

Kaggle Competition

HACKING THE HUMAN BODY



코드득

옥지성 박재현 박한빈 안홍민

CONTENT

01 개요

02 팀 구성

03 수행 절차 및 방법

05 수행 결과

05 자체 평가 의견

Hacking the Human body



목표

- kaggle 대회로 5개의 기관에서 주요 기능을 하는 세포 조직(FTUs)을 segmentation

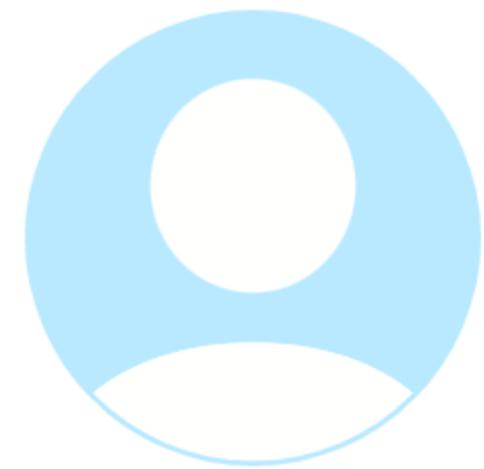
FTU(Functional tissue units)란?

- FTU를 기관의 주요 생리학적 기능을 수행하는 가장 작은 수준의 조직(즉, 세포 집단 이웃)으로 정의한다.

목적

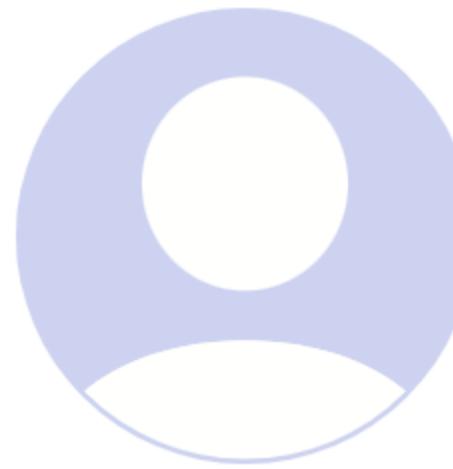
- 대회 참가를 통해서 이미지를 다루는 기술과 Segmentation에 대한 이해를 높이는 것

MEMBER



옥지성

PM



박재현

MEMBER



박한빈

MEMBER

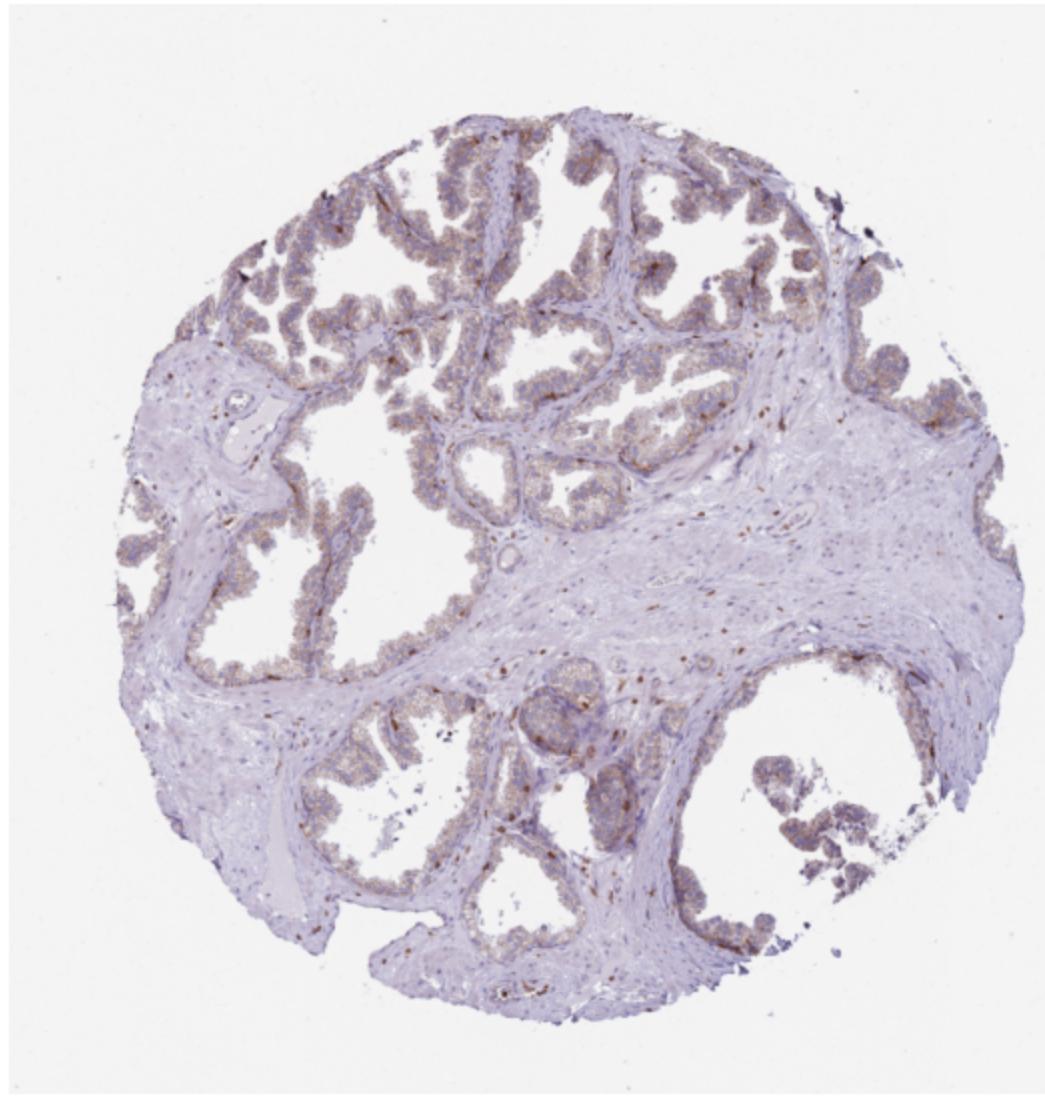


안홍민

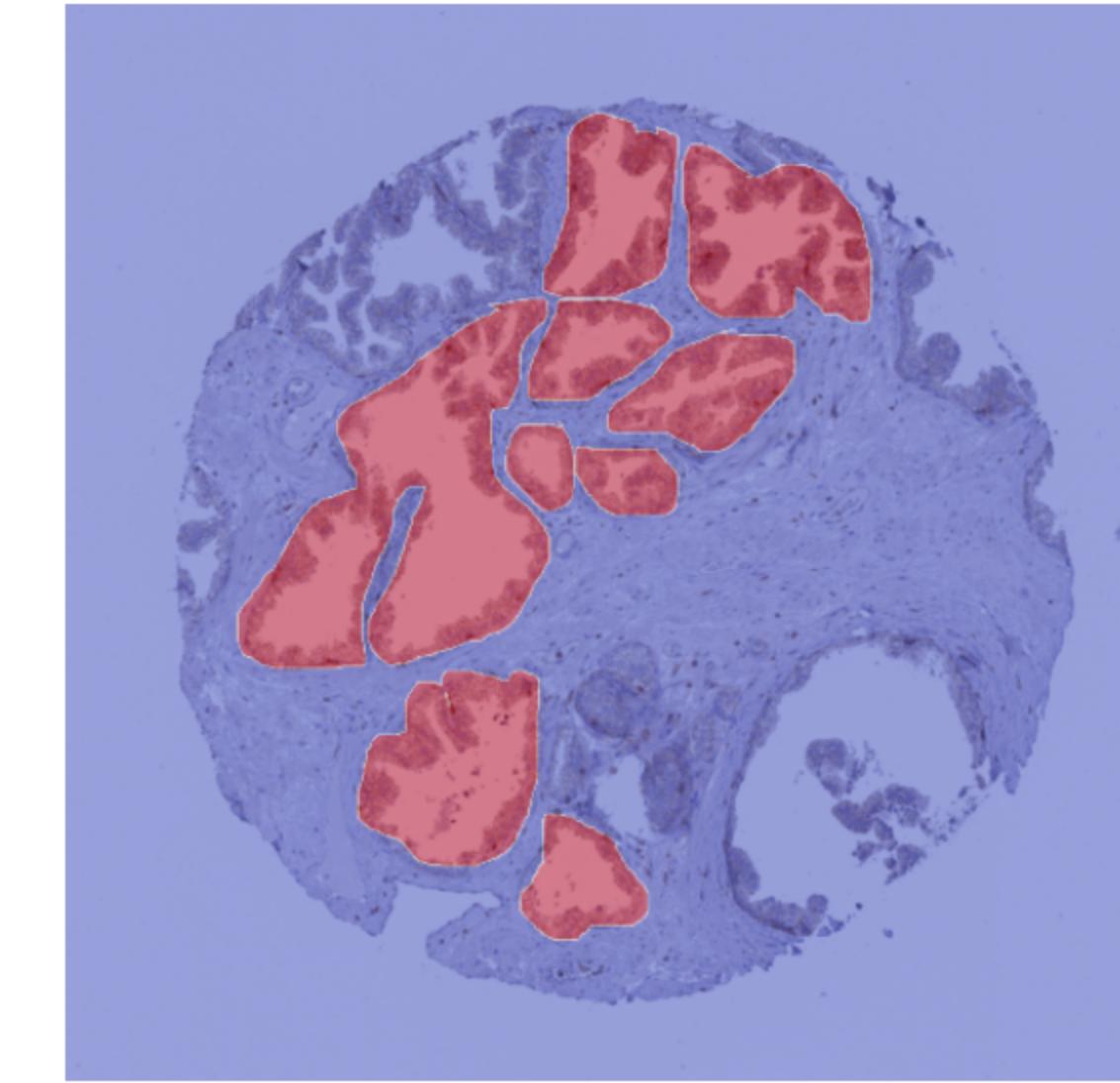
MEMBER

DATA

- 5개 기관(신장, 전립선, 비장, 대장, 폐)의 조직검사 슬라이드 이미지와 FTUs 정보



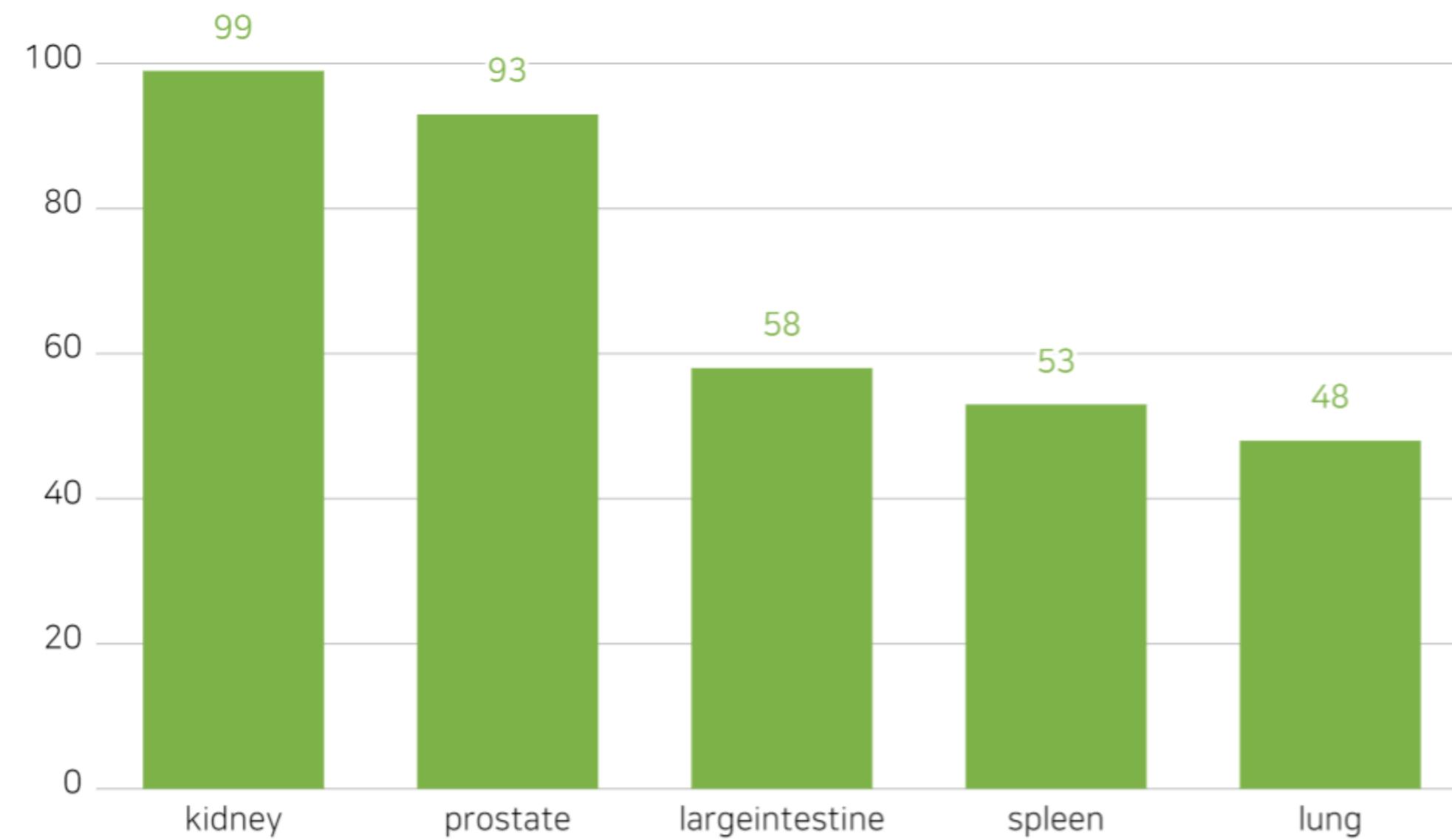
Original



Mask

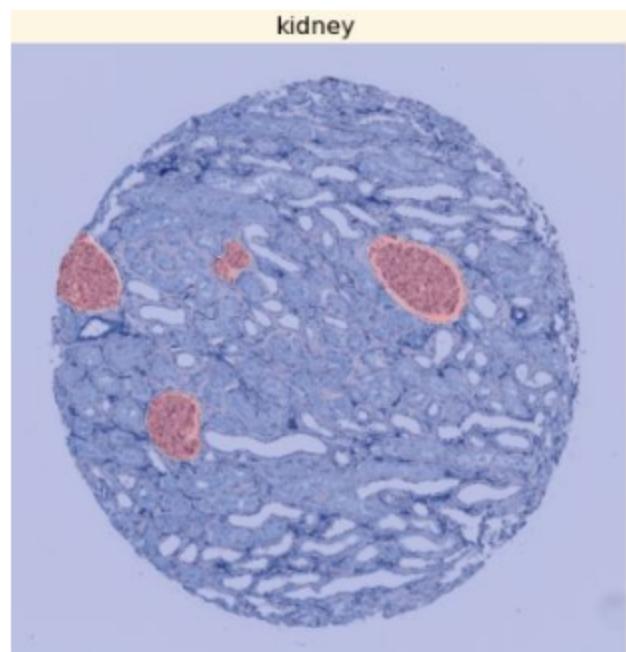
DATA

- 5개 기관(신장, 전립선, 비장, 대장, 폐)의 데이터 수 비교 (총 351개의 데이터 제공)

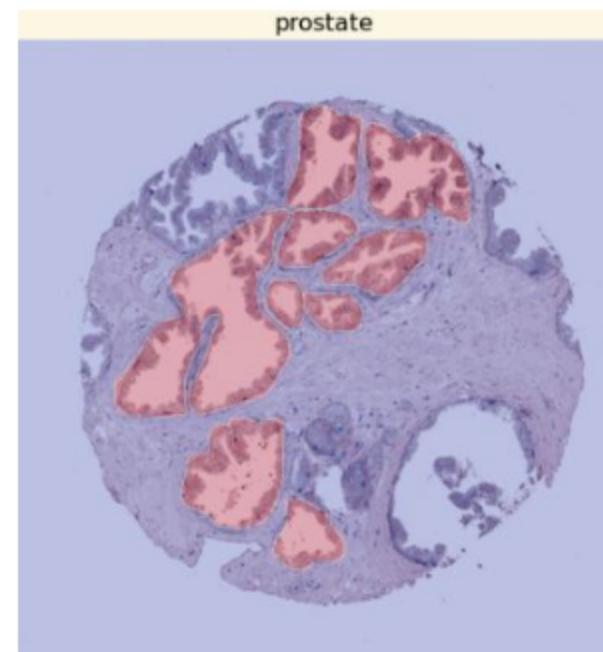


DATA

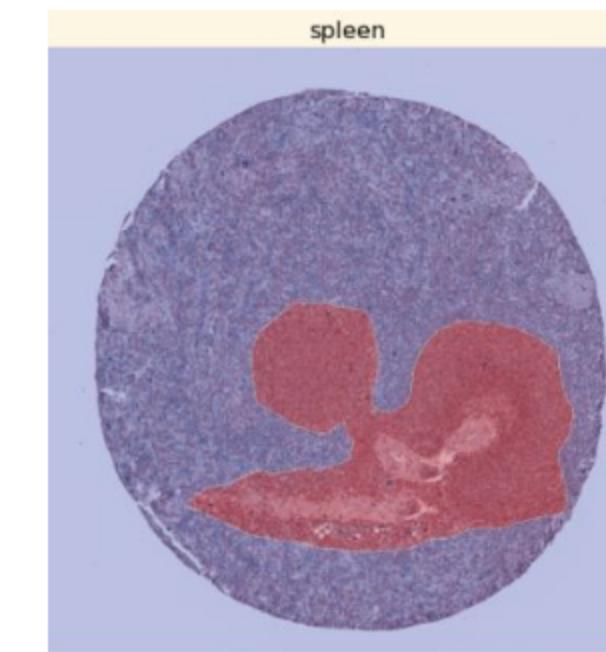
- 5개 기관(신장, 전립선, 비장, 대장, 폐) 별 이미지



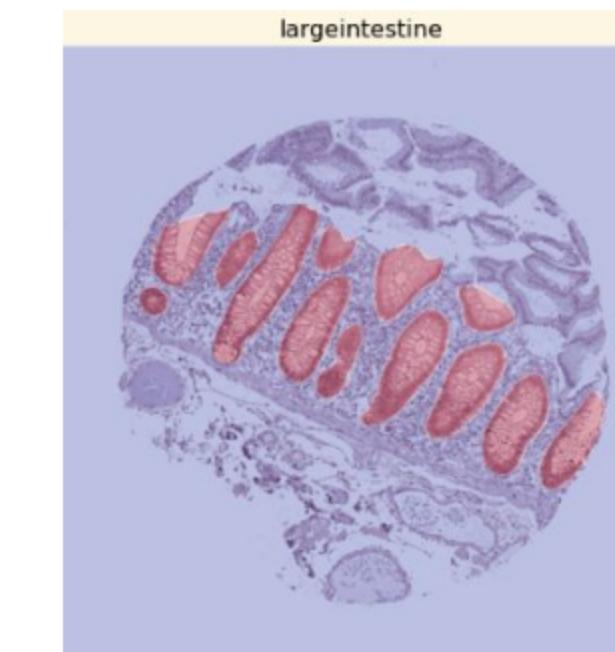
신장



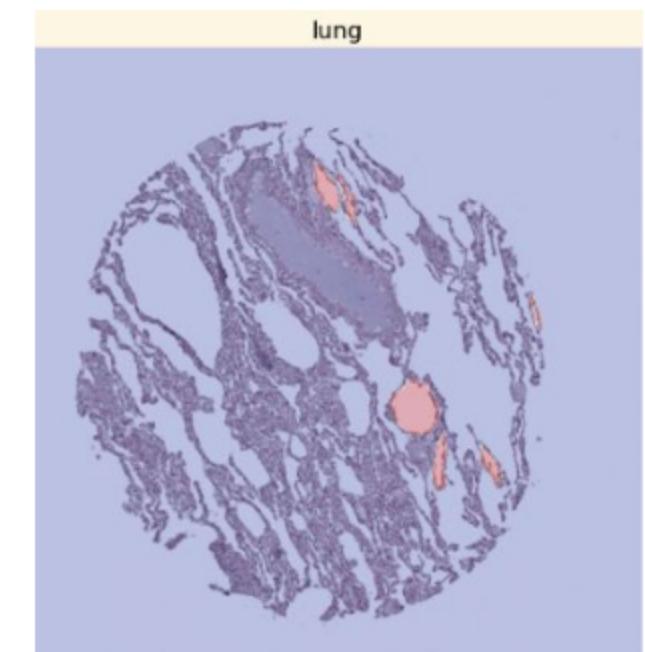
전립선



비장



대장

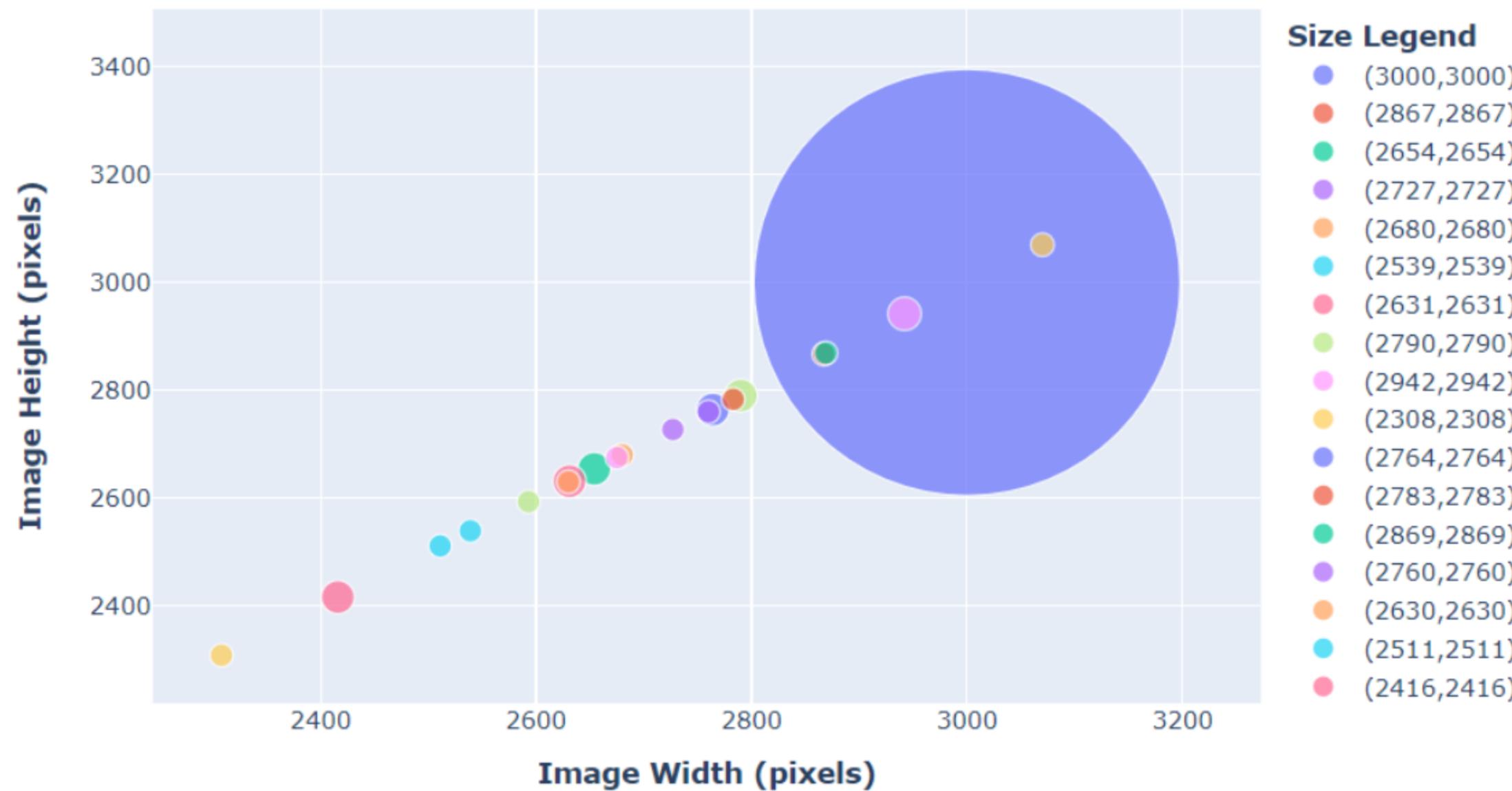


폐

DATA



- Image Size (Height, Width) 별 분포도



DATA



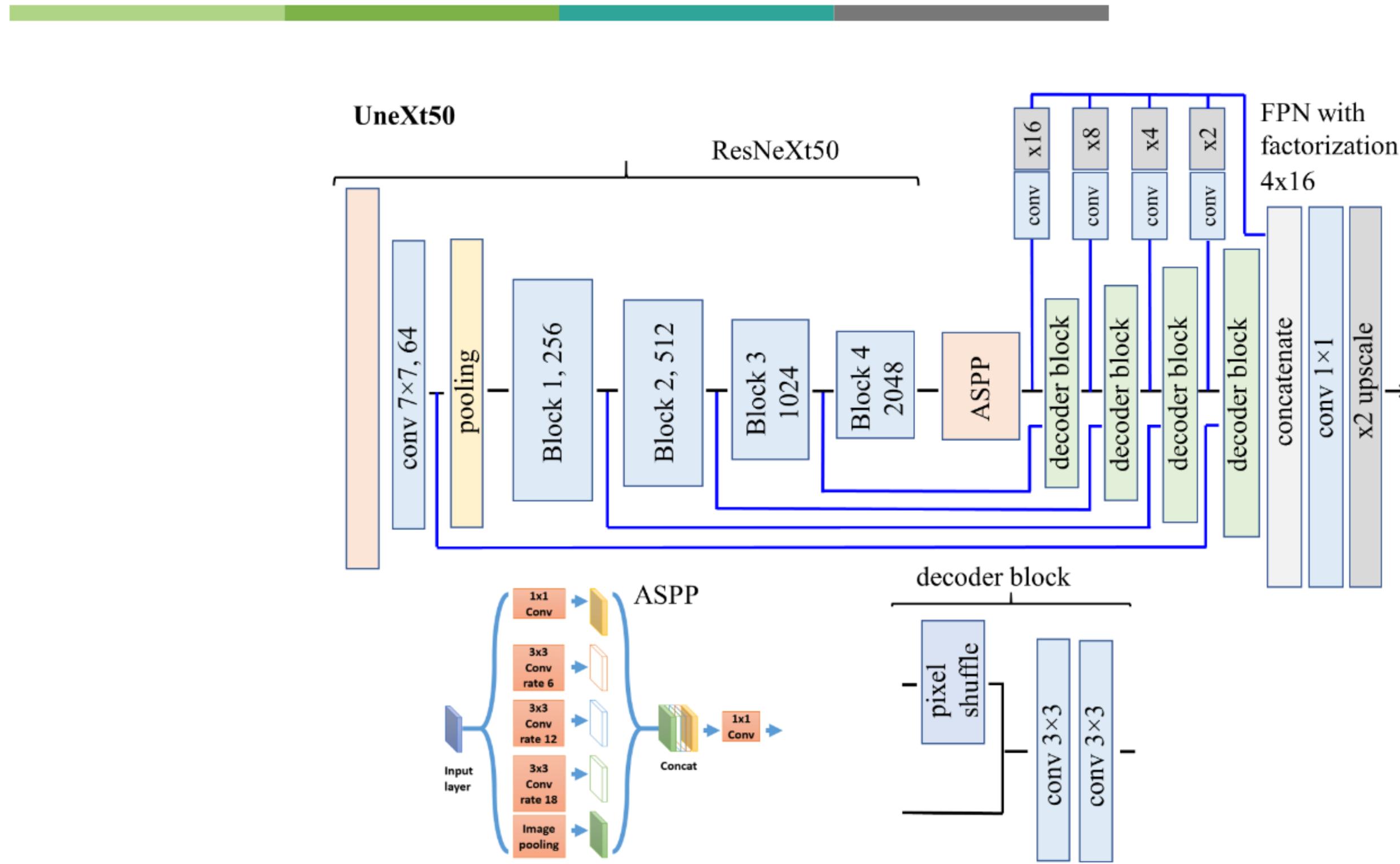
- Train data와 Test data에 대한 설명

This competition uses data from two different consortia, the [Human Protein Atlas](#) (HPA) and [Human BioMolecular Atlas Program](#) (HuBMAP). The training dataset consists of data from public HPA data, the public test set is a combination of private HPA data and HuBMAP data, and the private test set contains only HuBMAP data. Adapting models to function properly when presented with data that was prepared using a different protocol will be one of the core challenges of this competition. While this is expected to make the problem more difficult, developing models that generalize is a key goal of this endeavor.

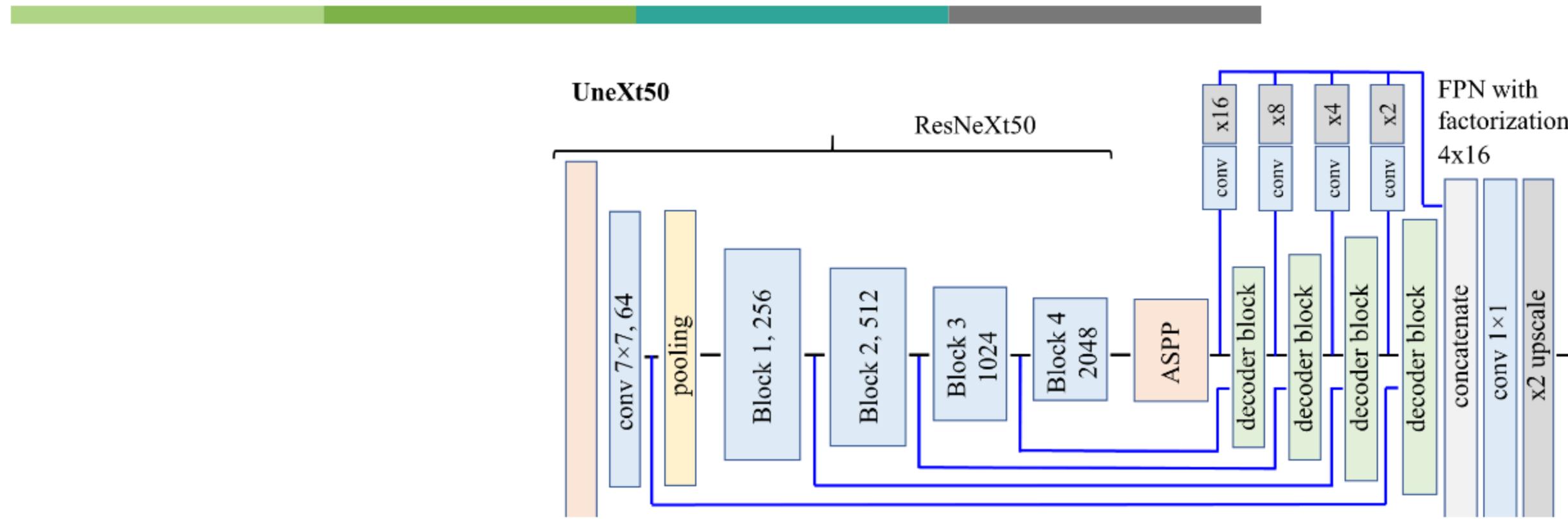
[train/test]_images/ The images. Expect roughly 550 images in the hidden test set. All HPA images are 3000×3000 pixels with a tissue area within the image around 2500×2500 pixels. The HuBMAP images range in size from 4500×4500 down to 160×160 pixels. HPA samples were stained with antibodies visualized with 3,3'-diaminobenzidine (DAB) and counterstained with hematoxylin. HuBMAP images were prepared using Periodic acid-Schiff (PAS)/hematoxylin and eosin (H&E) stains. All images used have at least one FTU. All tissue data used in this competition is from healthy donors that pathologists identified as pathologically unremarkable tissue.

- 2가지 유형의 data로 구성되어 있는데 특이한 점은 **Train data는 HPA로만, Test data는 HPA+HuBMAP data**로 구성되어 있다는 점이다. 두 data는 이미지의 크기도 다르고 염색 방법에도 차이가 있어서 이렇게 train data와 test data에 차이가 있을때 어떻게 모델의 성능을 올릴 것인지가 이 대회의 핵심요소라고 설명하고 있다.

BASE MODEL - UNET

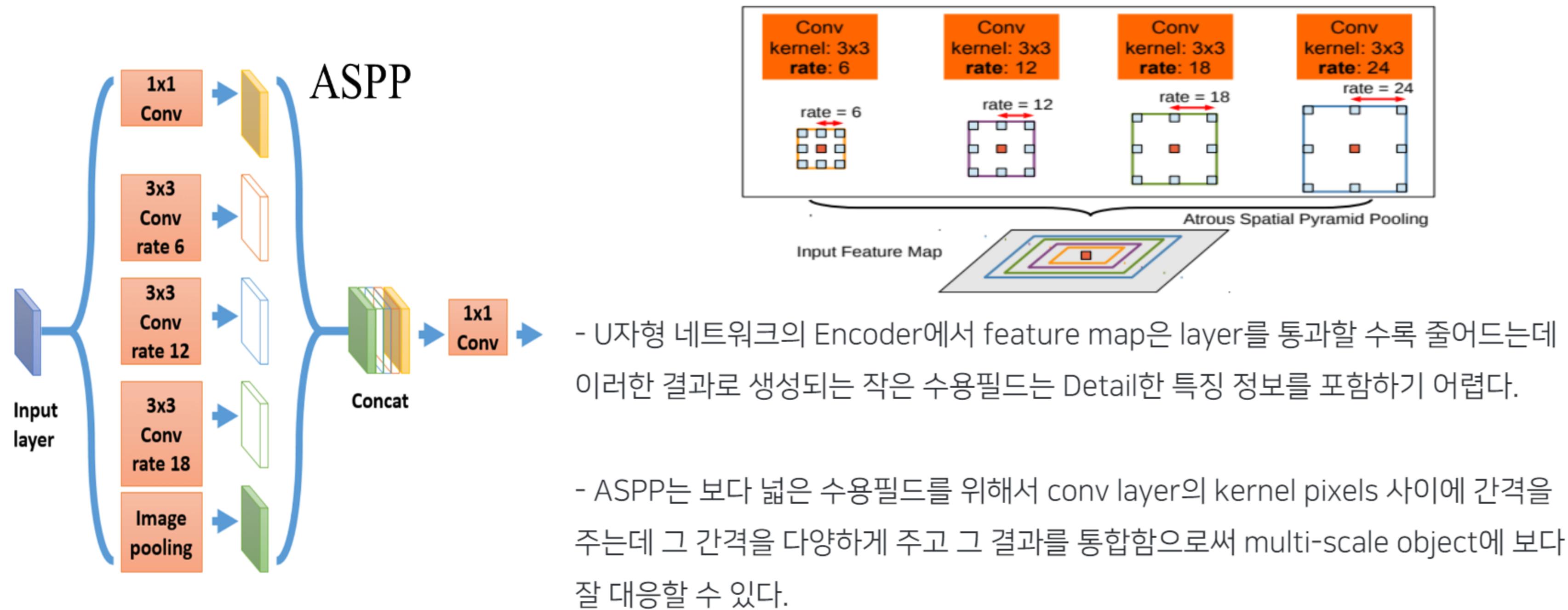


BASE MODEL - UNET



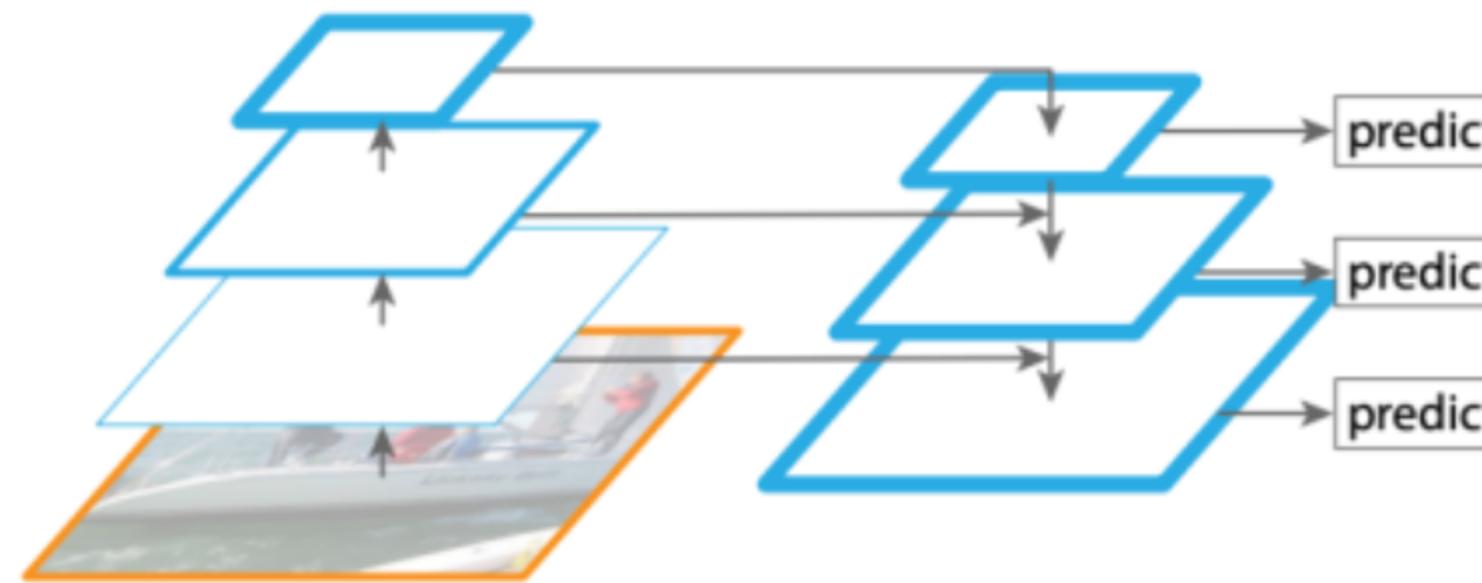
- 의료 이미지에 많이 사용되는 모델 구조로 Encoder 부분에서 다양한 수준에서의 feature map을 추출하고 Decoder 부분에서 추출된 feature map들을 결합해 segmentation mask를 예측한다.
- Encoder와 Decoder 사이의 Skip connection을 통해서 효율적으로 인코더의 중간 layer들의 feature를 통합할 수 있다.

BASE MODEL - ASPP

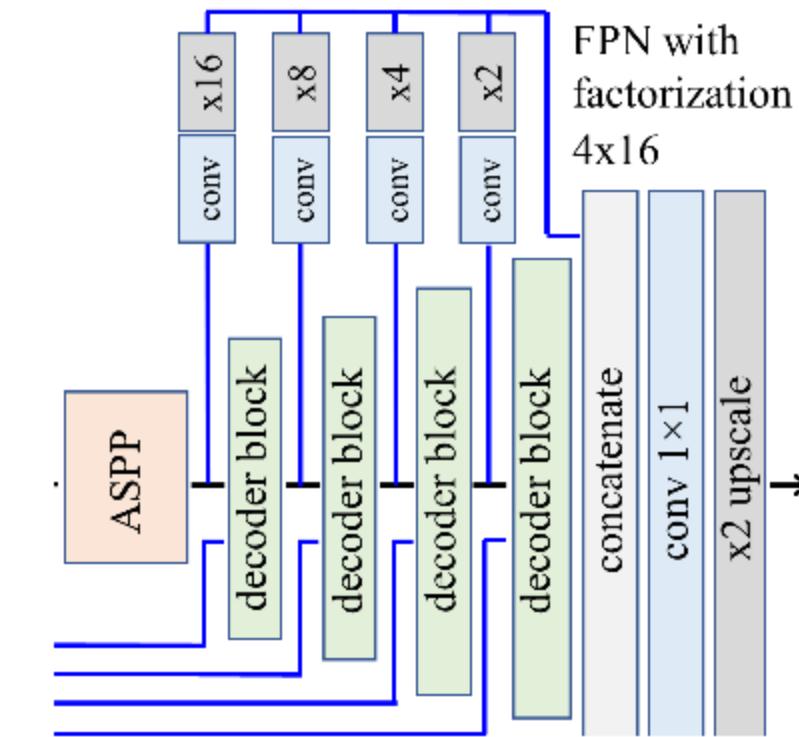


- U자형 네트워크의 Encoder에서 feature map은 layer를 통과할 수록 줄어드는데 이러한 결과로 생성되는 작은 수용필드는 Detail한 특징 정보를 포함하기 어렵다.
- ASPP는 보다 넓은 수용필드를 위해서 conv layer의 kernel pixels 사이에 간격을 주는데 그 간격을 다양하게 주고 그 결과를 통합함으로써 multi-scale object에 보다 잘 대응할 수 있다.

BASE MODEL - FPN

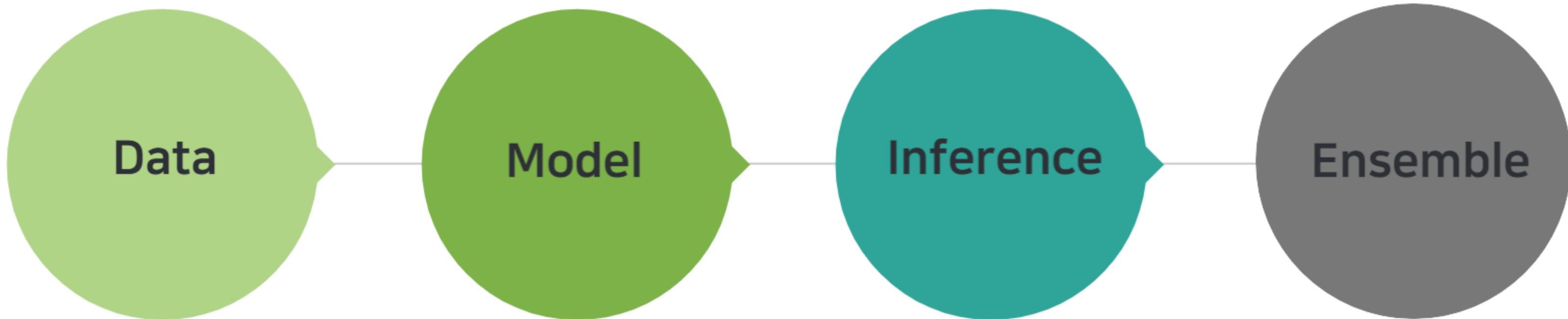


(d) Feature Pyramid Network



- FPN은 top-down pathway와 lateral connection을 사용해 고해상도와 저해상도의 feature map을 결합함으로써 다양한 scale의 정보를 이용할 수 있다.
- FPN으로 각 단계의 decoder block과 출력 계층을 skip connection으로 연결함으로써 모델의 성능과 수렴 속도를 향상시킬 수 있다.

WORKFLOW

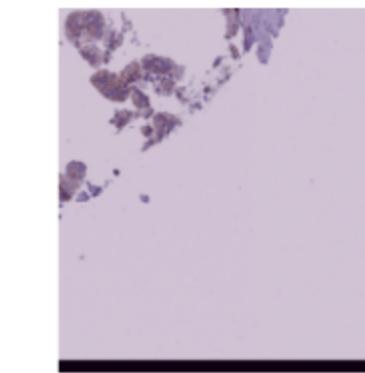
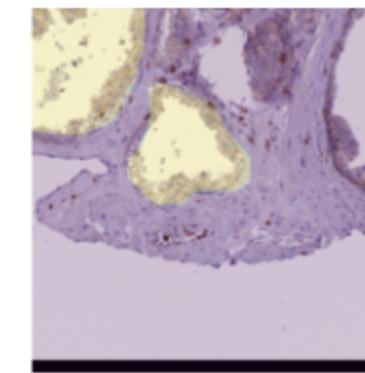
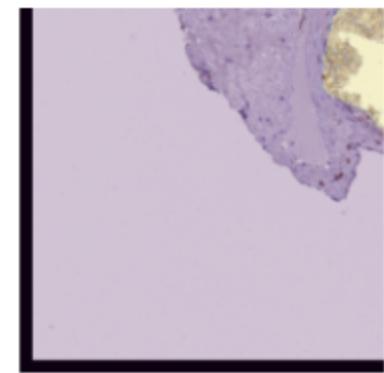
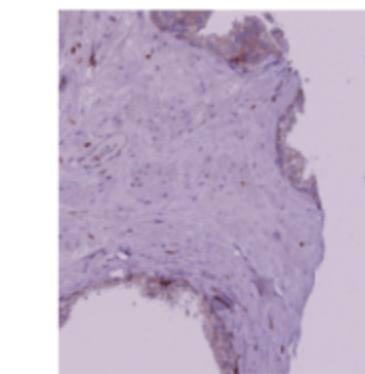
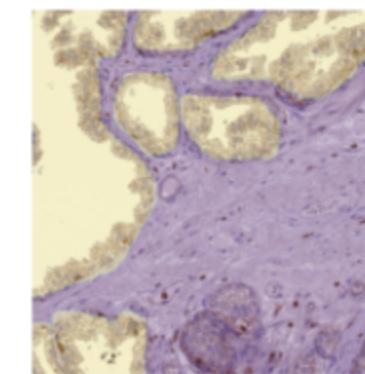
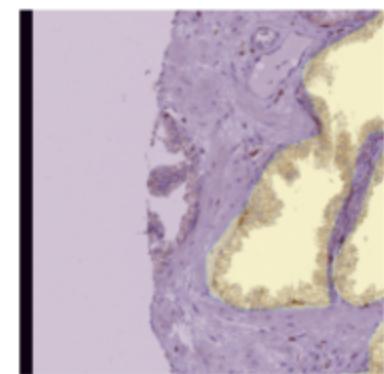
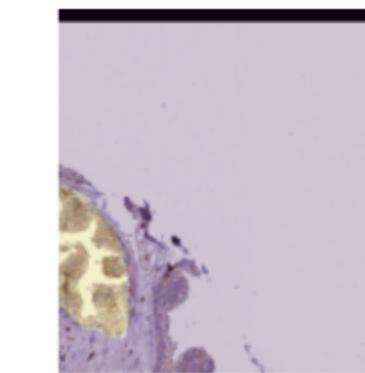
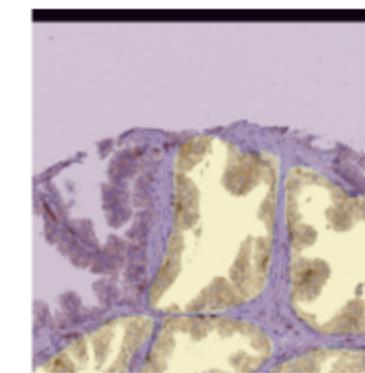
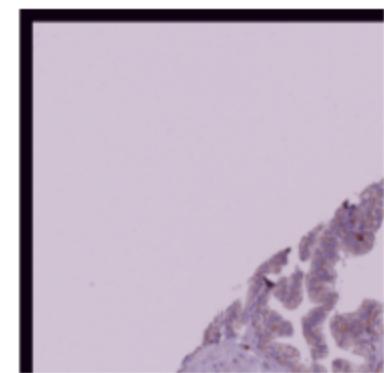


DATA



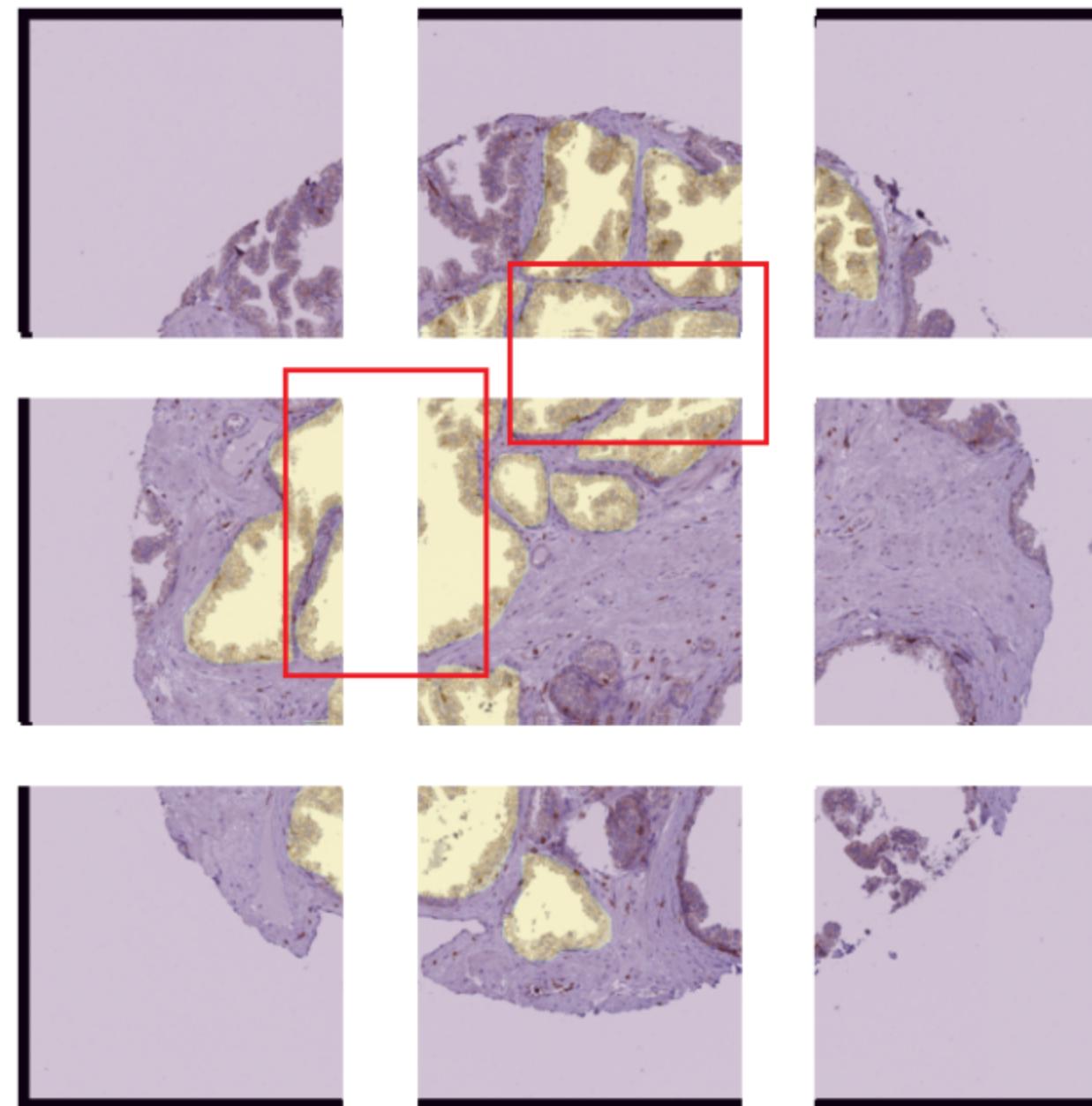
- 01 Multi-scale tile Dataset
- 02 LPB (Laplacian Pyramid Blending)
- 03 Transform

MULTI-SCALE TILE DATASET



- 이미지의 사이즈가 3000*3000으로 너무 커서 여러개의 tile로 잘라서 학습하는 방식을 사용했다.
- 하지만 이 경우 label 부분이 잘리게 되면 정보의 소실이 발생할 수 있다.
- 이와 같은 문제를 보완하기 위해서 tile 구성을 여러가지 수준에서 진행 했다. (전체, 4등분, 9등분 구성으로 함께 학습)

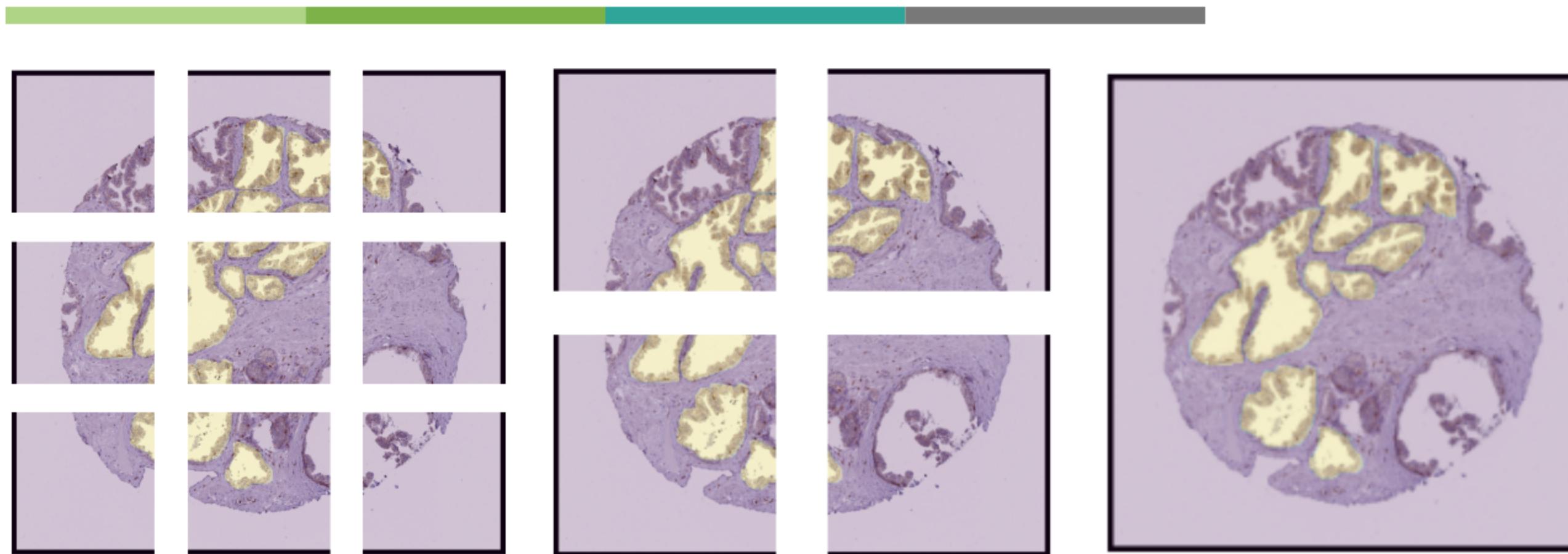
MULTI-SCALE TILE DATASET



- 이미지의 사이즈가 3000*3000으로 너무 커서 여러개의 tile로 잘라서 학습하는 방식을 사용했다.
- 하지만 이 경우 label 부분이 잘리게 되면 정보의 소실이 발생할 수 있다.
- 이와 같은 문제를 보완하기 위해서 tile 구성을 여러가지 수준에서 진행 했다. (전체, 4등분, 9등분 구성으로 함께 학습)

3. 수행 절차 및 방법

MULTI-SCALE TILE DATASET



[Inference] - FastAI Baseline
resnest101-256-4 (version 5/24)

6 days ago by [jisung403](#)

Notebook [Inference] - FastAI Baseline | Version 5

Succeeded 0.72

[Inference] - FastAI Baseline
resnest101-256-multi-scale (version 6/24)

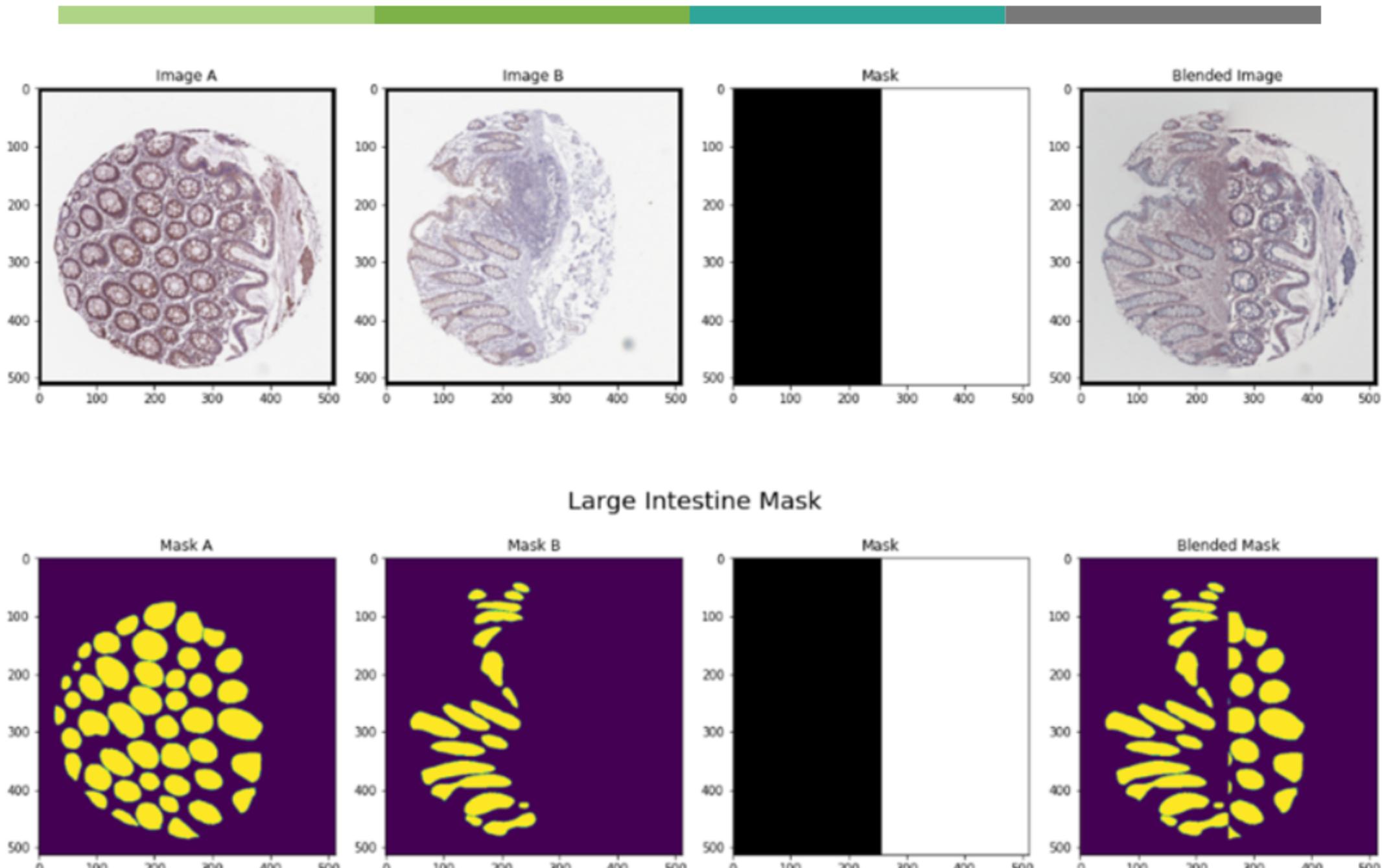
6 days ago by [jisung403](#)

Notebook [Inference] - FastAI Baseline | Version 6

Succeeded 0.74

3. 수행 절차 및 방법

LAPLACIAN PYRAMID BLENDING



- 같은 기관 내에서 서로 다른 이미지 2장을 각각 반반씩 혼합하여 새로운 데이터 생성
- 단순히 이미지를 반반씩 붙이는 것이 아닌 색상 요소도 혼합되어 데이터가 생성된다.
- train data와 test data 사이에 염색 방법의 차이가 있어 이러한 데이터 증식 방법이 효과가 있을 것이라고 생각한다.

3. 수행 절차 및 방법

LAPLACIAN PYRAMID BLENDING



원본 데이터만 사용(256size, EffB7)

[Ucode HuBMAP \(EfficientNet \) Submission](#)

b7-v2-3-1 (version 23/32)

a month ago by Jiseong Ok

[Notebook Ucode HuBMAP \(EfficientNet \) Submission | Version 23](#)

Succeeded

0.71

원본 데이터+ LPB데이터 사용(256size, EffB7)

[Ucode HuBMAP \(EfficientNet \) Submission](#)

lpb-v1 (version 25/32)

25 days ago by Jiseong Ok

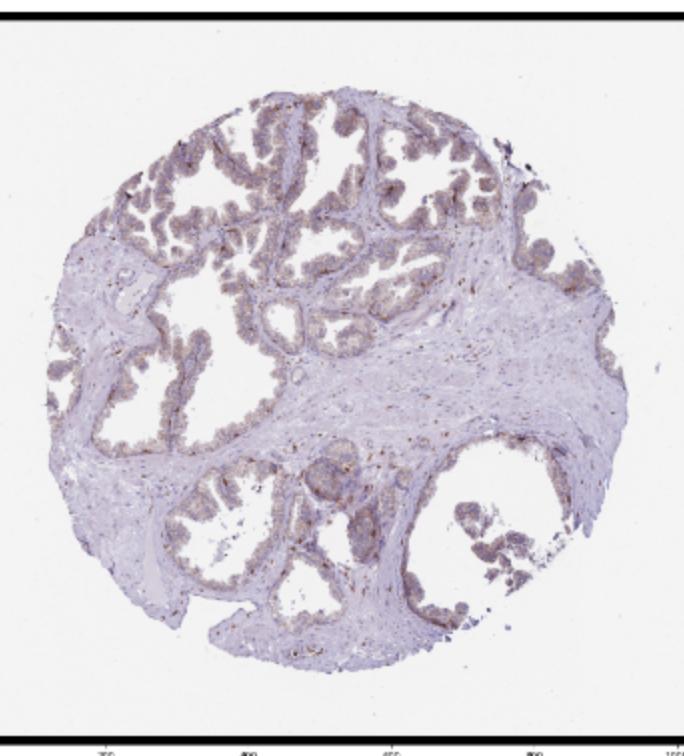
[Notebook Ucode HuBMAP \(EfficientNet \) Submission | Version 25](#)

Succeeded

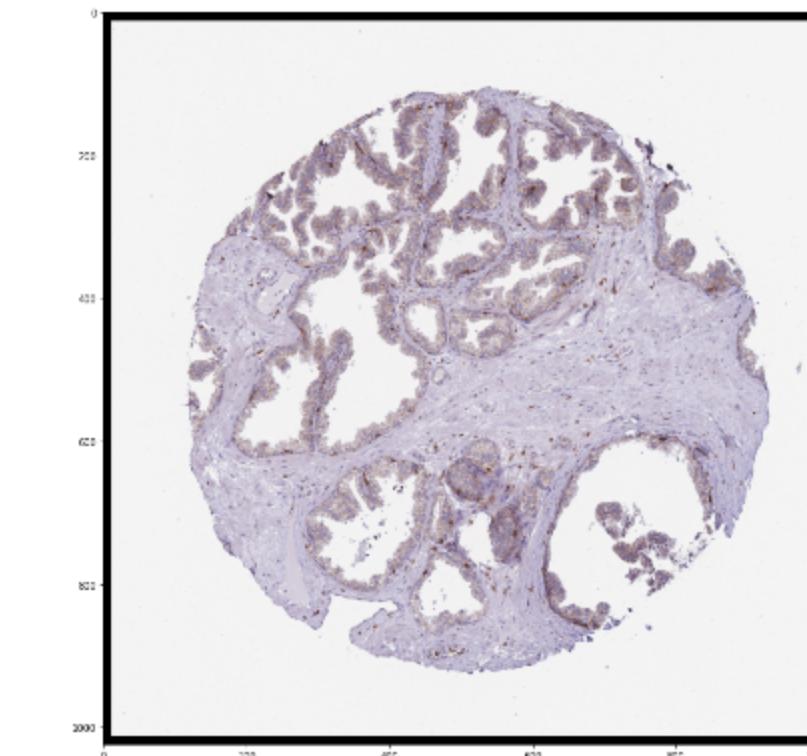
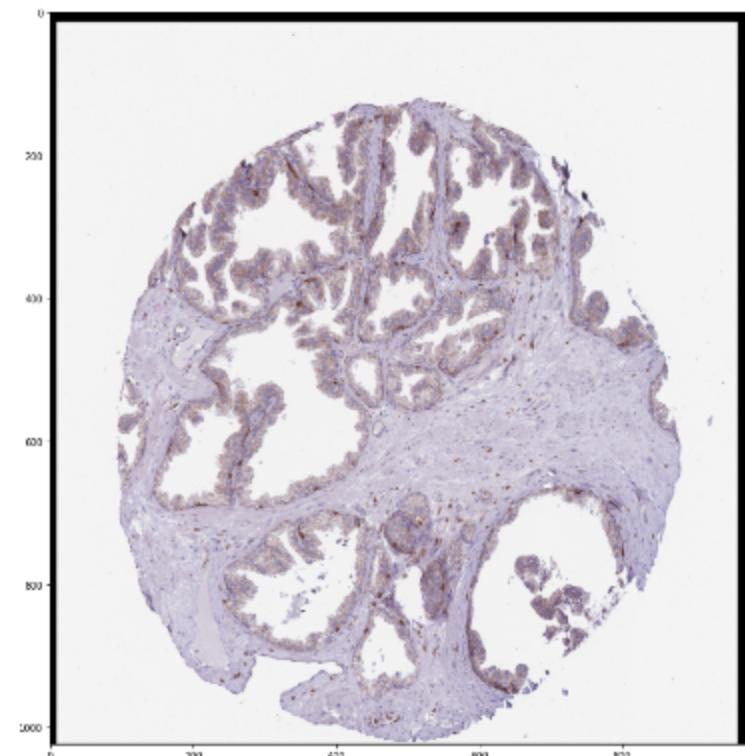
0.73

3. 수행 절차 및 방법

TRANSFORM



GridDistortion



HueSaturationValue

- Transform은 Filp, RandomRotate, ShiftScaleRotate, GaussNoise, GaussianBlur, RandomBrightnessContrast, HueSaturationValue, OpticalDistortion, GridDistortion, PiecewiseAffine 총 10가지를 사용했다.

3. 수행 절차 및 방법

TRANSFORM



Flip, Rotate만 사용

[Ucode HuBMAP \(EfficientNet \) Submission](#)

B7-v4 (version 3/32)

a month ago by [Jiseong Ok](#)

[Notebook Ucode HuBMAP \(EfficientNet \) Submission | Version 3](#)

Succeeded

0.55

10가지 모두 사용

[Ucode HuBMAP \(EfficientNet \) Submission](#)

B7_v2 (version 2/32)

a month ago by [Jiseong Ok](#)

[Notebook Ucode HuBMAP \(EfficientNet \) Submission | Version 2](#)

Succeeded

0.67

MODEL



01

Mmsegmentation - multi class

- Upernet-Convnext-xl

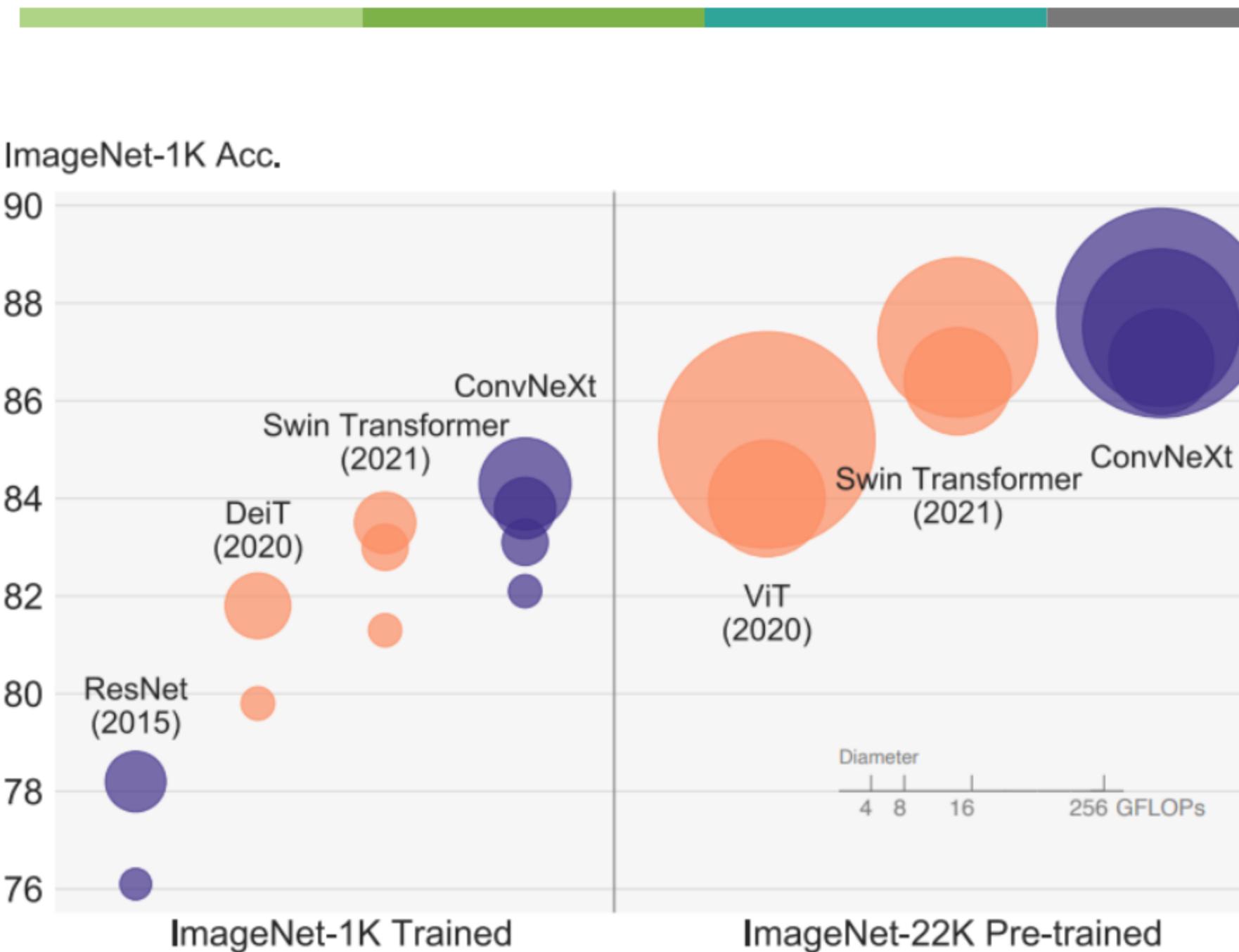
02

Fast.ai - binary class

- Unet-EfficientNetB7
- Unet-Resnest269

3. 수행 절차 및 방법

MODEL - Mmsegmentation (multi class)



```
norm_cfg = dict(type='SyncBN', requires_grad=True)
custom_imports = dict(imports='mmcls.models', allow_failed_imports=False)
checkpoint_file = 'https://download.openmmlab.com/mmclassification/v0/convnext/downstream/convnext-base_'
model = dict(
    type='EncoderDecoder',
    pretrained=None,
    backbone=dict(
        type='mmcls.ConvNeXt',
        arch='base',
        out_indices=[0, 1, 2, 3],
        drop_path_rate=0.4,
        layer_scale_init_value=1.0,
        gap_before_final_norm=False,
        init_cfg=dict(
            type='Pretrained', checkpoint=checkpoint_file,
            prefix='backbone.')),
    decode_head=dict(
        type='UPerHead',
        in_channels=[128, 256, 512, 1024],
        in_index=[0, 1, 2, 3],
        pool_scales=(1, 2, 3, 6),
        channels=512,
        dropout_ratio=0.1,
        num_classes=19,
        norm_cfg=norm_cfg,
        align_corners=False,
        loss_decode=dict(
            type='CrossEntropyLoss', use_sigmoid=False, loss_weight=1.0)),
    auxiliary_head=dict(
        type='FCNHead',
        in_channels=384,
```

3. 수행 절차 및 방법

MODEL - Mmsegmentation (multi class)



512size, SGD

[mmsegmentation LB0.78\[Inference\]\(1/5folds\)](#)

upernet_convnext_512_SGD_10epoch (version 7/16)

8 days ago by [bsnJaehyun](#)

Notebook mmsegmentation LB0.78[Inference](1/5folds) | Version 7

Succeeded

0.52

512size, AdamW

[mmsegmentation LB0.78\[Inference\]\(1/5folds\)](#)

512_change_convnext (version 14/16)

a day ago by [bsnJaehyun](#)

Succeeded

0.61

512size, multi class weight

[mmsegmentation LB0.78\[Inference\]\(1/5folds\)](#)

512_upernet_convnext_xlarge_50 (version 6/16)

9 days ago by [bsnJaehyun](#)

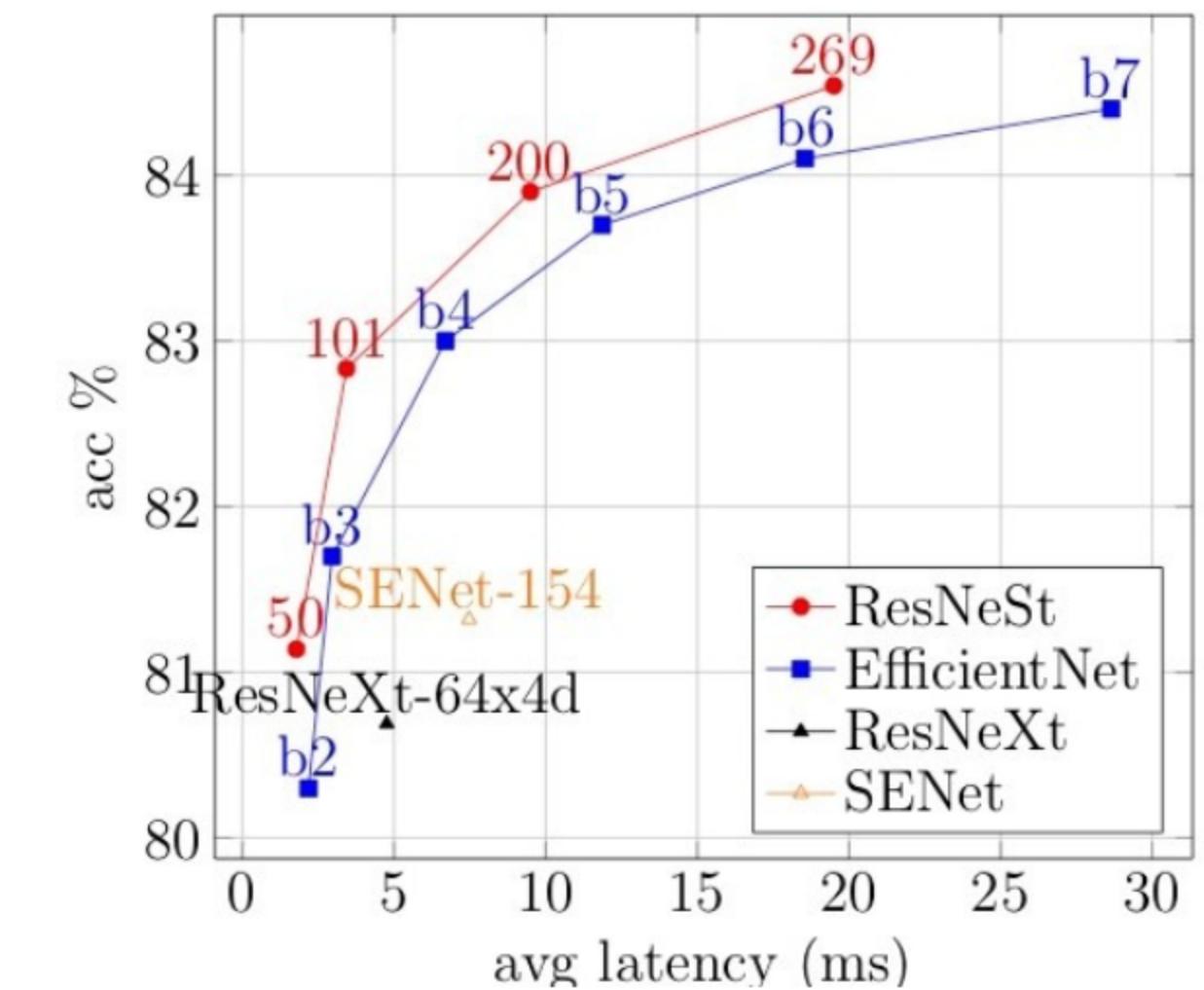
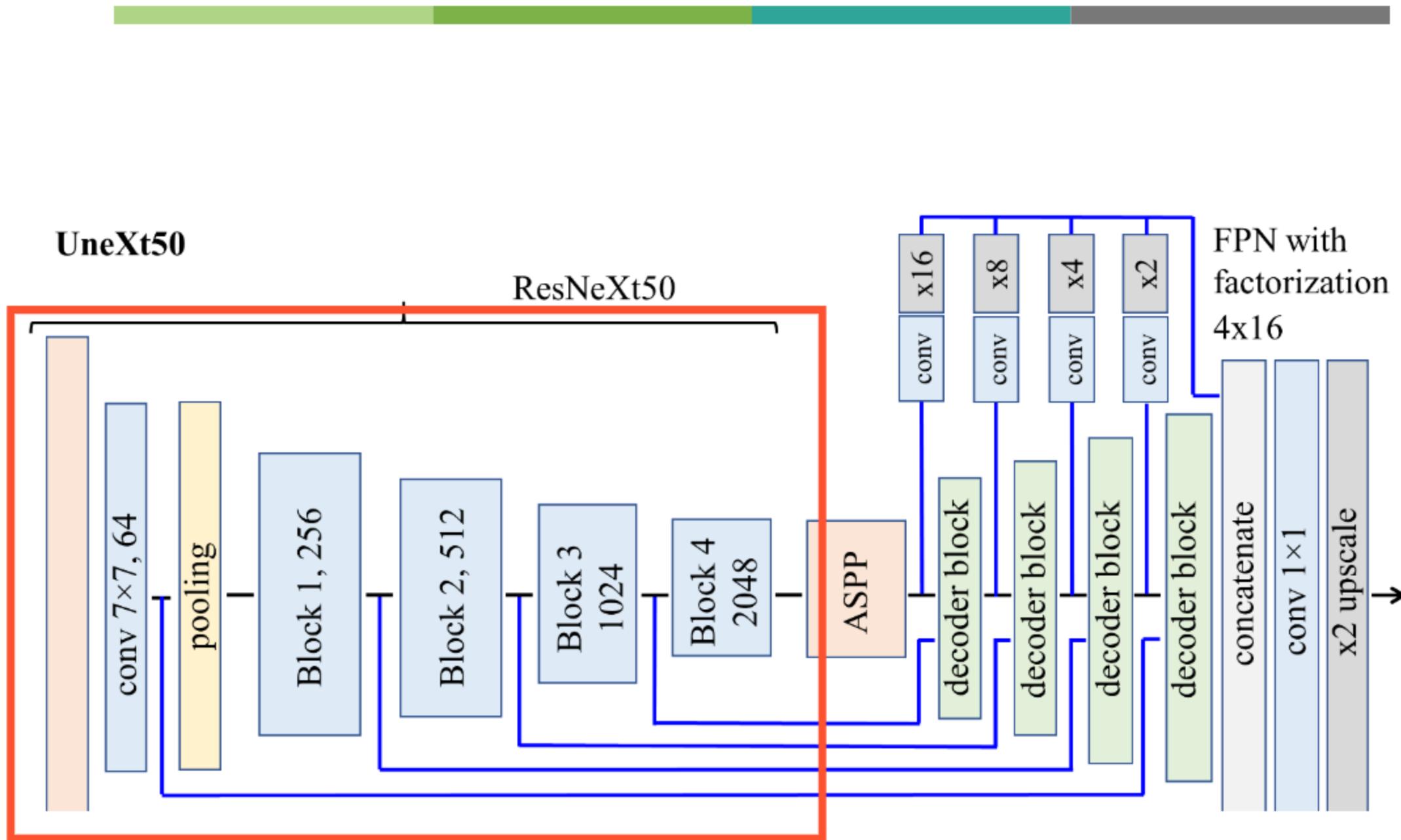
Notebook mmsegmentation LB0.78[Inference](1/5folds) | Version 6

Succeeded

0.62

3. 수행 절차 및 방법

MODEL - Fast.ai (binary class)



MODEL

 - Fast.ai (binary class)

Base model에서 backbone을 변경

Backbone	Dataset	Score
ResneXt50	256*4 tile set	0.61
ResneXt101	256*4 tile set	0.7
ResneSt101	256*4 tile set	0.72
EfficientNetB7	256*4 tile set	0.73
ResneSt101	256 multi tile set	0.75
EfficientNetB7	256 multi tile set	0.75
ResneSt269	256 multi tile set	0.76

3. 수행 절차 및 방법

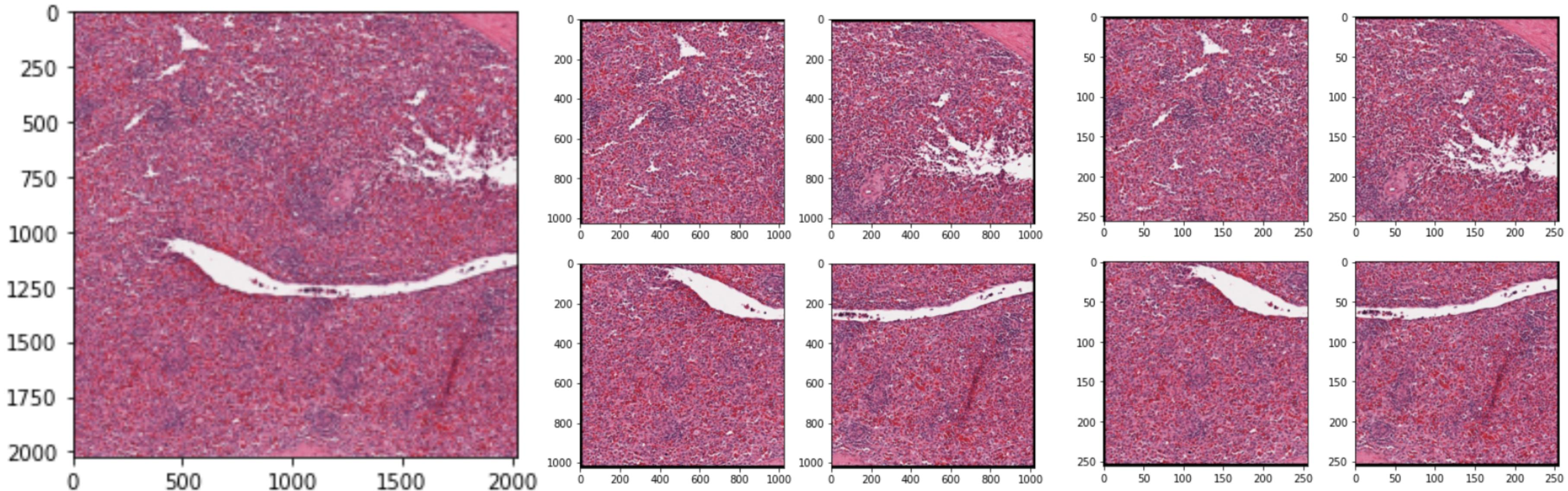
INFERENCE

[train/test]_images/ The images. Expect roughly 550 images in the hidden test set. All HPA images are 3000×3000 pixels with a tissue area within the image around 2500×2500 pixels. The HuBMAP images range in size from 4500×4500 down to 160×160 pixels. HPA samples were stained with antibodies visualized with 3,3'-diaminobenzidine (DAB) and counterstained with hematoxylin. HuBMAP images were prepared using Periodic acid-Schiff (PAS)/hematoxylin and eosin (H&E) stains. All images used have at least one FTU. All tissue data used in this competition is from healthy donors that pathologists identified as pathologically unremarkable tissue.

- Test data set의 이미지가 160*160 부터 4500*4500 까지 매우 다양하다고 명시되어 있었기 때문에 다양한 크기의 이미지를 전처리 하는 과정이 중요했다.
- Mmsegmentation의 경우 inference mode를 whole과 slide mode로 선택할 수 있어 문제가 없었다.
- Fast.ai의 경우 base line에서 제시된 방법은 학습에 사용했던 고정된 타일의 크기를 그대로 적용하는 것이었다. 하지만 이 방법을 그대로 적용시 이미지 크기에 따라 불필요한 패딩이 들어가는 경우가 많이 생기는 것으로 판단 되었고 이를 보완하기 위해 패딩이 최대한 적게 들어가도록 코드를 변경했다.

INFERENCE

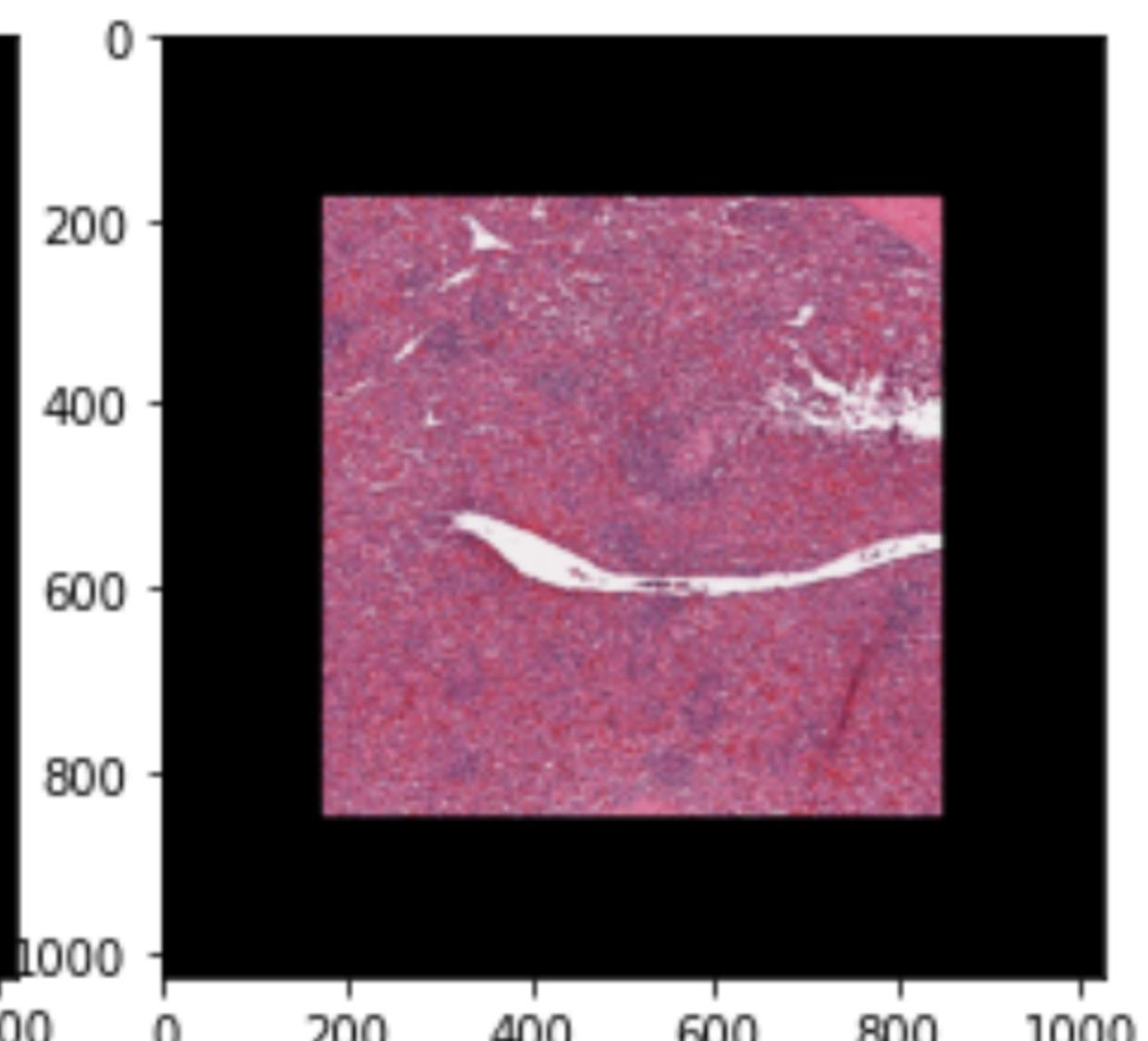
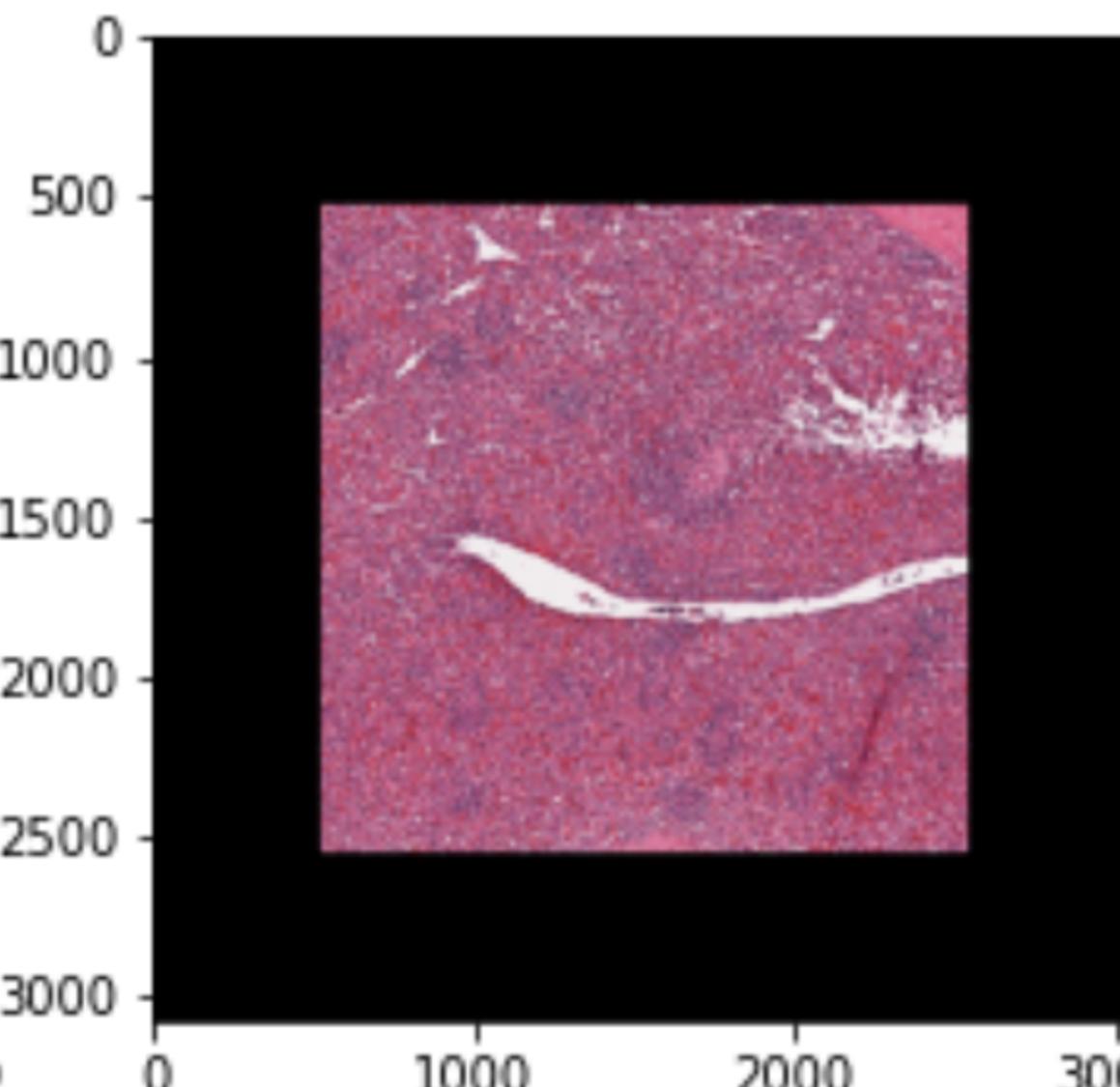
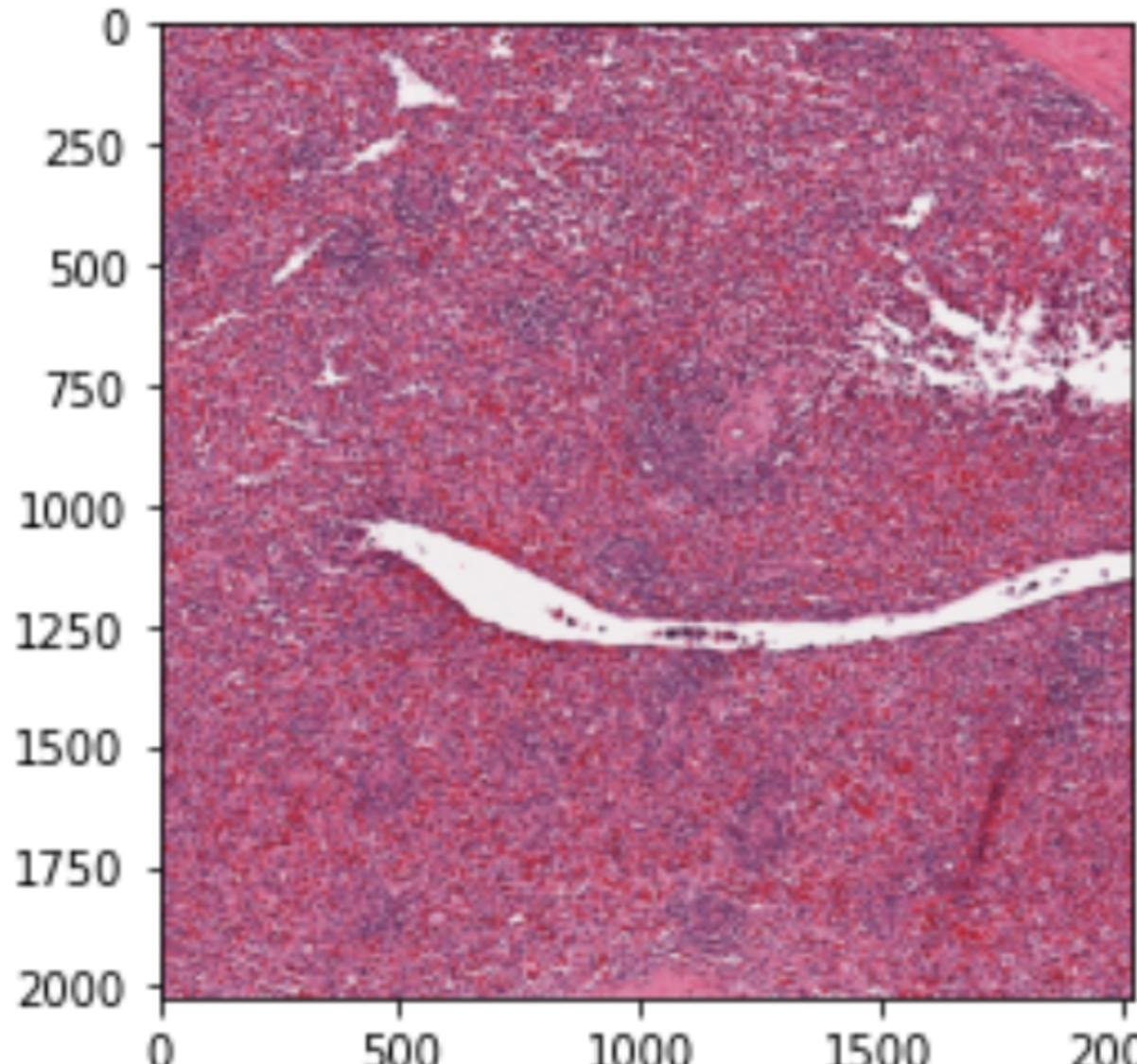
size = 256, reduce = 4로 Train set을 만들고 학습, 이 때 고정된 tile 크기는 $256 \times 4 = 1024$ 가 됨



- 2023*2023의 image를 1024*1024로 자르고(이를 위해 패딩이 25 들어감), 256*256으로 resize하는 과정
- 이 경우 이미지의 크기가 타일의 배수(2048)와 비슷해 패딩이 적게 들어간다.

INFERENCE

size = 1024, reduce = 3로 학습, 동일하게 적용

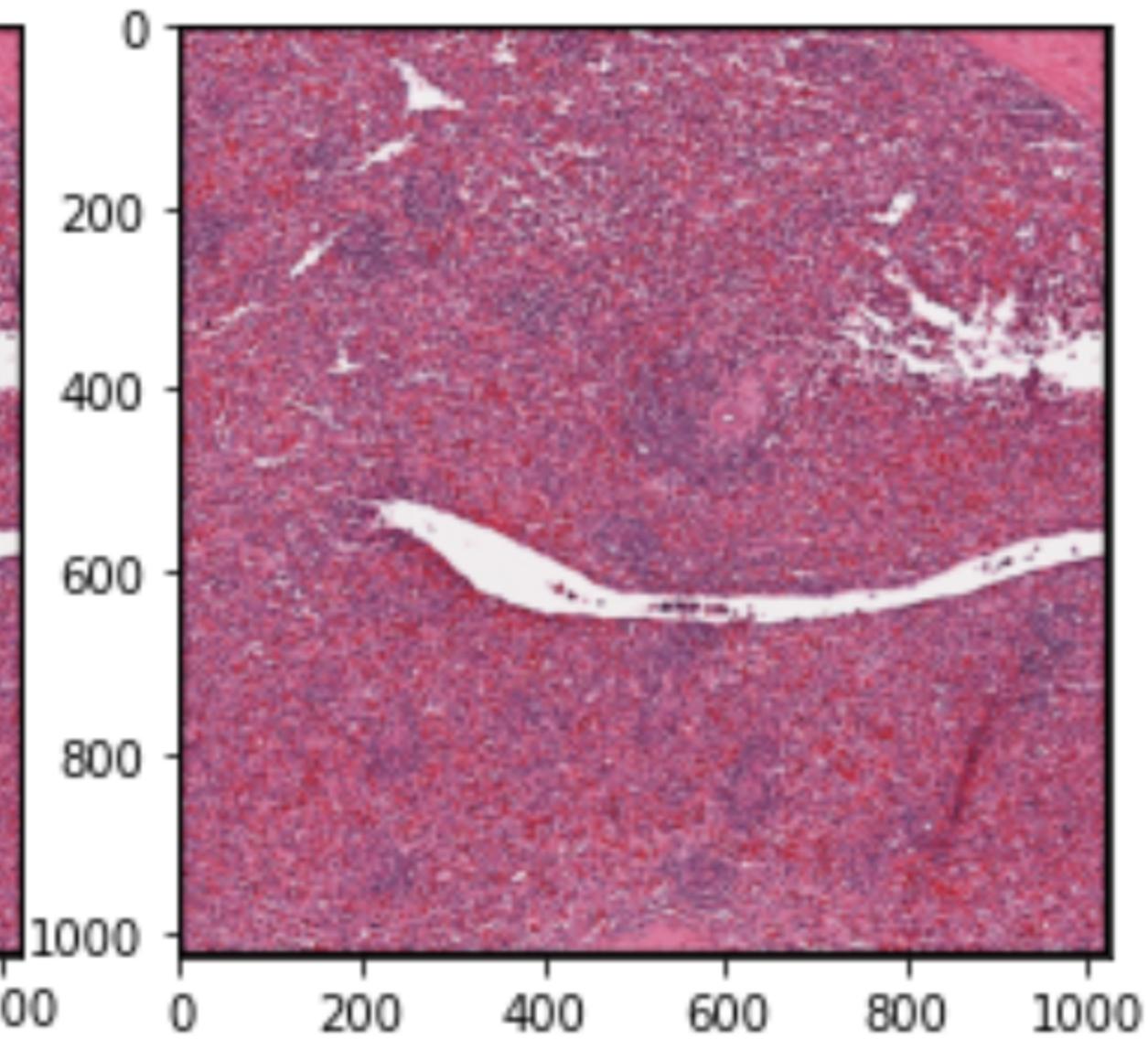
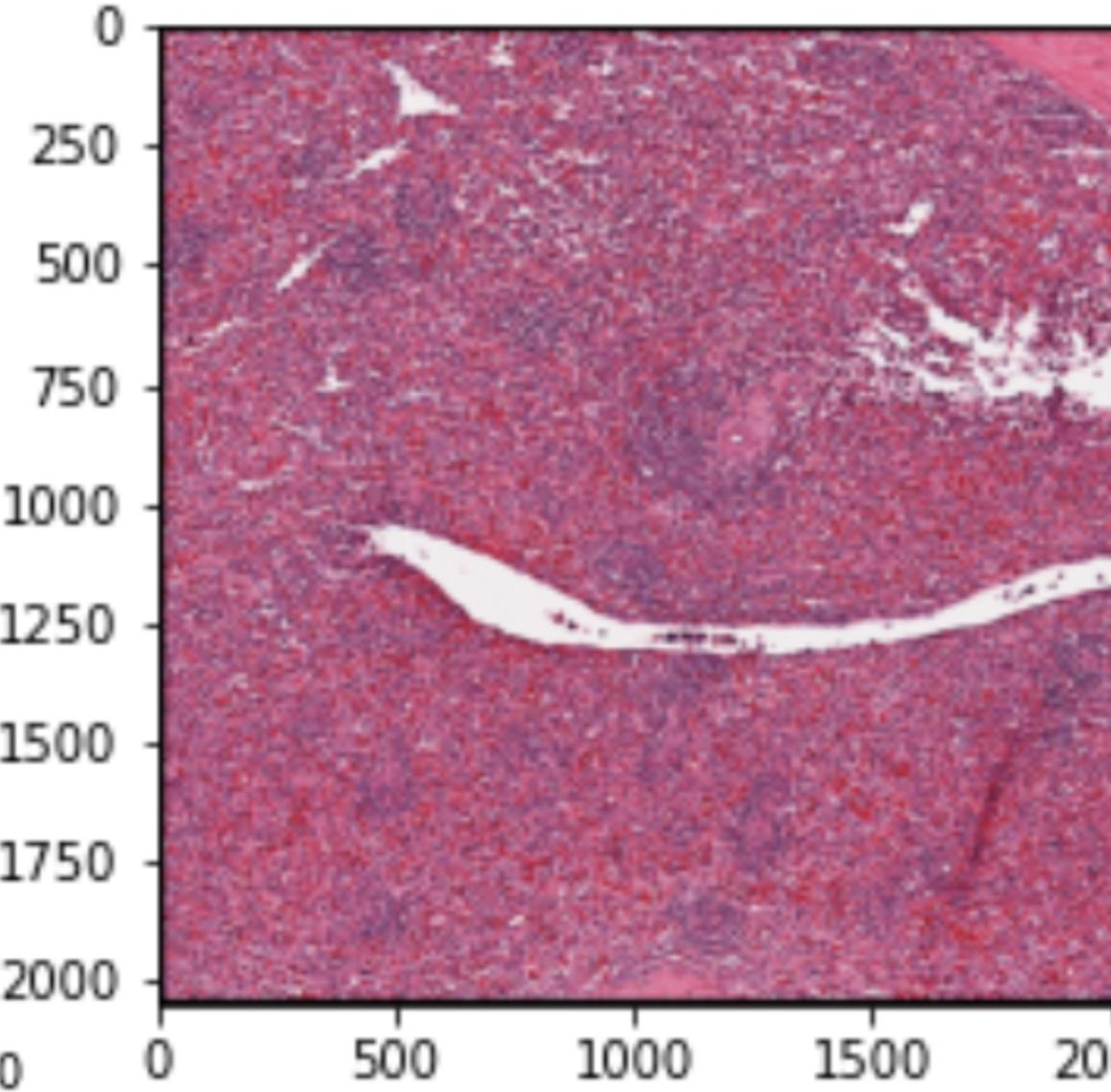
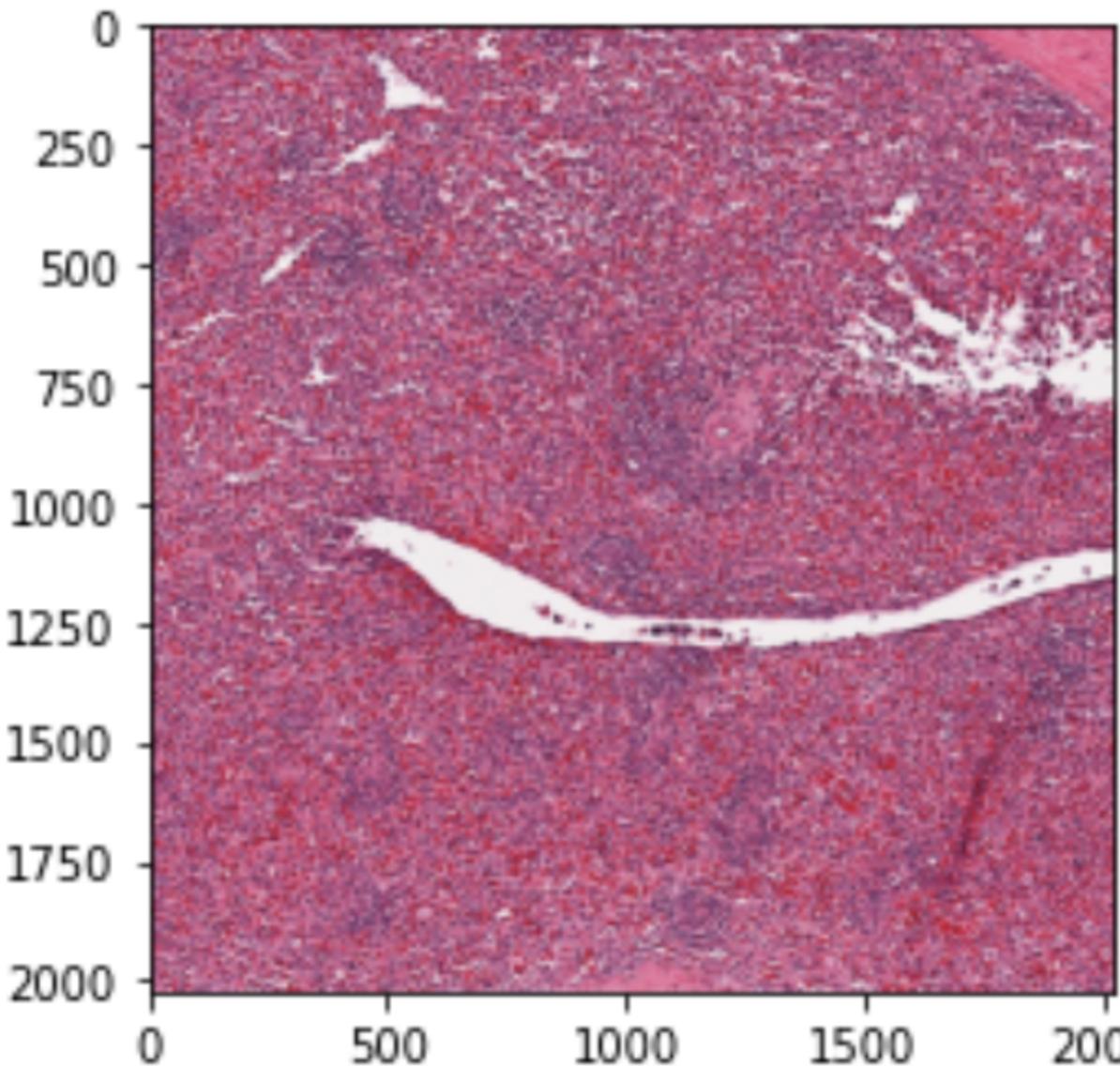


- 하지만 이처럼 고정된 타일 크기(3072)와 test image(2023)이 크기가 맞지 않는 경우 불필요한 패딩이 많이 들어가게 된다.

INFERENCE

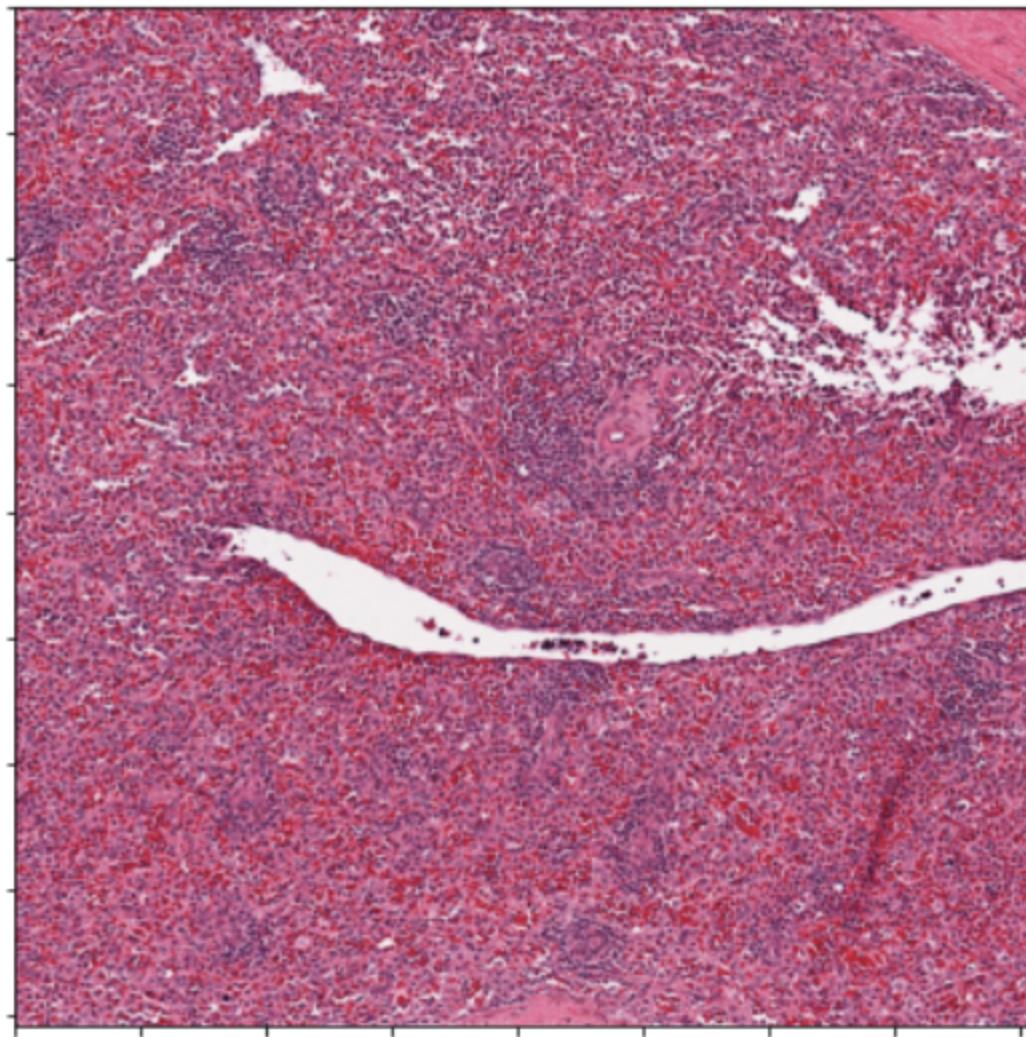
size = 1024, reduce = 3로 학습 -> image의 크기와 학습 size의 ratio를 통해 최적의 reduce값 적용

- reduce = $\text{math.ceil}(\text{shape}/\text{size})$ 이용, 이 경우에는 reduce가 2로 변경된다.

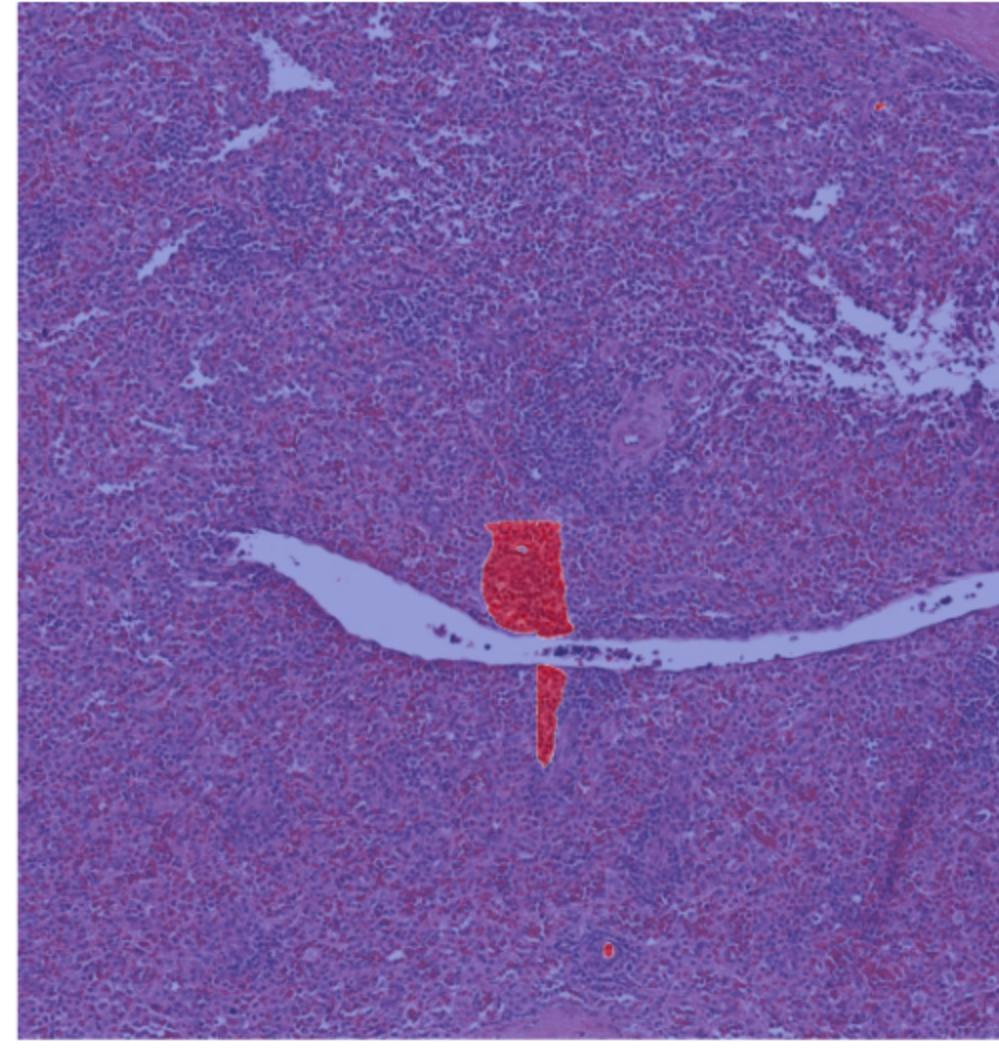


INFERENCE

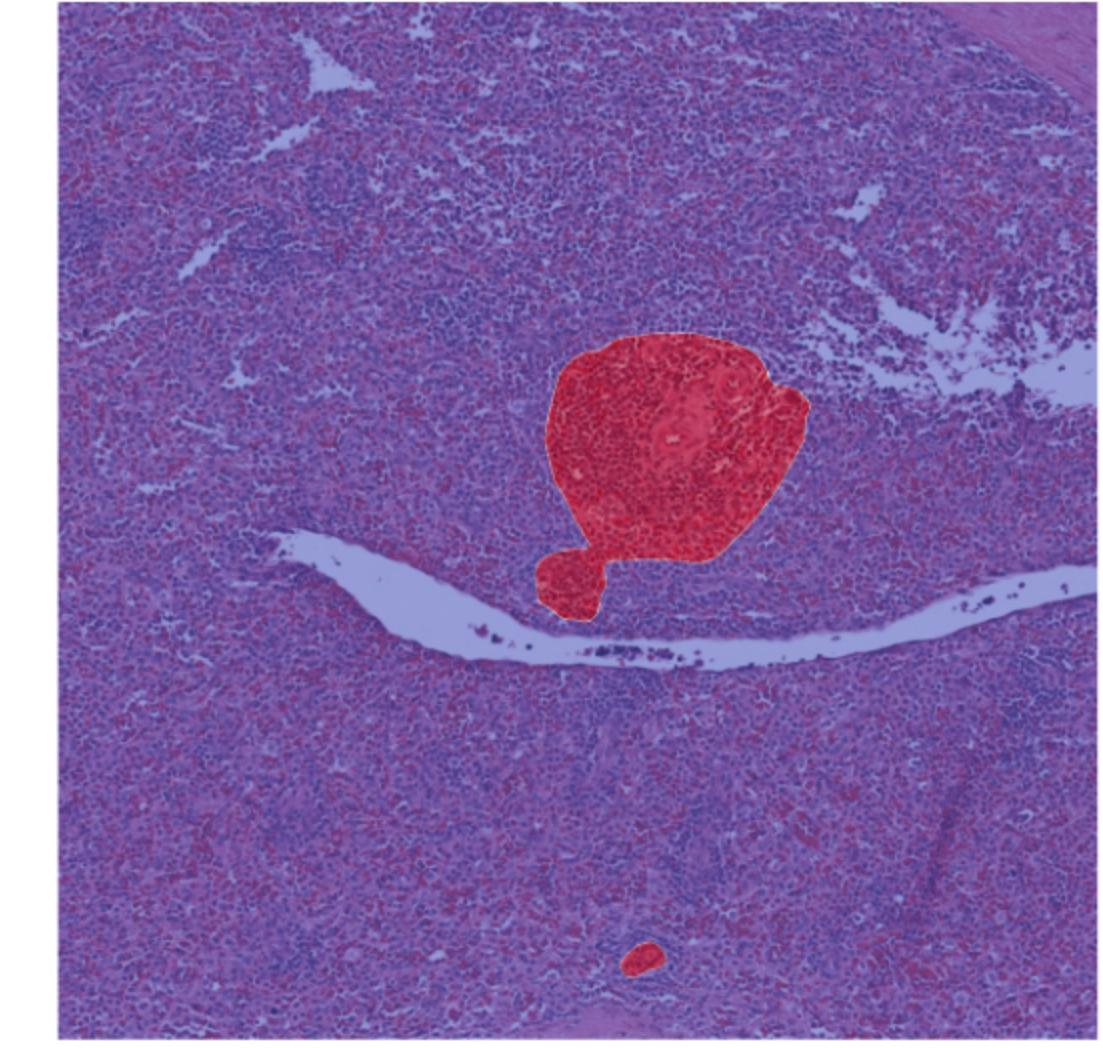
Unet-EfficientNetb4, size=1024, reduce=3으로 학습한 model



Test Image



Pred Image (Score : 0.48)

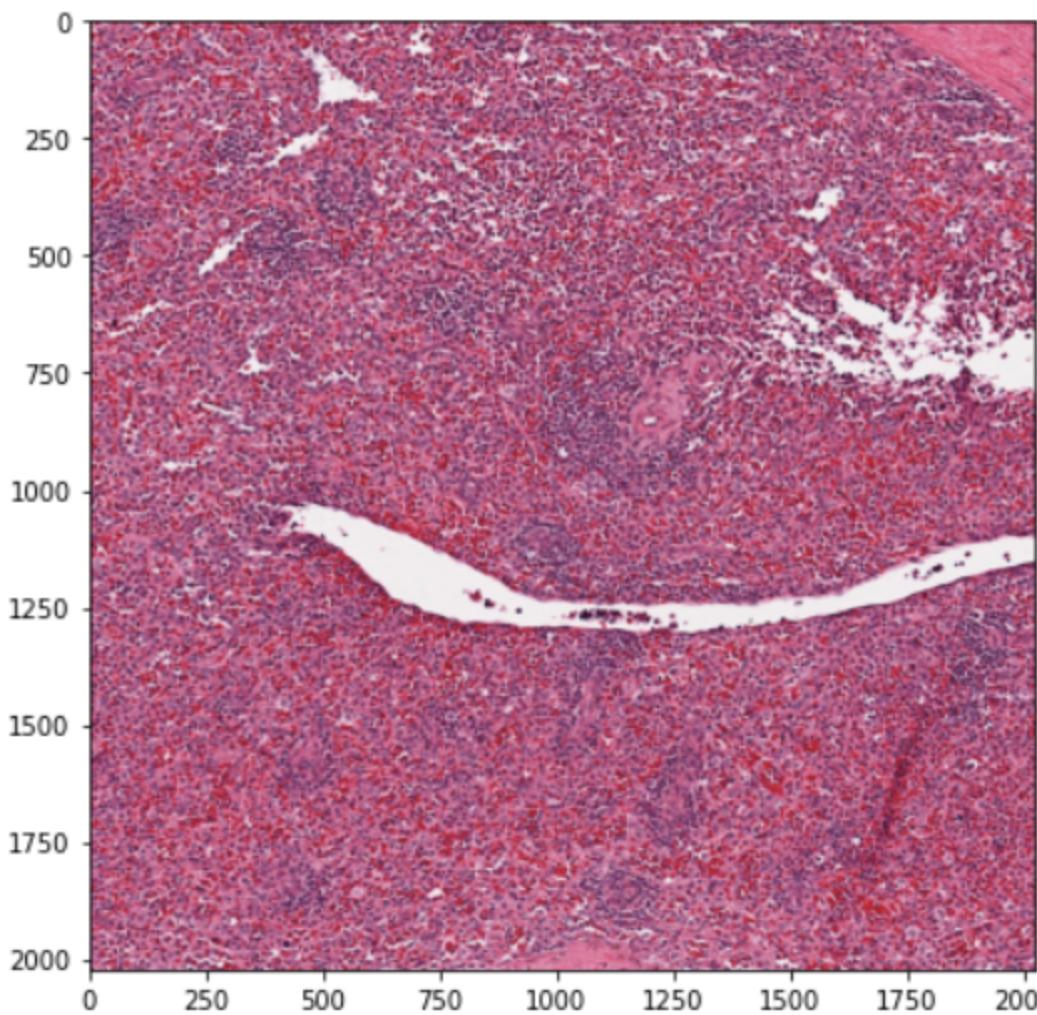


Pred Image (Score : 0.67)

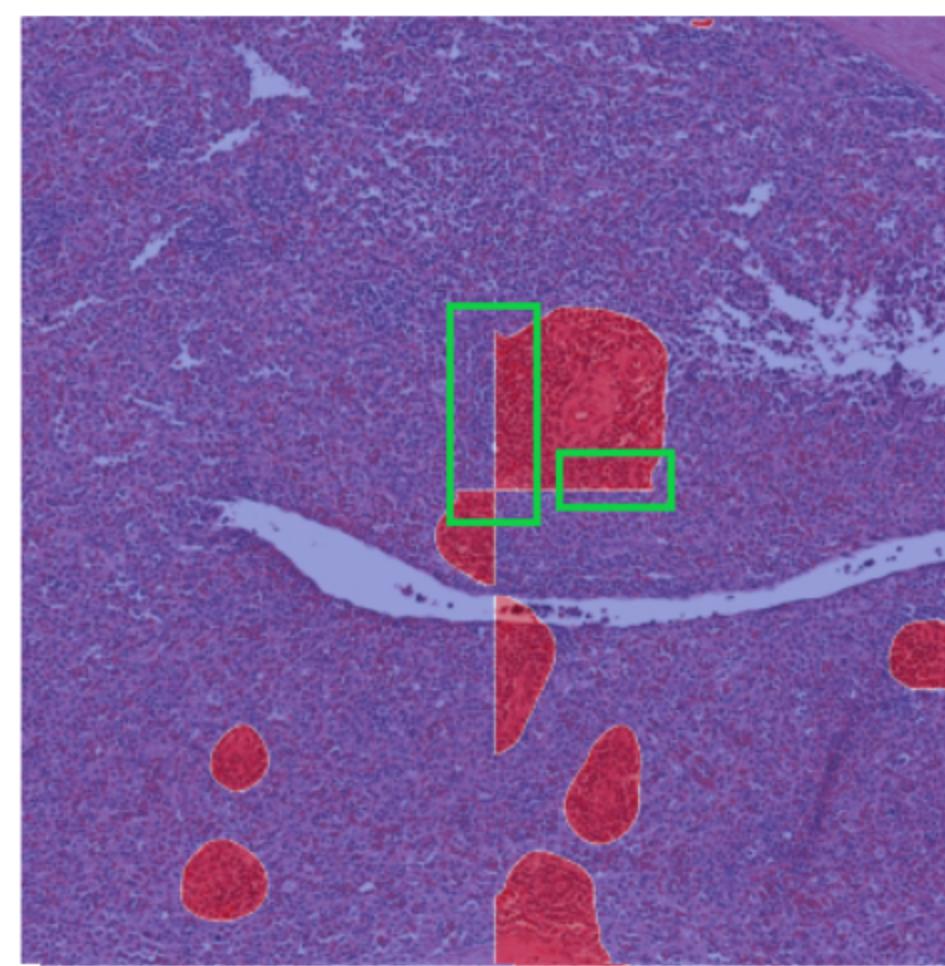
- 이미지 사이즈 최적화를 통해 성능 개선(0.48 -> 0.67)

3. 수행 절차 및 방법

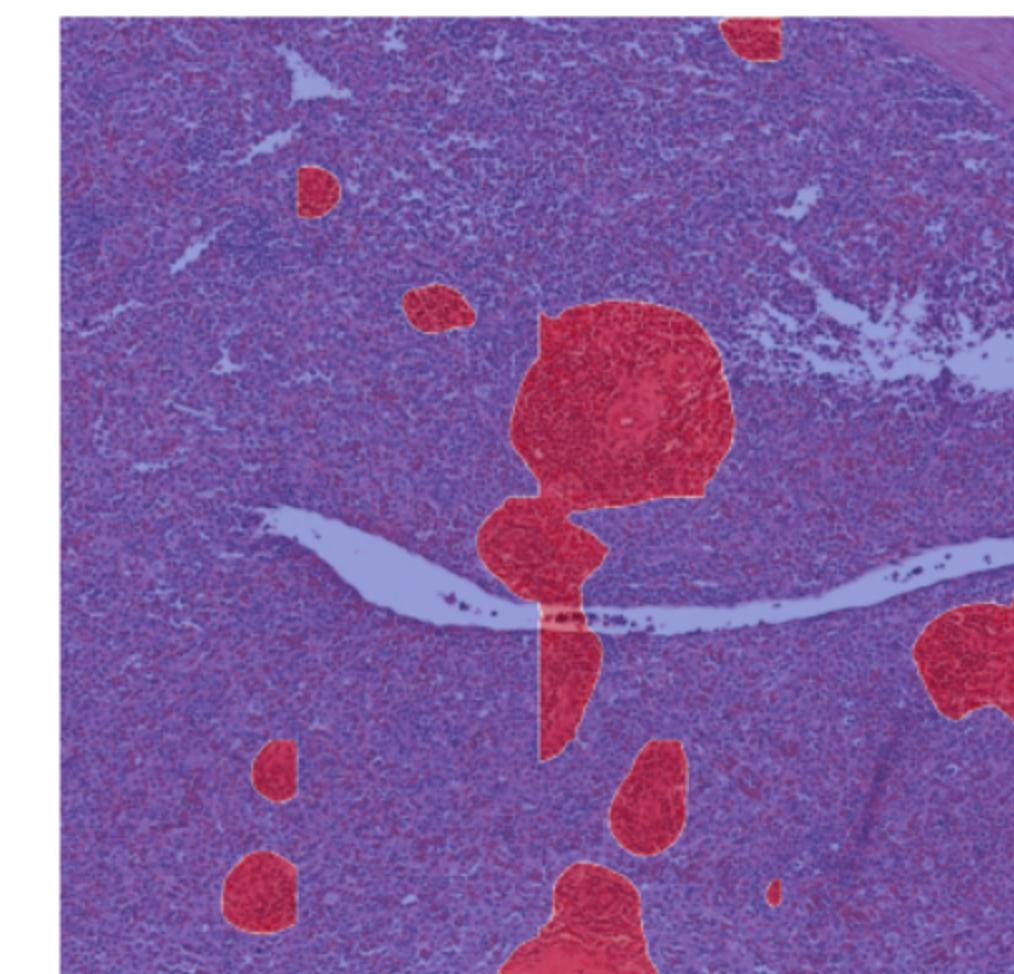
INFERENCE



Test Image



Pred Image (Score : 0.76)

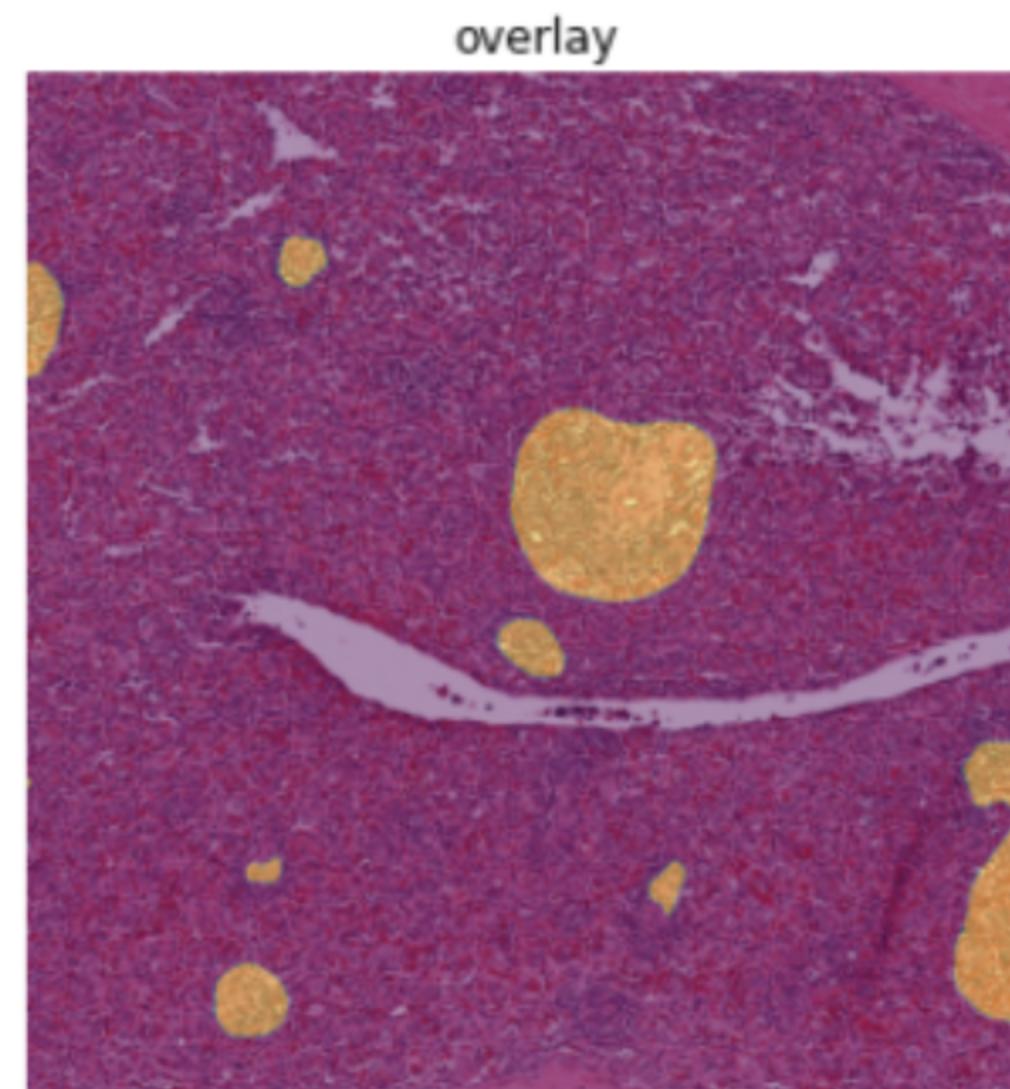


Pred Image (Score : 0.77)

- 타일로 잘라서 예측하는 경우 타일의 경계선에서 예측이 잘 되지 않는 문제 발생
- stride를 통한 slide 예측을 통해 경계면에서의 segmentation 개선($0.76 \rightarrow 0.77$)

INFERENCE

Multi-class 경우 mask가 해당 클래스의 값(1, 2, 3, 4, 5)으로 생성되므로 이를 이용해 하나의 이미지에 하나의 클래스 예측값만 선택되도록 처리했다.(0을 제외하고 가장 빈도수가 많은 값으로 클래스 선택)



[$(0, 3829036), (4, 263493)$]

3. 수행 절차 및 방법

ENSEMBLE



1. 모델별로 생성한 마스크 이미지(0,1로 구성)를 가지고 픽셀당 평균값을 구하고 특정 값 이상인 경우만을 선택해 최종 마스크 이미지를 구성

2. 모델별로 생성한 마스크 이미지를 모두 더해 1이상인 픽셀을 선택해 최종 마스크 이미지를 구성(합집합)

4. 수행 결과

BEST SCORE



		Model	Dataset	Ensemble	Score
1		2, 3번의 결과를 양상블(합집합 개념)			0.77
2	mmseg	Upernet-Convnext_xl	768 multi scale tile	X	0.77
3	fast ai	Unet-ResneSt269	256 multi scale tile	X	0.77

68 알파코 부트캠프 2기 - "코드득"   0.77 140 2h

 Your Best Entry!
Your most recent submission scored 0.77, which is the same as your previous score. Keep trying!

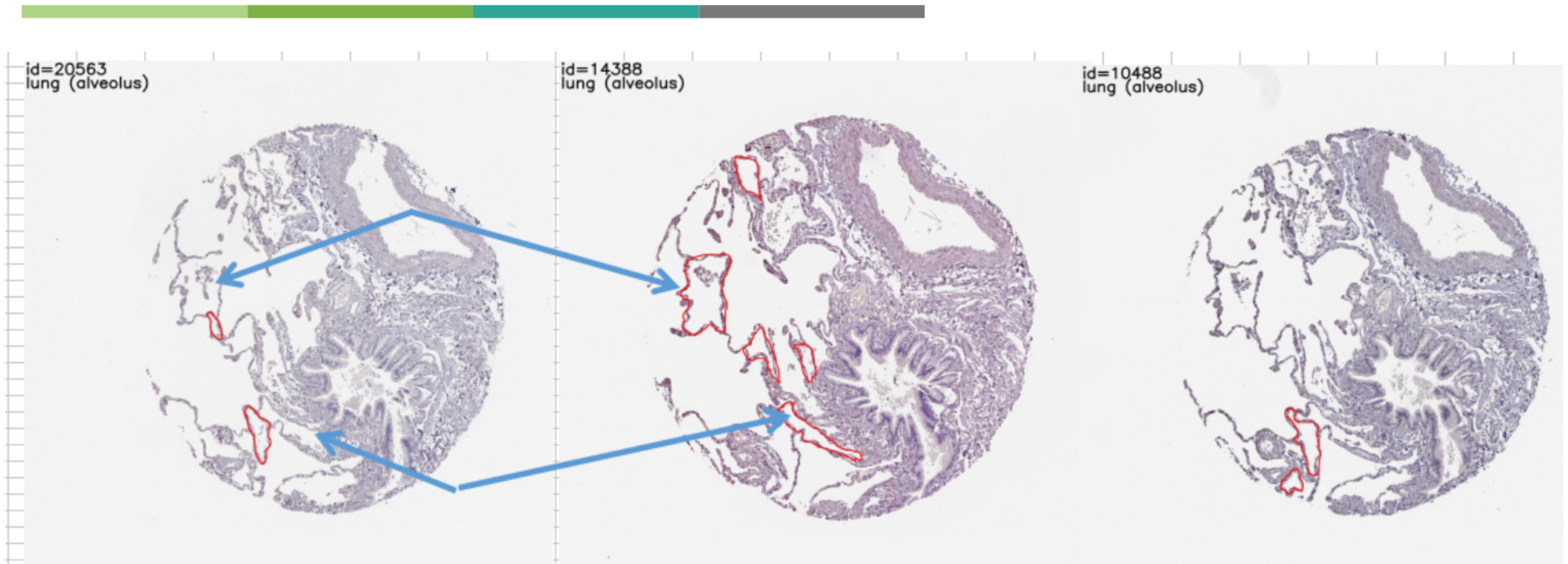
DISCUSSION

- 진행했던 4가지 작업 중에서 Data 에서의 작업이 성능 개선에 가장 큰 영향을 주었다. 전체 데이터가 35 1개로 적고 사이즈가 큰 고해상도 이미지이기 때문에 이 이미지를 학습 데이터로 가공하는 단계에서의 작업이 성능에 큰 영향을 주었다고 생각한다.

Class	Dice	Acc
background	99.03	99.25
kidney	91.85	88.6
prostate	91.42	92.79
largeintestine	91.40	88.85
spleen	89.29	89.16
lung	26.67	16.36

- 폐(lung)의 경우 multi class 모델에서 다른 기관에 비해 현저히 낮은 dice 점수를 기록했는데 이 또한 data의 문제라고 생각된다.

DISCUSSION



- 폐 데이터의 불일관성이 폐의 낮은 dice 점수에 원인이라고 생각한다.

DISCUSSION



- Computing 자원의 한계로 학습 과정에서 좀 더 큰 이미지를 다뤄보지 못한 것이 아쉽다.
Downsampling을 통해 예측을 하게 되면 mask를 다시 원본 이미지의 크기로 되돌리는 과정에서 오차가 생길 수 있기 때문이다.
- 데이터 측면에서 부족한 데이터를 보완할 수 있는 방법이 부족했던 것 같아서 아쉽다. 또한 해당 Domain에 대한 전문적인 지식이 없어서 데이터를 효과적으로 다룰 수 없었고 test data set이 공개되지 않아서 우리가 한 작업의 결과에 대해서 점수 말고는 얻을 수 있는 정보가 없어서 아쉬웠다.

코드득 (CODE GAIN)

발표를 들어주셔서

감사합니다