캡스톤 디자인 개인 보고서

전자공학과 20184197 박건우

주제: 암전이 예측 알고리즘 개발

주제 선정 이유: 최근 50세 미만 성인의 암 발병률이 증가하고 있습니다.

특히 젊은 나이층인 20대 중후반부터 30대 초반 연령대에서, 30대 중반부터 40대 초반

연령대보다 발병률이 더 많은 것으로 조사됐습니다. 과거 나이 든 사람이 걸리는 병에서

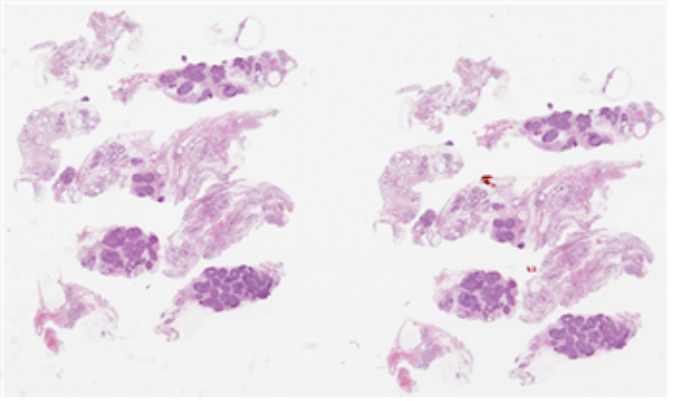
전 연령층이 걸릴 수 있는 병으로 인식이 변하고, 젊은 층의 암 발병률이 오르고 있습니다.

프로젝트 진행 목표및 방향성: 암 환자에 포커스를 두었고, 암인지 아닌지를 판별하는 것이 아닌 환자의 예후에 도움을 줄 수 있는 암세포 전이 예측주제를 선정하였습니다.

데이터셋: 데이콘에서 세포 이미지와 환자 정보를 포함한 데이터셋

개발환경: hpc 서버를 통해 파이토치, 주피터 환경, 또한 7개의 py 파일을 합쳐 한 개의 알고리즘을 생성하고 예측

림프절 조직 이미지데이터를 활용



림프절 이미지

림프절 이미지를 학습시키기 위한 모델 : CNN 모델이 이미지를 판별하는 데 있어 가장 많이 사용되는 모델이라고 합니다. 따라서 저희는 암전이 예측을 위해 CNN 기반 모델을 사용했습니다.

이 데이터셋을 선정한 이유: 결국 암세포는 림프절을 통해 전이 될 수 있습니다. 전이를 예측 가능하다면 사전에 파악해 환자의 치료에 도움을 줄 수 있습니다.

데이터셋 설명

데이콘에서 세포 이미지와 환자 정보를 포함한 데이터셋을 확보했습니다. 이 데이터셋은 연세대학교 의과대학에서 공유된 데이터로 신뢰성을 확보했습니다.

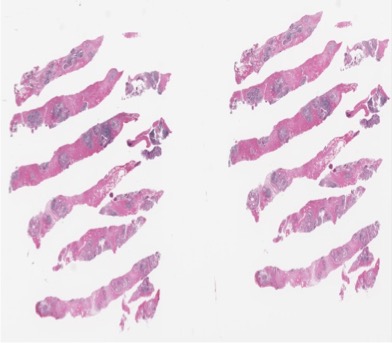
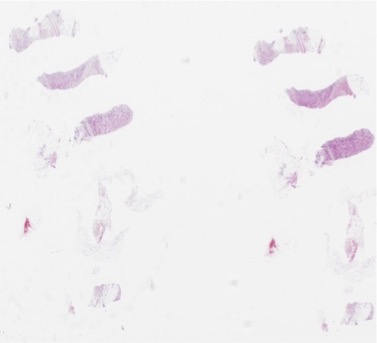
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

(train image 정상 이미지, 비정상 이미지를 보여주며)

이 이미지 데이터는 훈련용 정상 이미지와 비정상 이미지를 파악할 수 있습니다.

train 데이터는 1,000장을 포함하고 있습니다. test 데이터는 250장을 포함하고 있습니다.



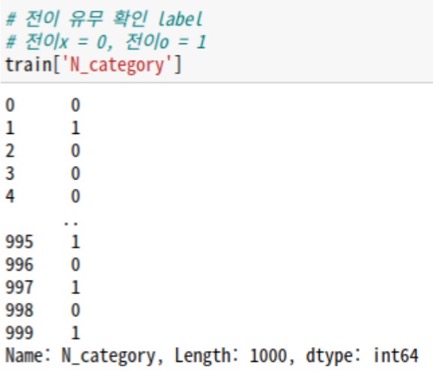
왼쪽이 비정상 오른쪽이 정상

Mask 이미지

train 데이터 중 비정상 이미지에 대한 mask 이미지입니다. mask 이미지는 50장을 포함하고 있으며, 컬러 이미지가 아니기 때문에 그레이스캐일로 불러와 이용하였습니다.

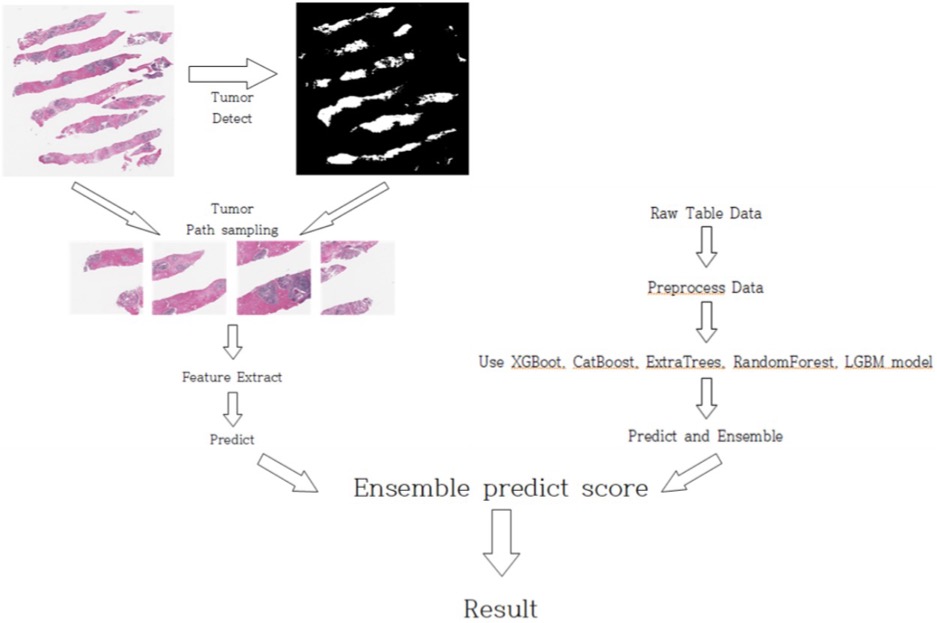


데이터의 전이유무 확인



(csv predict socre + image predict score 형태로 진행할 거라는 이미지 보여주며)

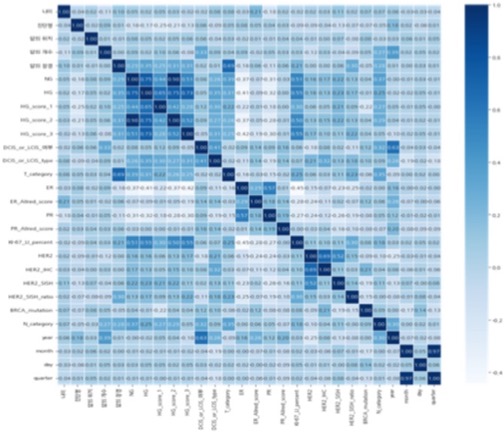
저희는 csv 데이터만을 학습시켜 예측한 정확도와 이미지만을 학습시켜 예측한 정확도를 합칩니다. 이 과정을 앙상블이라고 하는데, 앙상블은 여러 개의 모델을 결합해 성능을 끌어올리는 기법입니다. 이 기법을 이용해 예측 정확도를 충분히 확보하고, 예측 정확도와 예측한 이미지를 웹을 통해 시각화하였습니다.



csv 컬럼별 상관관계 이미지

csv 데이터를 학습시키기 위해서 우선 csv 데이터의 컬럼별 상관관계를 파악해야 합니다.

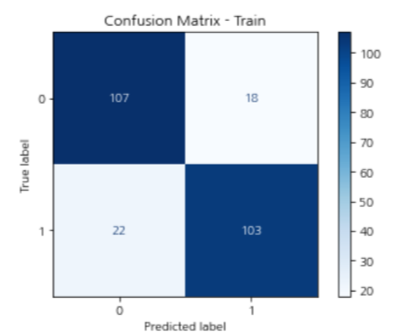
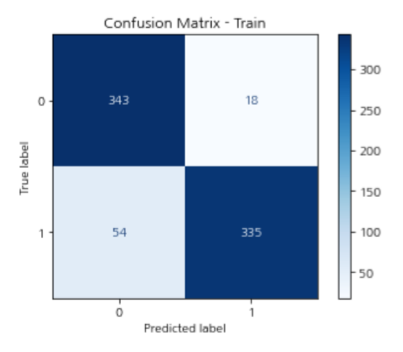
상관관계를 파악해야 머신러닝이 예측한 지표와 비교하며 모델의 성능을 평가할 수 있기 때문입니다. 이 사진을 통해 csv 데이터의 컬럼별 상관관계를 알 수 있습니다.



- 좌측 상단부터 대각선 방향은 서로 같은 컬럼의 상관관계값이라 1의 값을 나타냅니다. 값이 1에 가까울수록 상관관계가 높습니다. 따라서 csv를 머신러닝에 학습시키기 이전에 중요한 컬럼의

값은 살리고 중요하지 않은 값은 평균값으로 바꾸는 전처리가 필요합니다.

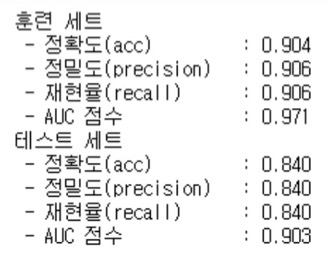
XGB 결과값 이미지

이 사진은 머신러닝 모델 중 가장 값이 좋았던 XGB모델의 값을 나타냅니다. 좌측사진은 train 이미지에 대한 confusion matrix이며 우측 사진은 test 이미지에 대한 confution maxrix 입니다.

이 사진은 XGB에 대한 정확도, 재현율, AUC점수를 나타내봤습니다.

정확도는 모델이 예측한 값이 얼마나 일치하는지를 측정하는 지표입니다. 재현율은 모델이 positive로 예측한 값이 실제 positive인지를 측정하는 지표입니다. AUC점수는 모델이 False Positive Rate와 True Positive Rate 사이의 관계를 나타내는 곡선 ROC Curve의 아래 면적을 나타내며 모델의 분류 성능을 나타내는 지표입니다.

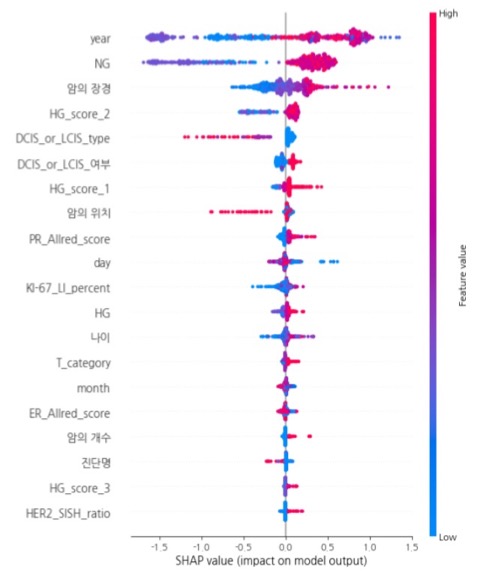


여기서 False Positive Rate는 모델에서 Positive로 잘못 분류된 값을 나타내는 값입니다.

Ture Positive Rate는 모델에서 Positive로 잘 분류된 값을 나타내는 값입니다. 이 비율들은 이전에 보여주었던 confusion matrix 이미지를 통해 잘 알 수 있습니다.

XGB가 판단한 컬럼별 중요도이미지를

아래 사진을 통해 csv를 예측하는데 사용한 모델인 XGB가 판단한 컬럼별 중요도를 알 수 있습니다. 이 값과 초기에 구했던 컬럼별 상관관계를 통해 XGB 모델이 얼마나 정확하게 예측했는지 평가할 수 있습니다.



아래 사진은 csv 앙상블 모델로 예측한 정확도입니다. 앙상블한 머신러닝 모델은

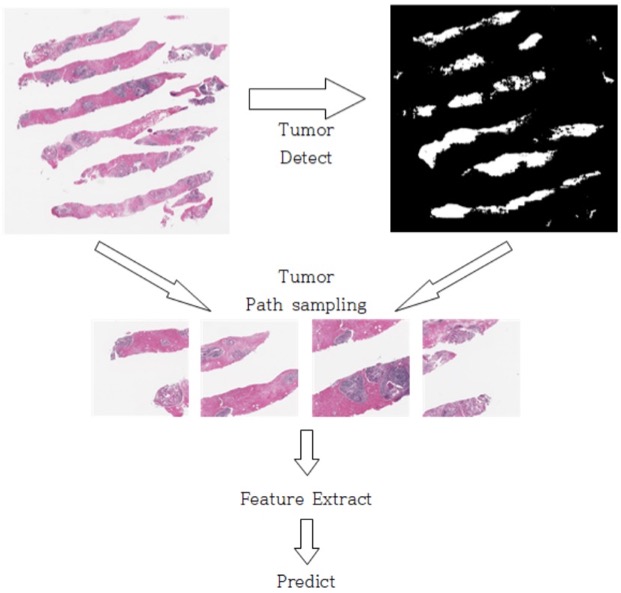
XGBoot, CatBoost, ExtraTrees, RandomForest, LGBM 이며, 자동으로 가장 낮은 예측값을 가진 모델은 삭제하여 4개만 선정해 앙상블을 진행했습니다. voiting을 hard로 설정하면, 각 모델이 예측한 값을 타이트하게 평가해 새로운 예측값을 도출합니다.

이 모델 중 가장 높은 예측값을 보였던 XGBoost 모델은 0.84의 정확도를 보였으며,

앙상블 후 정확도는 0.848의 값을 나타냈습니다. 백분율로 나타내면 약 0.8%의 상승을 보였습니다.

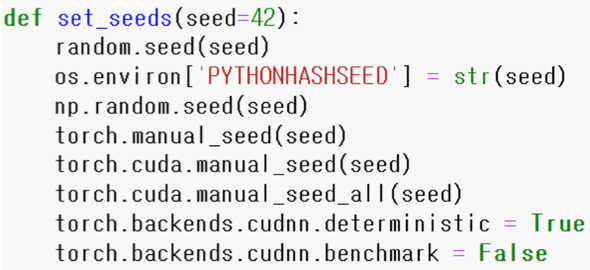


이미지 예측 알고리즘의 전반적인 overview



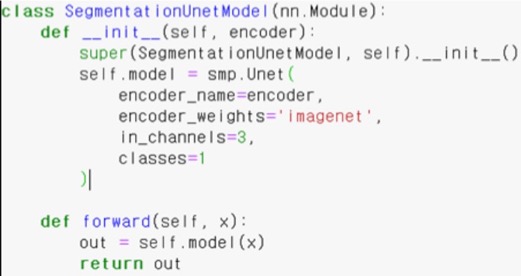
각 py 파일 설명

첫 번째 py 파일은 유틸리티를 정의한 파일입니다.



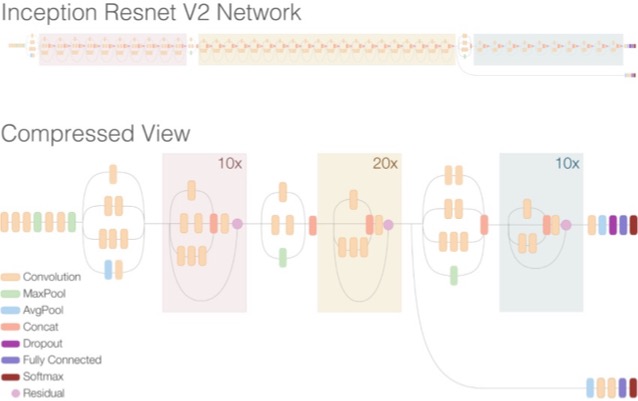
유틸리티에선 cuDNN 라이브러리의 연산 최적화 기능 비활성화, 파이썬 해시시드를 설정해줍니다. 이렇게 설정해주는 이유는 재현성 보장, 자료형 데이터를 불러올 때 항상 동일한 순서로 로드하기 위해 해줘야 합니다. 여기서 재현성이란 동일한 GPU 환경에서 실행할 때마다 랜덤값이 들어가 정확도가 달라지는 경우를 방지할 수 있습니다.

두 번째 py 파일은 모델을 정의한 파일입니다.

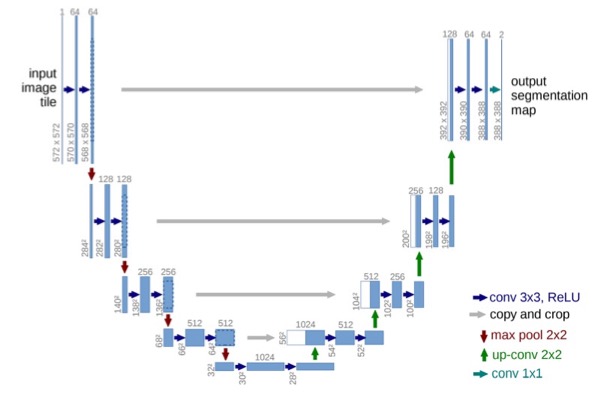


사용한 모델:

Inception-ResNet V2 모델과 segmentation을 학습시키기 위해 파이토치에서 제공해주는 Unet 을 이용했습니다. 훈련에 사용될 이미지의 해상도 5311 x 2308이며, 96dpi, 24비트의 픽셀정보를 가지고 있는 데이터입니다. 이 데이터는 고해상도 이미지에 속하기 때문에 논문을 통해 얻어낸 정보에 의해 일반적으로 알고있는 ResNet 모델보다는 Inception-ResNet 모델을 사용하게 되었습니다.



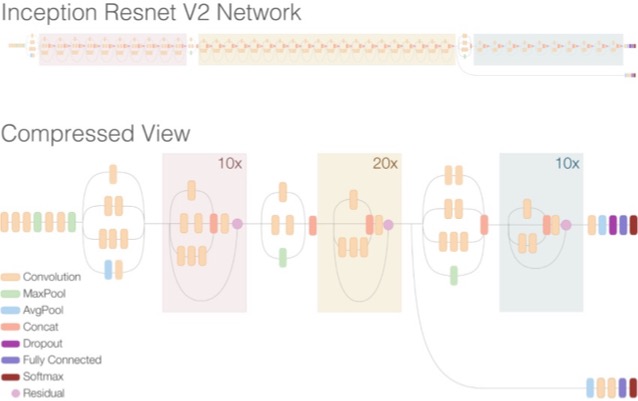
Inception-Resnet 모델은 이미지의 특성값을 추출하기 위해 인코더로 사용하고, Unet 모델은 세그멘테이션 학습을 수행하기 위해 저희조는 디코더 부분으로 이용했습니다. 여기서 인코더는 스킵레이어를 기준으로 좌측을 인코더, 우측을 디코더라고 합니다.



인코더를 Inception-ResNet, 디코더를 Unet으로 설정하면 생기는 장점은 명확합니다. Inception-ResNet은 고해상도 이미지에서 고해상도 데이터로 추출할 수 있다는 장점을 가지고 있으며, 이를 Unet으로 업샘플링 시켜주게 됩니다.

위 이미지는, 모델을 smp 라이브러리로 불러온 것을 확인할 수 있습니다. 각 모델의 가중치는 전이학습 모델의 가중치를 불러올 수 있기 때문에 학습을 최적화할 수 있습니다. 가중치는 ImageNet이며 이미지 분류, 이미지 특성 및 추출, 객체 감지 모델에 효과적입니다.

데이터가 적거나 고해상도 이미지 분류에 좋은 성능을 보여주기 때문에 저희 조의 상황과 잘 맞아 이렇게 선택하였습니다.

 Inception-Resnet 모델에 대해 자세히 설명하겠습니다. 구조를 보시면 잔차연결이라는 개념이 도입되어있습니다. 일반적으로 신경망은 데이터를 변환하고 출력하기 위해 비선형 활성화 함수를 통해 변환합니다. 이렇게 되면 신경망이 깊어질수록 기울기 소실 문제가 발생합니다. 이를 해결하기 위해 도입된 방법으로 컨볼루션 필터와 폴링 연산을 병렬로 연결해 입출력을 더해 기울기 소실을 막습니다. 따라서 Inception-Resnet 은 고해상도 데이터를 얻을 수 있으면서도 깊은 신경망이 가능합니다.

Unet 의 가장 큰 특징인 인코더에서 디코더를 이어주는 레이어에 스킵레이어가 존재한다는 점입니다. 이는 인코더의 특성값을 손실 없이 디코더로 연결해 주어 학습할 수 있다는 장점이 있습니다. 또한 segmentation 학습에 특화되어있다는 점이 있습니다.

세 번째 py파일은 커스텀 데이터셋을 정의한 파일입니다.

커스텀 데이터셋을 제작하기 위한 코드는 캐글과 깃허브에서 가져왔습니다. 수학적으로 계산된 수식으로 전처리 하기 때문에 대부분 기준코드를 제공해줍니다. 따라서 직접 설정한 부분은 파라미터입니다. 저희가 설정한 파라미터는 데이터 파일의 주소, target\_value, image\_transform, image\_size, stride, threshold, binary\_threshold, cnt\_threshold, max\_width, n\_patches 입니다.

각 파라미터 역할설명

- target\_value : 세그멘테이션의 대상이 되는 값입니다. 객체의 픽셀에 해당하는 값이 무엇이라고 지정할 수 있습니다.

- image\_transform : 데이터셋 증강에 활용됩니다. 예를 들어 모델이 이미지 데이터를 학습 할 때, 좌우로 뒤집어 학습한다거나 상하로 뒤집거나 등을 이용해 데이터셋 증강에 도움을 줍니다.

- stride : 이미지의 패치를 추출할 때 사용되는 값입니다. 필터를 적용하는 간격을 의미하는데 만약 값이 2라면 필터가 2의 간격으로 이미지 픽셀값을 조절합니다.

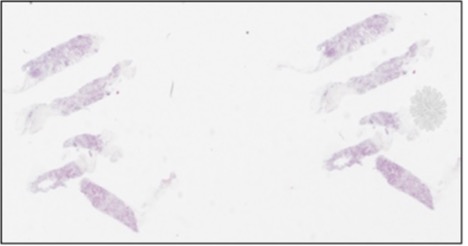
- threshold : 이미지의 특성값을 이진화 시키는 데 사용합니다. 여기서 이진화를 선택하는 옵션을 binary\_threshold에서 설정할 수 있는데 대표적인 설정 옵션으로 ostu가 있습니다. 이 옵션을 적용하면 이진화 임곗값으로 ostu의 알고리즘을 사용하게 됩니다. Otsu 알고리즘은 이미지의 히스토그램을 분석하여 최적의 이진화 임곗값을 자동으로 결정하는 방법입니다. 이미지의 밝기 변화나 노이즈에 강해 이미지 이진화 작업에 효과적으로 사용할 수 있습니다.

- cnt\_threshold : 패치의 값이 특정 값 이하면 그 값은 사용하지 않겠다는 값을 정해주는 파라미터입니다. 너무 작은 값이면 충분한 특성이 있지 않아 모델에 넣어 학습 시 정확도를 떨어뜨릴 수 있어 설정해줍니다. 패치는 이미지를 작은 조각으로 잘라 추출한 값을 의미합니다.

- max\_width : 주로 image\_size에서 설정된 이미지 크기와 동일하게 가져갑니다. 추출할 이미지 특성의 최대 너비를 의미합니다. 이 값보다 이미지가 크게 된다면 추가적인 이미지 보정이 필요하므로 두 개의 값을 동일하게 가져갑니다.

- n\_patches : 생성할 패치의 개수를 의미합니다. 패치는 이미지를 작은 조각으로 잘라 추출한 값들을 의미합니다. 대부분 세그멘테이션 작업에 활용됩니다. 설정해주는 이유는 암 종양이 있는 패치와 없는 패치를 조절해 데이터셋에 포함될 종양이 있는 패치의 비율을 조절할 수 있습니다. 데이터셋의 비율을 조절할 수 있어 여러 값으로 실험해 가장 좋은 값을 찾아야 하는 파라미터입니다.

네번째 py파일은 이미지를 분할해주는 파일입니다.

첨부된 이미지를 참고하시면, 원본 mask 이미지는 원본 train과 다르게 일부분만 제공되고 있습니다. 심지어 train 이미지 1000장, test 이미지 250장에 대한 mask 이미지를 제공하고 train 이미지에 대한 mask 50장만 제공됩니다. 이 데이터를 학습에 바로 사용했을 땐 정확도를 0.4-0.5정도 밖에 얻지 못하였습니다.

cropped한 이미지

데이터의 불균형으로 판단하여 주어진 1250장에 대한 mask 이미지를 생성했습니다.

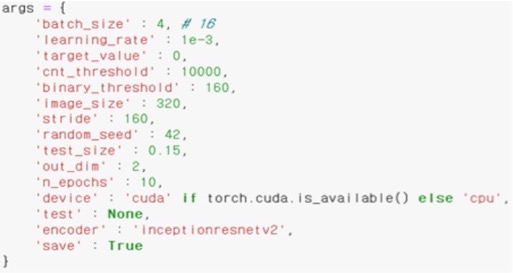
원본 mask의 피처를 기준으로 train 이미지를 모두 cropped 해주어 학습시킵니다.

눈, 스크린샷, 겨울이(가) 표시된 사진

자동 생성된 설명

segmentation\_train.py 이미지를

다섯 번째 py파일은 네번째에서 실행 후 나온 이미지를 학습시키기 위한 파일입니다.



이전 장에서 cropp한 이미지를 이용해 훈련 시킵니다. 훈련모델은 전에 설명드렸던 모델에 넣어 학습시킵니다. optimizer은 Adam 을 이용했으며 scheduler 함수를 사용했습니다. scheduler 함수의 파라미터로 T\_max, eta\_min을 조절했습니다.

각 파라미터별 설명

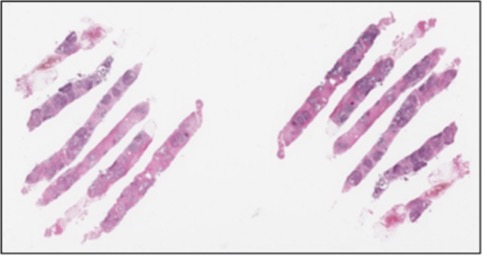
- T\_max : 코사인 함수에서 설정된 값만큼의 주기 동안 학습률이 조정됩니다. 지정된 값만큼 epoch가 지나면 학습률을 초기속도로 돌아갑니다.

- eta\_min : 과적합 방지를 위해 설정해줍니다. 학습률을 감소시키는 최솟값으로 설정됩니다.

이미지가 학습되는 알고리즘은 다음과 같습니다. 데이터 파일 path 설정 – train, test, val 데이터로 분할 – 분할된 데이터들을 customdataset 을 통해 전처리 – dataloader 를 통해 모델에 넣기 위해 알맞은 형태로 변환 – 모델의 파라미터 설정 후 데이터를 넣어 학습. 으로 진행됩니다.

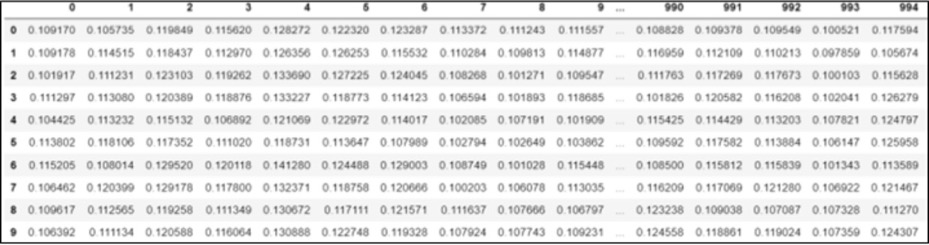
여섯 번째 py 파일은 mask 이미지를 생성하기 위한 파일입니다.

모델에 넣어 생성하면 이렇게 이미지 전체에 대한 mask 이미지를 얻을 수 있습니다.



일곱 번째 py 파일은 이미지 피처 추출을 위한 파일입니다.

train 테이블 데이터, train 이미지, 생성된 train mask 이미지, test 테이블 데이터, test 이미지, 생성된 test mask 이미지를 이용해 피처를 추출합니다. 피처 추출에 이용하는 모델은 inception-resnet v2 를 이용합니다. 추출한 값은 csv즉, 테이블 데이터로 저장합니다.



여덟 번째 py 파일은 최종 예측을 위한 파일입니다.

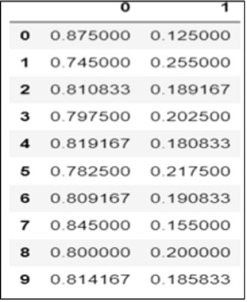
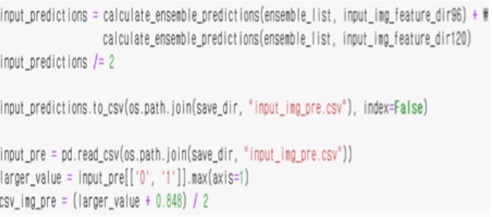
from autogluon.tabular import TabularPredictor 를 이용해 예측합니다.

(결과값 보여주며)

AutoGluon은 다양한 머신러닝 알고리즘과 모델 아키텍처를 지원하며, 사용자는 별도의 모델 선택 또는 구성없이도 최적의 모델로 csv 데이터를 분석하고 예측할 수 있습니다.

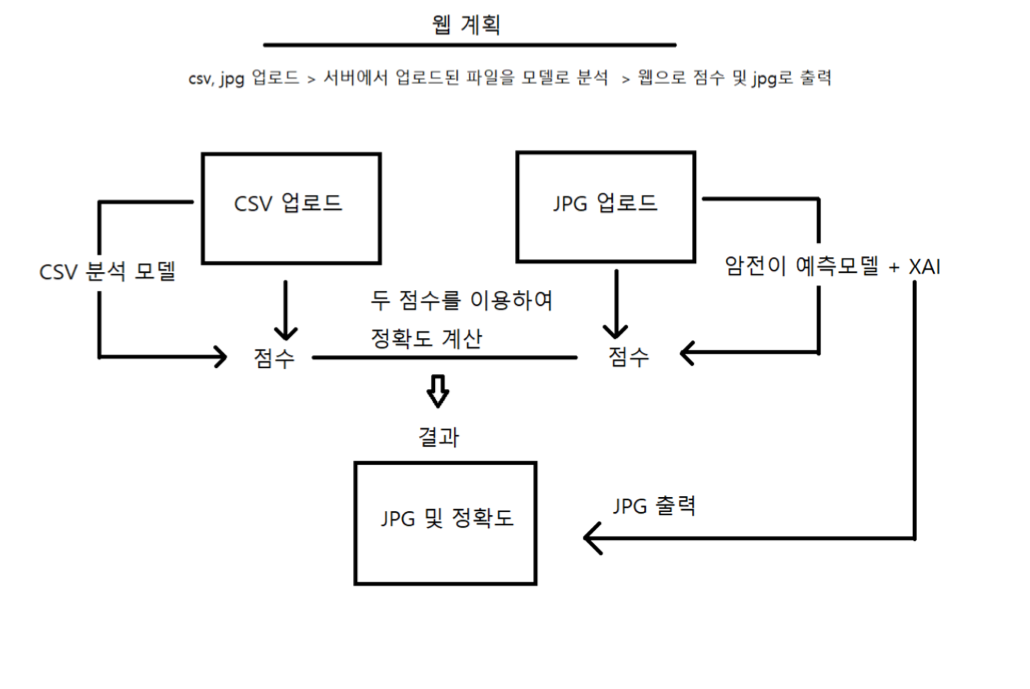
이미지의 피처를 csv 형태로 저장했기 때문에 사용가능한 라이브러리입니다. 이미지의 전이 유무 정답값인 N\_category를 합쳐 예측합니다.

단일 피처값으론 높은 정확도를 기대하기 힘들어 패치를 96 단위로 쪼깨어 추출한 피처 예측값과 패치를 120 단위로 쪼개어 추출한 피처 예측값을 앙상블해 결과로 활용하였습니다.



Csv 예측 정확도 + 이미지 예측 정확도 코드 autogoloun을 이용한 전이 예측 확률 1 전이 0 비전이

웹



이 이미지는 전이 예측 결과를 시각화하기 위해 사용한 웹 계획표입니다. 이전에 학습시킨 모델을 웹에 로드해 줍니다. 사용자가 원하는 임의의 이미지를 넣게 되면 이미지에서 암인 부분을 mask해준 이미지를 생성합니다. 그 후 이미지의 정보를 가진 csv 데이터를 넣게 되면 예측 후 Input Image에 대한 예측값을 제공해줍니다.

웹또한 hpc에서 제작했으며

flask를 이용해 웹에 인공지능 알고리즘을 구현했습니다. 파이썬 기반의 언어라 인공지능 알고리즘과 웹간 연동이 쉽다는 장점이 있습니다.

프론트엔드는 HTML, CSS, JavaScript를 이용했습니다.

Hpc에서는 외부 url을 읽어올 수 없어 이미지를 직접 다운받아 사용했습니다.

처음엔 간단한 박스모양의 라인으로만 정리하려다 사용자 관점에서 보니 너무 별로여서 뒷 배경을 투명하게 반영하되 형태는 남아있게 바꾼 후 그림자를 이용하여 디자인을 하였습니다. 사용자의 가독성을 고려하여 마우스를 원하는 박스에 올려 놓을시 흰색으로 배경색이 변해 가독성을 향상시킵니다. 전체적인 웹 구조는 css의 flex를 주로 이용하여 배치를 설정했습니다.

플라스크는 파이썬 기반 언어라 주피터노트북 환경에서 서버를 생성가능합니다.

저희는 플라스크를 이용하여 인공지능 모델 및 여러가지 파일을 통해 여러가지 결과를 웹에서 표현 할 수 있도록 만들었습니다. Ex(annotation mask생성 버튼 누를 시 flask에 실어둔 annotation mask 생성) 사전에 학습시킨 모델을 통해 csv, image 데이터의 예측을 웹 페이지에 출력합니다

(annotation 코드를 웹에 연동방법)

사용자가 넣어준 임의의 이미지에서 암이라 판단되는 부분을 mask 해주는 코드입니다. 이 코드는 이전에 구현했던 annotation 생성 코드를 웹에 맞게 고도화하여 넣어주었습니다. 임의의 이미지가 입력되면 플라스크에 구현된 이 코드가 실행되어 mask를 생성해줍니다.

(feature 추출 코드를 웹에 연동 방법)

사용자가 넣어준 이미지의 feature를 추출해주는 코드입니다. 마찬가지로 이전에 구현했던 feature 추출 코드를 웹에 맞게 고도화하여 넣어주었습니다. 웹 페이지의 featur 탭에서 사용자가 이미지에 대한 csv 데이터를 넣게 된다면, 플라스크에 구현된 이 코드가 실행되어 이미지의 feature를 추출해 csv 형태로 저장합니다.

텍스트, 스크린샷, 암초, 강장동물이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명텍스트, 편지, 명함, 스크린샷이(가) 표시된 사진

자동 생성된 설명