

남들이 안 한다면, 내가 하자

2022 PORTFOLIO

황성현 / Hwangseonghyeon

Tel 010-2075-0697

Email hhcs0706@gmail.com

PROJECT

차량파손부위 segmentation
class weight

Python3, Tensorflow, Pytorch

차량 파손 탐지 segmentation

SOCAR에서 제공받은 차량 파손 **segmentation** 데이터를 이용해
모델이 차량 파손 여부를 **pixel**단위로 예측하는 프로젝트입니다

차량 파손 **class**로는 A, B, C가 있으며
A와 B는 같은 사진에서 **mask**가 추출되었고,
C는 전혀 다른 사진에서 **mask**를 추출되어 있었습니다

데이터 가공:

차량 파손을 감지하는 프로젝트이기 때문에
mask가 비어있는 이미지는 제외하고 학습시켰습니다

class A, B는 데이터를 가공하지 않아도 예측이 잘 이루어지지만,
C의 경우에는 **raw** 데이터를 그대로 넣으면 예측을 전혀 못 합니다

때문에 **A, B, C mask**를 **overlap**해 성능개선을 하고자 했지만,
성능개선이 이루어지지 않았습니다

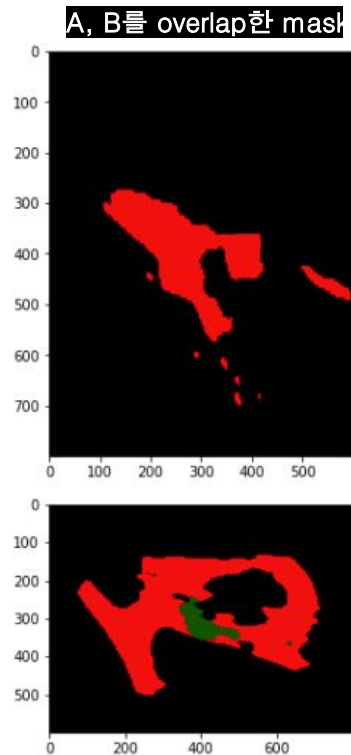
원인분석 결과 **model**이 **mask** 내에서만 예측하여 **mask** 영역이 작아진다는 것을 발견해
마스크 영역을 **dilate** 해줘서 실험해본 결과,
만족할만한 성능 개선이 이루어져 프로젝트를 끝냈습니다

DeepLabV3를 이용한 Semantic Segmentation

파손된 차량의 일부를 찍은 이미지에서
resnet으로 뽑은 feature map을 ASPP 디코더로 늘려서
mask를 만듭니다

제가 ASPP를 사용한 이유는 섬세한 mask 형태를 추출할
수 있기 때문입니다

허나 A와 B는 마스크가 섬세하지 않기 때문에 알맞지 않아
보입니다
그럼에도 제가 이 모델을 사용한 이유는 C class를
감지하기 위해서입니다



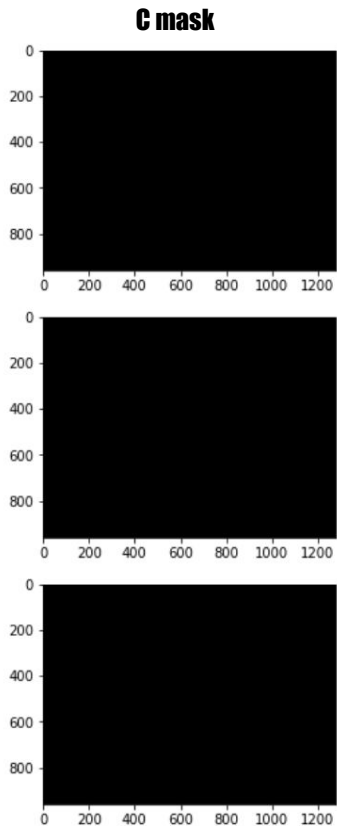
국소 마스크

C는 A, B와 이미지가 완전히 다르기 때문에
마스크를 겹쳐서 학습시킬 수 없었습니다

그래서 C만을 학습시켜본 결과 전혀 예측을 못했습니다
저는 마스크 값이 대부분 0이라 1을 이상치로 인식해 모든 값을 0으로
예측했다라는 귀무가설을 세웠습니다

그렇다면 마스크를 좀 더 세분화해서 0과 1의 차지하는 비율을 줄이면
성능이 좋아지지 않을까? 란 생각을 했고,
다음장에 그 실험 결과가 있습니다

전처리 전 예측 결과



라벨링 세분화 및 실험

먼저 dent와 scratch 마스크를 overlap 시킨 뒤 학습된 모델에 spacing 이미지를 넣어 마스크를 그리게 합니다

그렇게 그려진 A와 B 마스크에 C 마스크를 overlap 하면,

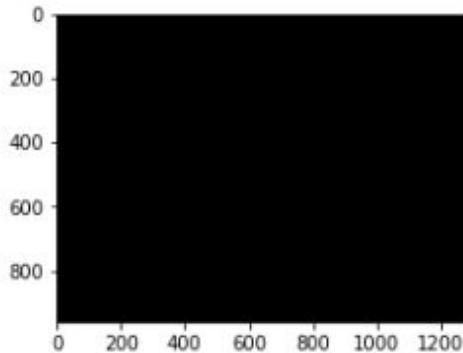
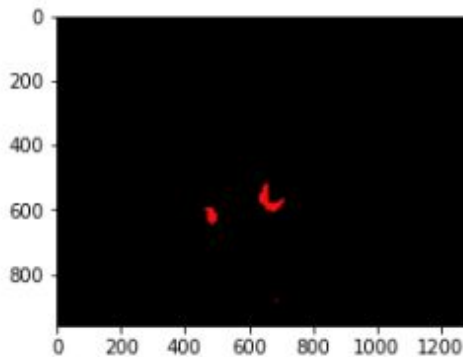
전체 영역 = $0 + C + \{(A+B) - C\}$

위와 같이 되어 0과 C 간의 비율 차이를 줄일 수 있습니다

이렇게 마스크를 만든 다음 학습을 시켜보았는데,
결과 성능이 그리 좋아지지 않았습니다
오히려 기존에 감지할 수 있던 class마저도 감지할 수 없게 되었습니다

일부는 C가 생기는 접합선 부분에 마스크가 생기긴 했지만,
답을 맞춘 mask는 거의 없었습니다

A, B, C overlap mask



2차 실험

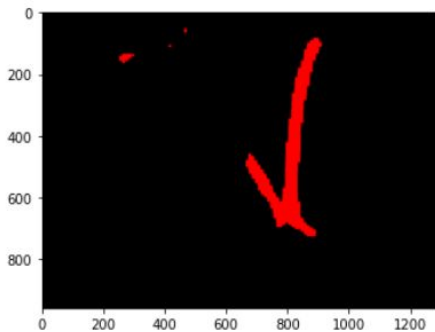
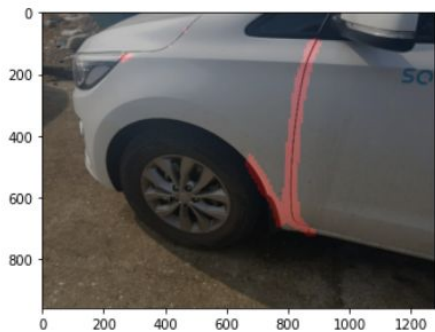
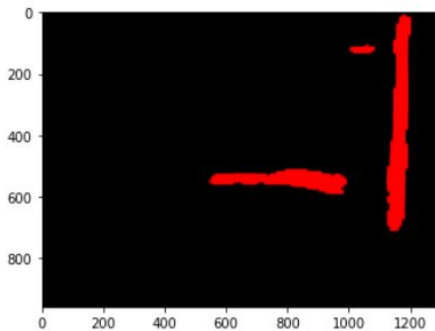
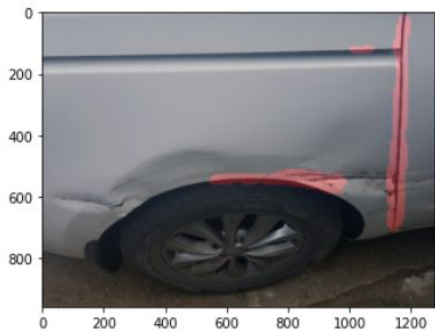
마스크가 차지하는 비율을 세분화했지만,
오히려 성능을 저하시키고 말았습니다

원인을 알아내기 위해 출력 결과를 살펴보니,
대부분이 **target mask** 영역을 벗어나지 않고
그 영역 안에서 예측을 하다보니 **mask**의 크기가 작아지는 현상을
발견했습니다

저는 마스크가 작아지는 현상 때문에 **C class mask**를 그대로 학습시키면,
예측을 못하는 게 아닐까란 가설을 세웠고,
C class mask의 외곽선을 기준으로 이미지를 넓혔습니다

그 결과 만족할만한 성능을 보여주어 프로젝트를 끝내게 되었습니다

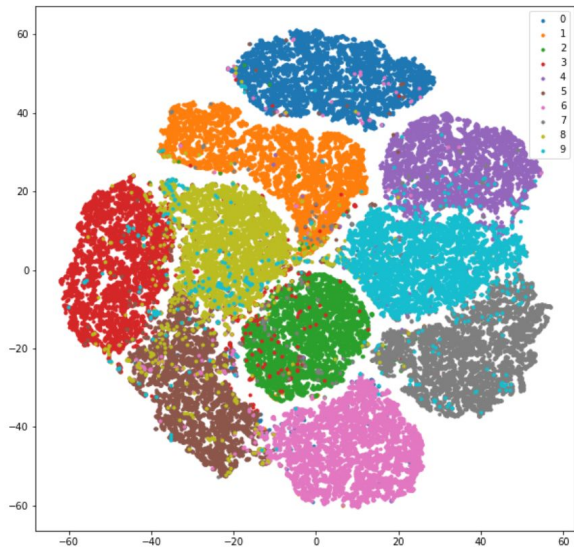
C dilate mask



원인 분석

제가 만든 deeplabv3는 **target** 영역 안에서 예측하려는 성향이 강했습니다
이는 **loss function** 때문이라 추측되며,
실제로 AIFELthon에서 **loss function**을 커스텀해 문제를 해결한 팀도 있었습니다

저 또한 **loss**를 바꿔서 해결해보고자 했지만,
model을 바꿀 수 없는 상황이 발생할 수 있다 생각해
저는 전처리 작업으로 해결했습니다



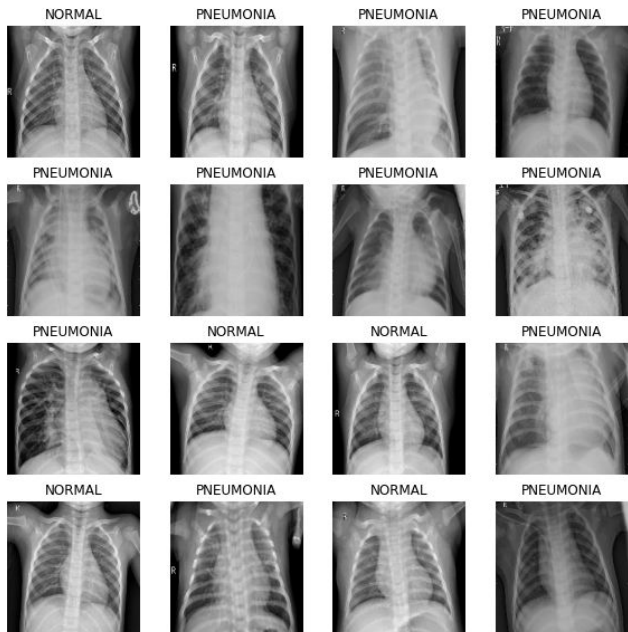
추후 과제

deep k-means cluster로 이미지를 나눠보고자 합니다
제가 C class mask를 봤을 때 규칙성을 찾기 굉장히 어려웠습니다

저는 그 원인이 일관적이지 않은 라벨링의 문제라 생각했고,
그렇다면 비슷한 이미지끼리 군집화시켜 규칙성을 부여하여
가장 군집이 큰 이미지셋을 추출하면 최적의 라벨링 샘플을 알 수 있기 때문입니다

Class Imbalance

class weight로 해결하기



```
Normal images count in training set: 1070  
Pneumonia images count in training set: 3115
```

class 분류 데이터가 unbalance할 때

Normal class num == 1000

Pneumonia class num == 3000

class weight로 클래스에 가중치를 부여해

부족한 데이터를 보완했습니다

그 결과 recall, accuracy, loss 가

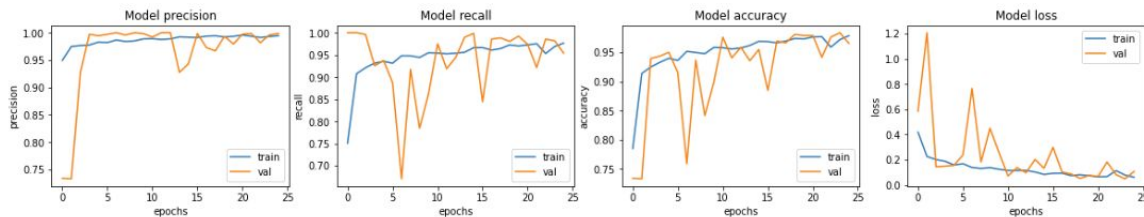
epoch가 증가할 수록 값이 안정적이게 되었습니다

https://github.com/hwangseonghyeon/test/blob/master/%5BE_09%5DXray.ipynb

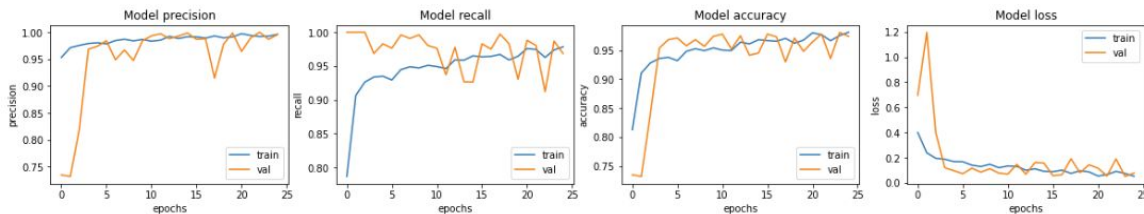
Class Imbalance

class weight로 해결하기

before



after



데이터 양이 **imbalance**할 때
class weight로 클래스의 가중치를 부여해
부족한 데이터를 보완했습니다

그 결과 **recall**, **accuracy**, **loss** 등이
epoch가 증가할 수록 값이 안정적이게 되었습니다