 |
| 2-7 | | 사과탄저병 | 21,452 |
| 합계 | | | 211,555 |

그림 1 원본데이터 질병 별 이미지의 개수

원본 데이터의 개수와 크기가 너무 커서 활용하기가 힘들었기 때문에 우선 증강 데이터들을 다 제거하였고 코드 1의 `resize_img` 코드를 통해 원본데이터를 (224,224)의 사이즈로 줄이는 작업을 하였다

```
import os
import numpy as np
import cv2
from PIL import Image

original_path = './Training/[원천]배_1.질병/'
resized_path = './Training/resized/'

file_list = os.listdir(original_path)
img_list = []

for item in file_list :
    img_list.append(item)

total_image = len(img_list)
index = 0
total_image
for name in img_list :

    img = Image.open('%s%s'%(original_path, name))
    img_array = np.array(img)
    img_resize = cv2.resize(img_array, (224,224), interpolation = cv2.INTER_AREA)
    img = Image.fromarray(img_resize)
    img.save('%s%s'%(resized_path, name))

    print(name + ' ' + str(index) + '/' + str(total_image))
    index = index + 1
print("Conversion completed")
```

코드 1 `resize_img` 코드

이미지 변환후 총 파일들의 크기는 1기가 미만으로 줄어들어서 학습에 용이하게 되었다

3. 데이터 분석

처음에는 배와 사과와의 구분없이 그림 2의 질병으로 구분을 하였고 거기서 정상 배와 정상 사과를 추가해 총 9개의 클래스들로 다중 클래스 이미지 분류를 시켰지만 배의 열매 데이터의 부족으로 학습 결과에서 배와 사과의 구분을 확실히 못하는 모습을 보여주었다.

이 데이터들을 각각 배와 사과의 폴더에 나누었고, 배에서는 정상, 배검은병무늬병, 배과수화상병의 3개의 클래스들로 나누어 다중 클래스 분류를 하였다. 그리고 사과에서는 정상, 사과갈색무늬병, 사과과수화상병, 사과부란병, 사과점무늬낙엽병, 사과탄저병의 6개 클래스들로 나누어 다중 클래스 분류를 하였다.

| 질병 코드 | |
|-------|----------|
| 00 | 정상 |
| 01 | 배검은별무늬병 |
| 02 | 배과수화상병 |
| 03 | 사과갈색무늬병 |
| 04 | 사과과수화상병 |
| 05 | 사과부란병 |
| 06 | 사과점무늬낙엽병 |
| 07 | 사과탄저병 |

그림 2 질병 구분

아래의 그림 3에서 라벨링 된 데이터의 파일명을 통해 각각 질병코드에 맞는 폴더로 분류를 해주었다.

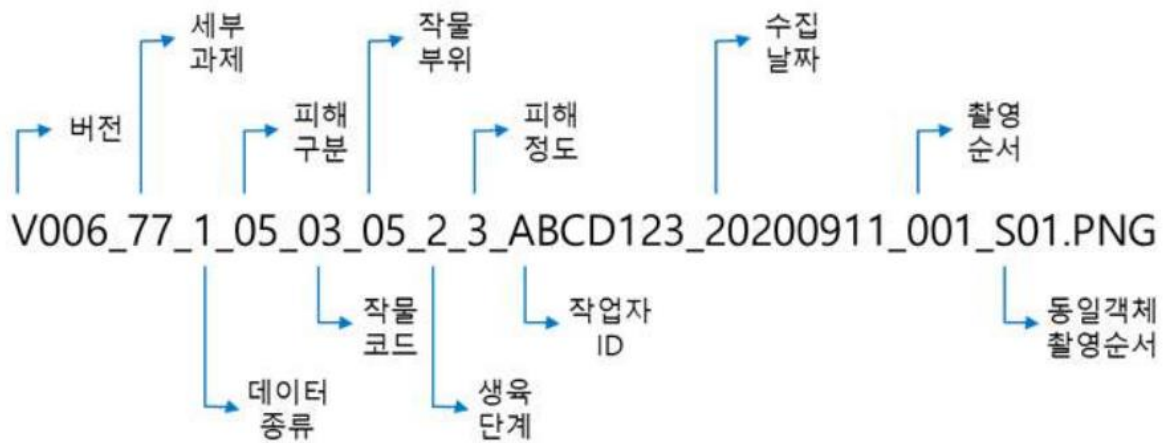


그림 3 파일명

4. 사용한 딥러닝 모델들

팀원들 각자 Resnet50, MobileNet, VGG19 모델들을 나누어 테스트를 해보았고 그 중에서 정확도가 제일 높았던 VGG19모델을 최종적으로 결정했다. 하지만 결과를 출력하는 과정에서 배의 열매의 데이터수가 상대적으로 많이 부족했기에 배와 사과를 구분하지 못하는 모습을 보여주었기에 3개의 배 클래스들을 MobileNet으로 학습시켰고, 6개의 사과 클래스들을 VGG19 모델로 학습시켰다. 학습 결과 아래 표 1과 같은 결과를 도출해 냈다.

| | 배 | 사과 |
|---------------------|-----------|--------|
| | MobileNet | VGG19 |
| Validation Loss | 0.0595 | 0.0922 |
| Validation Accuracy | 0.9774 | 0.9705 |

표 1 딥러닝 모델 학습 결과

5. 예측 결과 출력

```
model = load_model('./Models/Pear_Mobilenet-006-0.0761-0.9739.hdf5')
fruit = ['정상 배', '배검은별무늬병', '배과수화상병']
path = './배/Test/02_배과수화상병/'
for i in range(10):
    FileName=random.choice(os.listdir(path))
    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
    img = copy.deepcopy(PIL.Image.open(path + FileName))
    img = ImageOps.fit(img, (224, 224), Image.ANTIALIAS)
    normalized_image_array = np.asarray(img, dtype = 'float32') / 255.
    plt.imshow(normalized_image_array)
    plt.show()

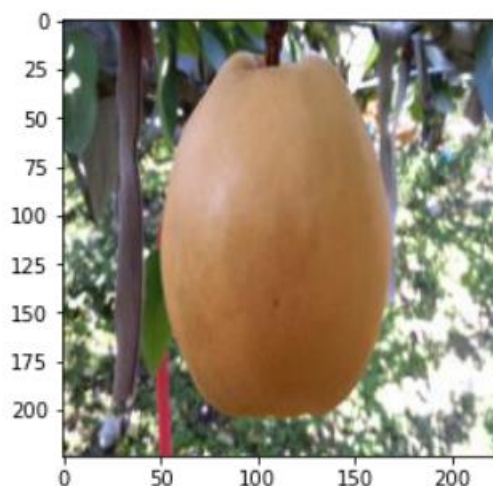
    data[0] = normalized_image_array
    result = (model.predict(data))*100
    result1 = pd.DataFrame(result)
    a= result.tolist()
    a.sort()
    print("파일 이름 : " + FileName)
    for i in range(len(result[0])) :
        if result[0][i] > 40:
            a = result[0][i]
            a = a.astype(str)
            print("업로드한 사진과 유사한 이미지는 " + a + "% 확률로 " + fruit[i] + "입니다.")
            print('\n')
```

코드 2 배 예측 결과 출력 코드

위의 출력 코드를 통해 아래와 같은 결과를 얻었다

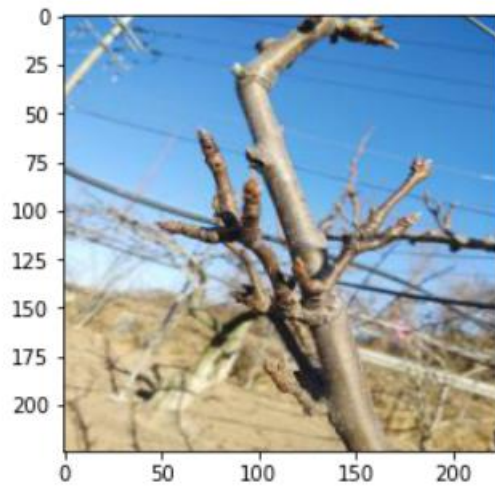
배 예측결과

정상 배를 모델에 넣어서 예측을 시켜봤을 때 아래와 같은 열매와 나무, 그리고 잎을 각자 구분할 수 있었다



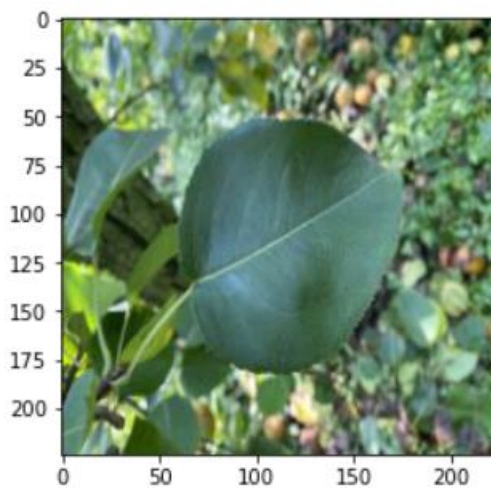
```
1/1 [=====] - 0s 15ms/step
파일 이름 : Y006_80_0_00_01_01_25_0_B07_20201005_0001_S01_1.JPG
업로드한 사진과 유사한 이미지는 94.99603% 확률로 정상 배입니다.
```

그림 4 정상 배의 열매



1/1 [=====] - 0s 17ms/step
파일 이름 : V006_80_0_00_01_05_26_0_C32_20201215_0082_S01_1.JPG
업로드한 사진과 유사한 이미지는 99.99572% 확률로 정상 배입니다.

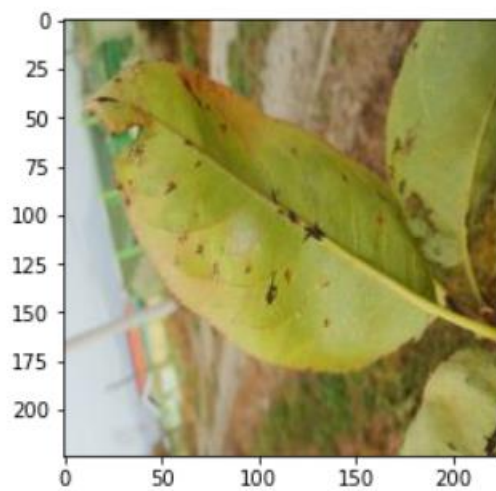
그림 5 정상 배의 나무



1/1 [=====] - 0s 17ms/step
파일 이름 : V006_80_0_00_01_03_25_0_C12_20201022_0072_S01_1.JPG
업로드한 사진과 유사한 이미지는 99.99937% 확률로 정상 배입니다.

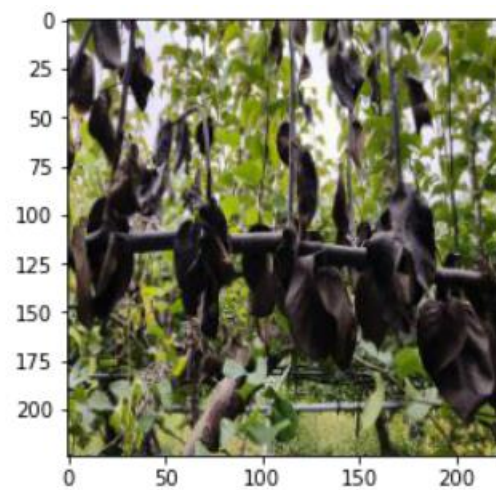
그림 6 정상 배의 나무

질병에 걸린 배를 모델에 넣어 예측을 시켜봤을 때에도 정확한 결과가 나왔다



1/1 [=====] - 0s 16ms/step
 파일 이름 : V006_80_1_01_01_03_23_2_3903y_20201109_13.jpg
 업로드한 사진과 유사한 이미지는 99.99898% 확률로 배검은별무늬병입니다.

그림 7 배검은별무늬병



1/1 [=====] - 0s 18ms/step
 파일 이름 : V006_80_1_02_01_04_23_3_2581b_20201006_9.jpg
 업로드한 사진과 유사한 이미지는 99.99879% 확률로 배과수화상병입니다.

그림 8 배과수화상병

```

model = load_model('./Models/Apple_VGG19-009-0.0922-0.9705.hdf5')
fruit = ['정상 사과', '사과갈색무늬병', '사과과수화상병', '사과-부란병', '사과점무늬낙엽병', '사과탄저병']
path = './Test/00_사과정상/'
for i in range(10):
    FileName=random.choice(os.listdir(path))
    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
    img = copy.deepcopy(PIL.Image.open(path + FileName))
    img = ImageOps.fit(img, (224, 224), Image.ANTIALIAS)
    normalized_image_array = np.asarray(img, dtype = 'float32') / 255.
    plt.imshow(normalized_image_array)
    plt.show()

    data[0] = normalized_image_array
    result = (model.predict(data))*100
    # result1 = pd.DataFrame(result)
    a= result.tolist()
    a.sort()
    print("파일 이름 : " + FileName)
    for i in range(len(result[0])) :
        if result[0][i] > 40:
            a = result[0][i]
            a = a.astype(str)
            print("업로드한 사진과 유사한 이미지는 " + a + "% 확률로 " + fruit[i] + "입니다.")
            print('\n')

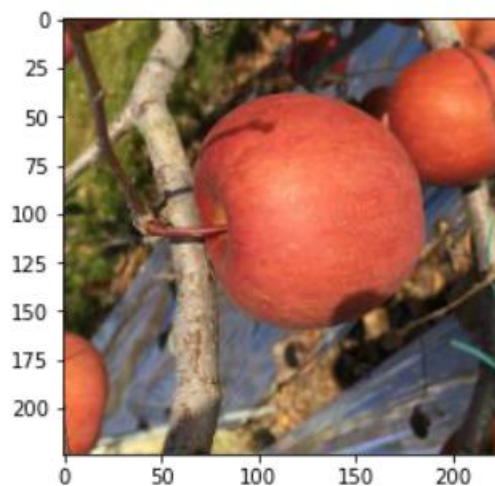
```

코드 3 사과 예측 결과 출력 코드

위의 출력 코드를 통해 아래와 같은 결과를 얻었다

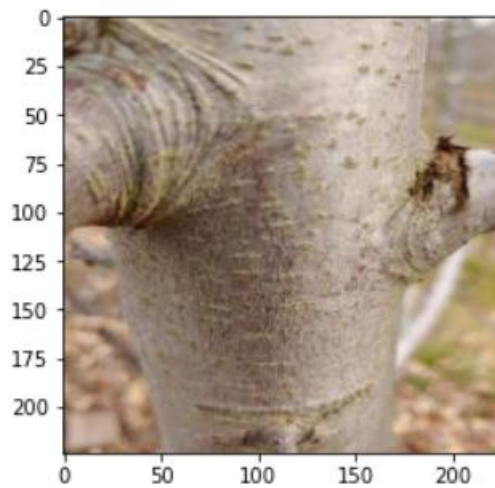
사과 예측 결과

정상 사과를 모델에 넣어서 예측을 시켜봤을 때 아래와 같은 열매와 나무, 그리고 잎을 각자 구분할 수 있었다



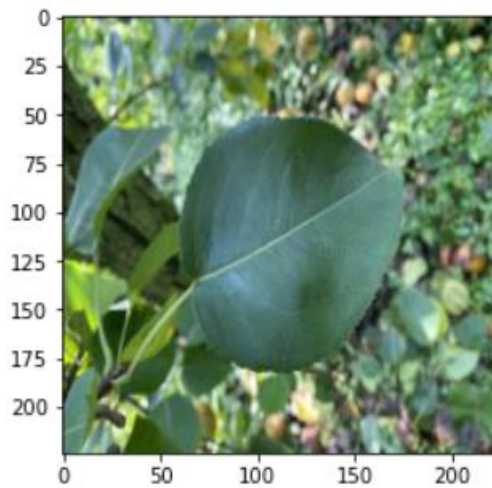
1/1 [=====] - 0s 14ms/step
 파일 이름 : Y006_80_0_00_02_01_25_0_C24_20201028_0002_S01_1.JPG
 업로드한 사진과 유사한 이미지는 100.0% 확률로 정상 사과입니다.

그림 9 정상 사과 열매



1/1 [=====] - 0s 14ms/step
 파일 이름 : V006_80_0_00_02_05_25_0_A01_20201116_0042_S01_2.JPG
 업로드한 사진과 유사한 이미지는 100.0% 확률로 정상 사과입니다.

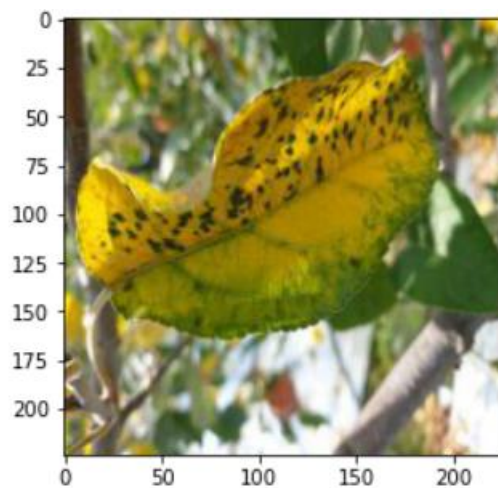
그림 10 정상 사과 나무



1/1 [=====] - 0s 17ms/step
 파일 이름 : V006_80_0_00_01_03_25_0_C12_20201022_0072_S01_1.JPG
 업로드한 사진과 유사한 이미지는 99.99937% 확률로 정상 배입니다.

그림 11 정상 사과 잎

질병에 걸린 사과들을 모델에 넣어 예측을 시켜보았을 때도 정확한 예측 결과가 나왔다

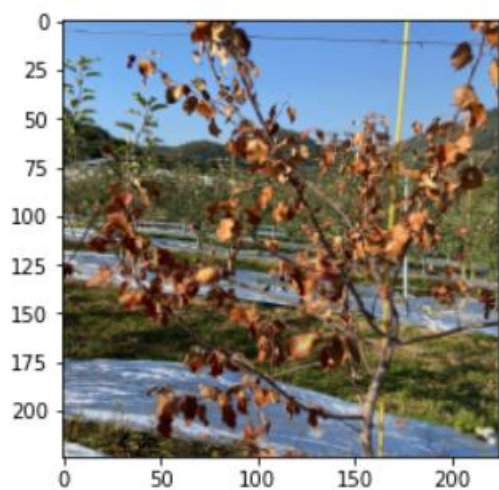


1/1 [=====] - 0s 90ms/step

파일 이름 : Y006_80_1_03_02_03_22_2_0103t_20201015_4.JPG

업로드한 사진과 유사한 이미지는 100.0% 확률로 사과갈색무늬병입니다.

그림 12 사과갈색무늬병

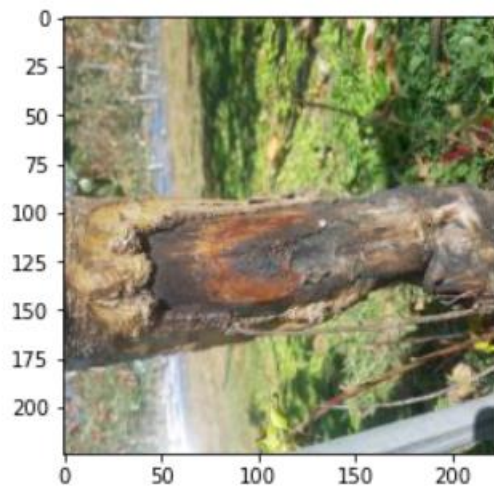


1/1 [=====] - 0s 16ms/step

파일 이름 : Y006_80_1_04_02_03_23_3_2034y_20201016_5.png

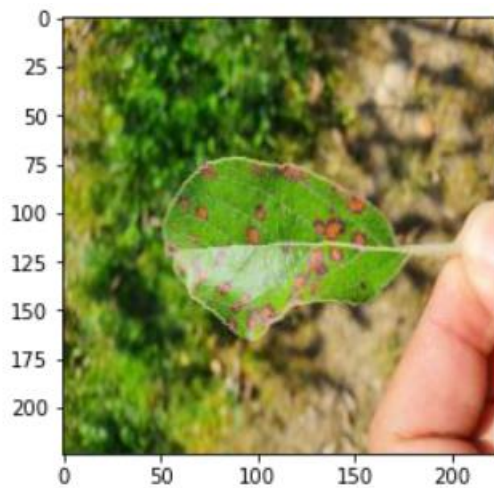
업로드한 사진과 유사한 이미지는 99.59459% 확률로 사과과수화상병입니다.

그림 13 사과과수화상병



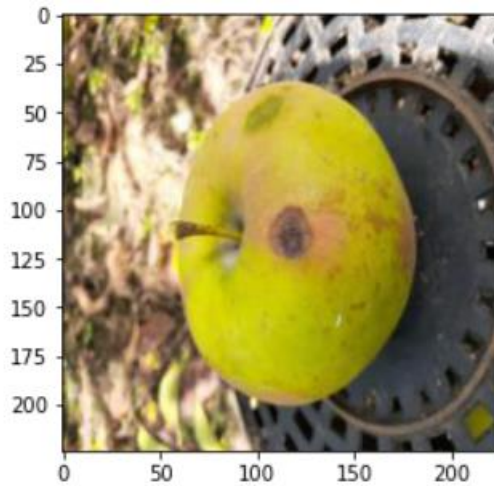
1/1 [=====] - 0s 13ms/step
 파일 이름 : V006_80_1_05_02_04_22_2_3880y_20201015_30.jpg
 업로드한 사진과 유사한 이미지는 99.869576% 확률로 사과부란병입니다.

그림 14 사과부란병



1/1 [=====] - 0s 15ms/step
 파일 이름 : V006_80_1_06_02_03_21_2_3694y_20201019_141.jpg
 업로드한 사진과 유사한 이미지는 99.99634% 확률로 사과점무늬낙엽병입니다.

그림 15 사과점무늬낙엽병



1/1 [=====] - 0s 92ms/step
 파일 이름 : V006_80_1_07_02_01_23_2_4498w_20200916_70.JPG
 업로드한 사진과 유사한 이미지는 99.99067% 확률로 사과탄저병입니다.

그림 16 사과탄저병

6. Flask API 구현

반응형 웹페이지로 구현을 해서 데스크탑과 모바일에 맞게 화면이 자동으로 바뀐다.

홈페이지 화면에서 사과와 배를 선택하는 버튼을 아래의 이미지와 같이 넣어주고, 하나를 선택할 시에 업로드 화면으로 넘겨준다.

이미지가 업로드 되면 그 이미지를 예측해서 결과를 보여주고 자세히 보기 버튼을 추가해서 병의 정보를 보여주는 페이지로 넘겨준다

FruitDoctor



created by @Bopy_CV Team.

그림 17 데스크탑 홈페이지

Apple Disease Classifier

Upload the Apple Picture...



사과과수화상병일 확률이 100.0% 입니다.

자세히보기

그림 18 데스크탑 결과 페이지



FruitDoctor



🍏 Apple Disease Classifier

Upload the Apple Picture...



사과과수화상병일 확률이 100.0% 입니다.

자세히보기

created by @Bopy CV Team.

created by @Bopy CV Team.

그림 19 모바일 결과페이지

그림 20 모바일 홈페이지



과수 화상병

초기에 꽃이 마르고, 잎이 주맥부터 갈게 변하며, 줄기, 과실 등에 세균 누출액이 맺히고, 어린 가지가 갈고리처럼 휘어지는 것이 특징이다.

INFO MORE

배 검은 별 무늬병

처음에는 잎의 양맥을 따라 뚜렷하지 않은 부정형 또는 다형성의 검은 별반이 생기고, 점차 진전되면서 그늘진부분의 모서리가 무수히 발생한다.

INFO MORE



사과 갈색 무늬병

잎에는 처음 자색 또는 흑갈색의 작은 반점이 형성, 점차 확대되어 갈색 또는 흑갈색의 대형 반점이 형성되며, 망판 출애가 녹색으로 남고, 다른 부위는 황색으로 변하여 조기 낙엽된다.

INFO MORE



사과 부란병

처음에는 수피가 갈색으로 변색되어 부패에 이르러 쉽게 벗겨지며, 밑부분에서 시작된다. 병충해가 진전되면 수분을 상실, 종횡파괴 그 표면에 출애의 작은 점이 형성된다. 나무껍질이 갈색으로 되며 약간 부풀어 오르고 쉽게 벗겨지고 시용한 냄새가 난다.



그림 21 자세히 보기

7. 한계

여러 개의 사과나 배의 이미지가 주어진다면 그 이미지에서 정확한 증상이 있는 부분을 탐지할 수 없기 때문에 잘못된 결과가 나온다.

우선 정확한 결과를 얻으려면 이미지를 선별하여 1개의 확대된 이미지가 주어져야 하고 정확한 질병의 확대된 이미지가 주어져야 좋은 예측 결과를 기대해 볼 수가 있다

8. 구현하고 싶은 것들

아래와 같이 질병 피해 정도에 따라 어떤 치료제를 투입할지 말지를 결정할 수 있기 때문에 그것에 따른 예측 결과에 따른 정보를 제공한다

| | | |
|--|--|--|
|  |  |  |
| 발병초기 | 발병중기 | 발병후기 |

그림 22 질병 피해 정도

Bounding Box를 통한 Segmentation 또는 Object Detection을 통해 사진에서 사과와 배의 구분을 먼저 할 수 있도록 만들고, 질병이 확대된 이미지가 아니더라도 질병을 알아서 구분해 줄 수 있도록 한다



그림 23 Bounding Box

9. 결론

본 프로젝트에서는 사과와 배에서 많이 발생하는 질병들의 이미지 데이터를 통해 정상과 질병들을 분류하기 위해 3개와 6개의 다중 클래스 이미지 분류를 통해 학습을 시켰고 높은 예측 결과를 도출해냈다. 하지만 한계점들이 존재하기에 그것에 대한 보완이 필요한 것으로 보인다.