HeLP Challenge 2019

Contest 3. Breast Cancer Classification

1st Team – **GoldenPass**

김태우(서강대학교), 최종현(고려대학교), 김대영(고려대학교)

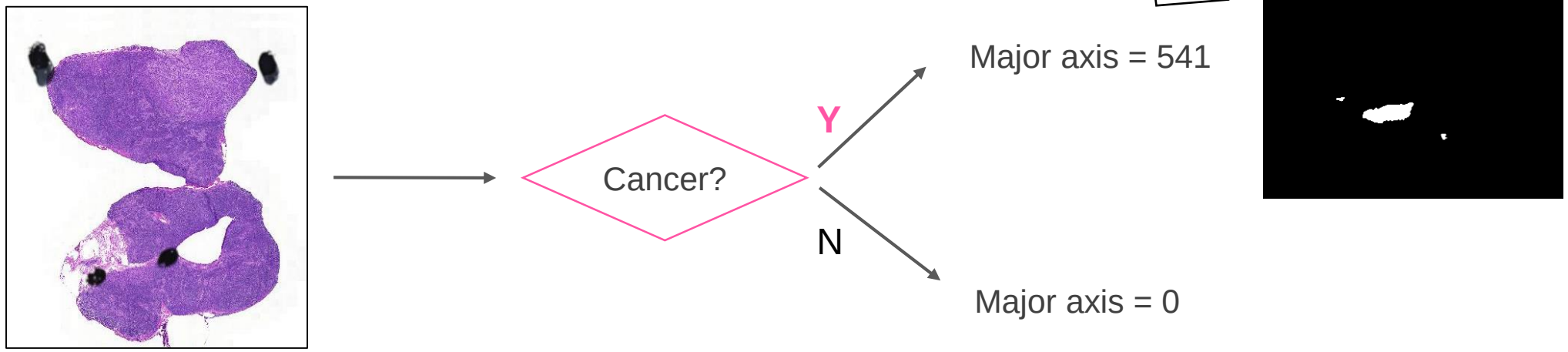
Contents

➢ Breast cancer classification on frozen pathology

① Pathology 슬라이드를 암 또는 정상으로 **분류**하기 (with AUC[Area Under Curve]) → *2018년도와 동일!*

② 암인 슬라이드의 경우, 암인 부분의 **major axis** 계산하기 (with Accuracy) → *2019년도에 추가!*

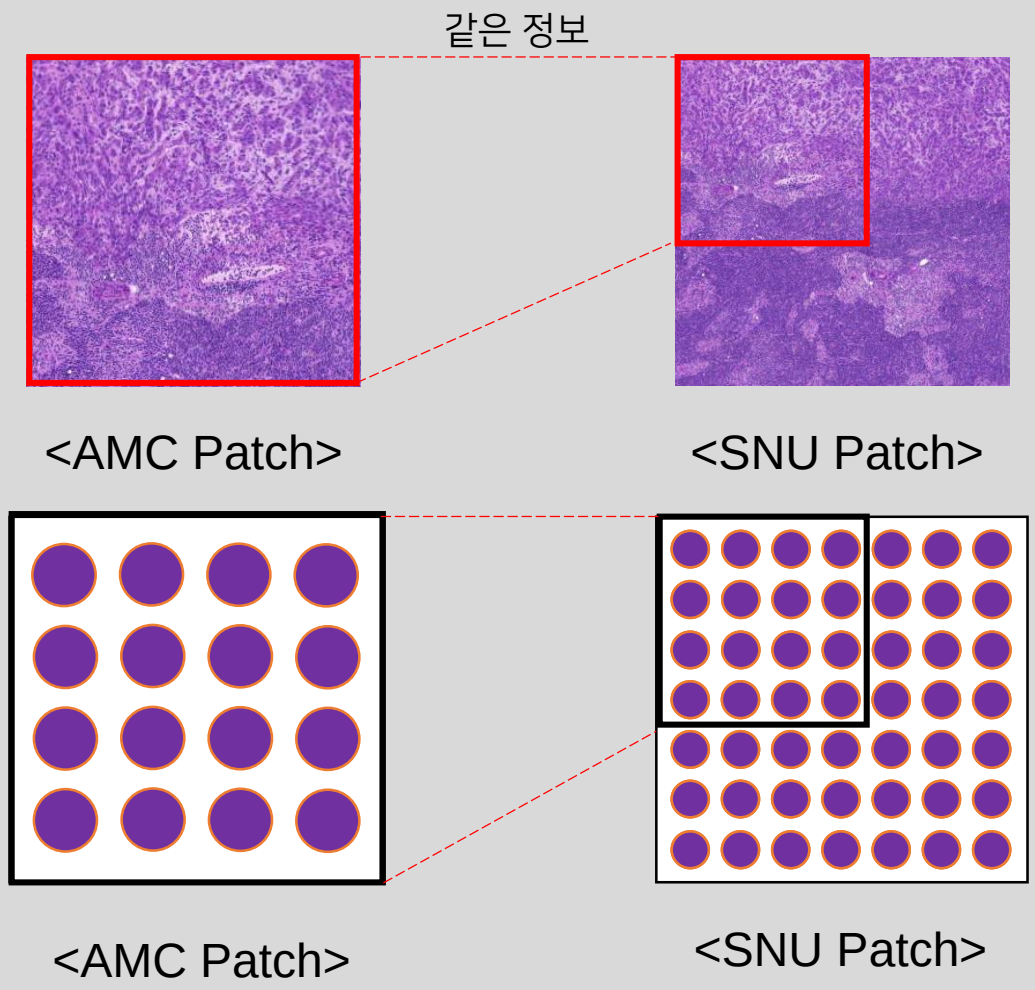
Task* – *Cancer classification + Major axis



이미지 출처: HeLP Challenge 2019 설명회 발표자료

1. Background – 대회 설명

➢ Data Description



<Patch 하나가 담은 정보의 양>

(level 4) 256 x 256 크기 Patch

서울아산병원(AMC)

$256 \times 16 \times 0.221 = 905.216(\mu m)$

☞ **0.905** x **0.905** (mm^2)

분당서울대병원(SNU)

$256 \times 16 \times 0.389 = 1593.344(\mu m)$

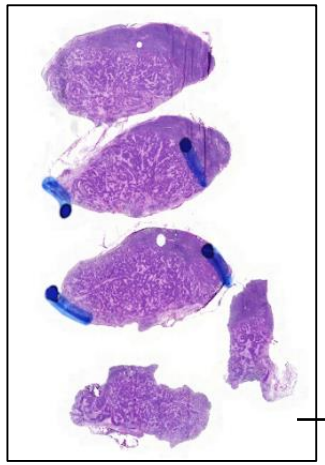
☞ **1.593** x **1.593** (mm^2)

1. Background – 대회 설명

➢ Summary

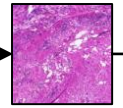
	HeLP Challenge 2018	HeLP Challenge 2019
Task	**Frozen pathology** 를 암 또는 정상으로 분류하기	
	-	암인 부분의 **major axis** 계산
Data	**서울아산병원**	서울아산병원, **분당서울대병원**

STAGE 1

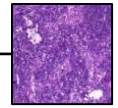


Whole slide image

Region of interest extraction + Rescale

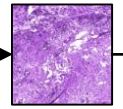


patch

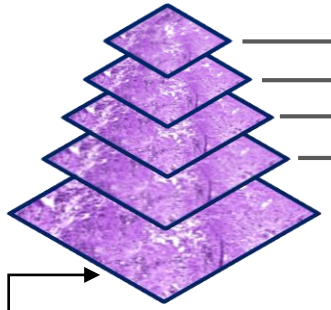


target

Vahadane's Stain Normalization



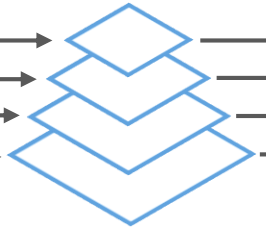
patch norm



add

add

add

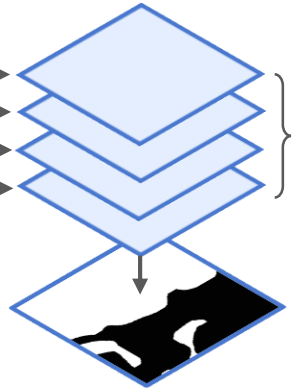


up

up

up

up



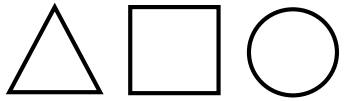
concat

STAGE 2

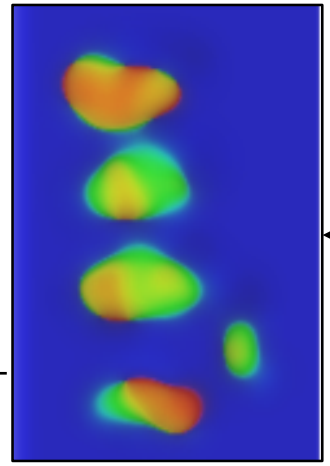
Normal ? Tumor ?

Major axis

ML regression / classifier



Feature vector extraction

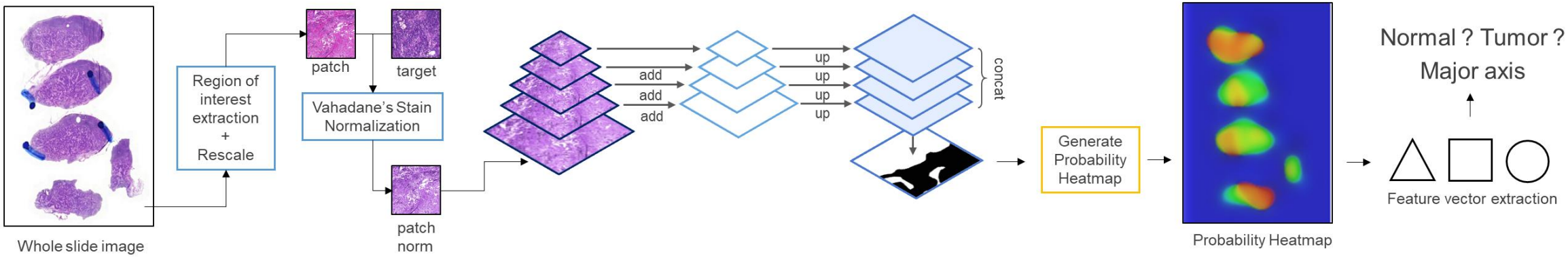


Probability Heatmap

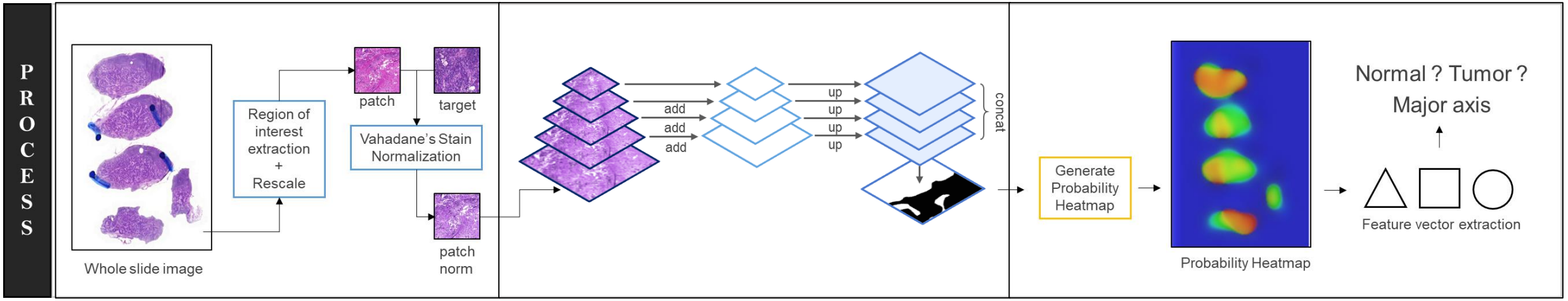
Generate Probability Heatmap

2. Algorithm

PROCESS



2. Algorithm



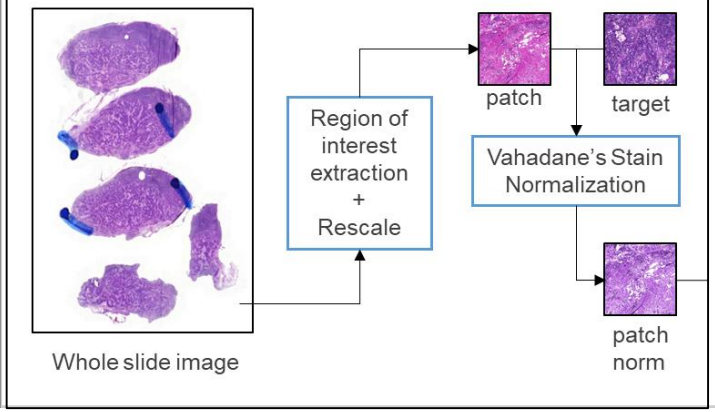
Preprocessing
for minimizing domain gap
& training

Pixel-level
tumor regression

Extraction feature
from probability heatmap

2. Algorithm

PROCESS



Region of interest extraction

학습할 패치 영역 추출

Rescale

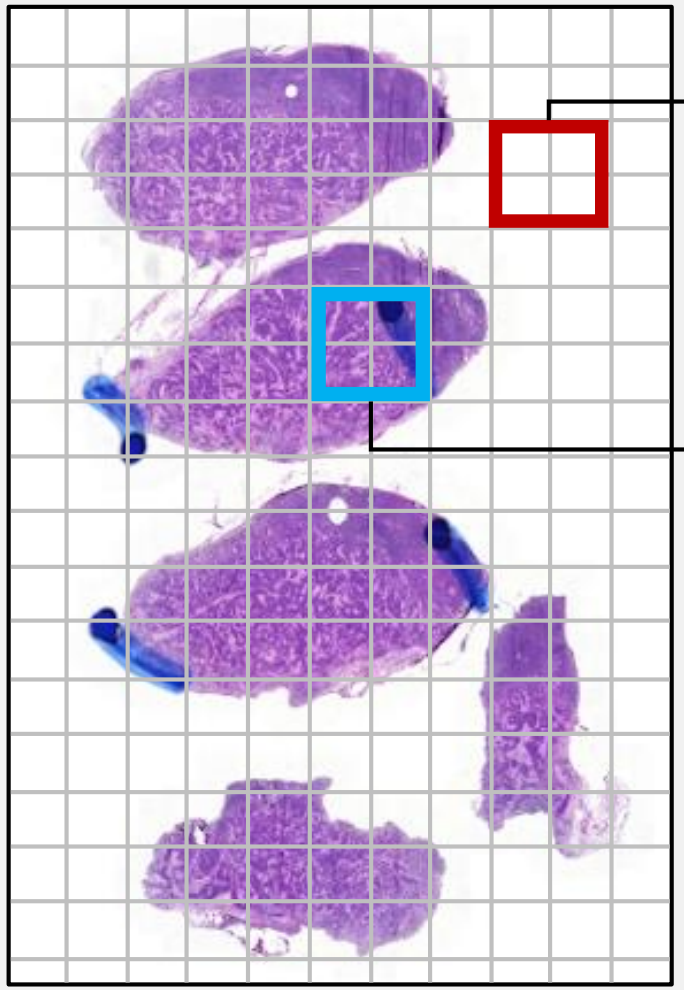
병원기관별
서로 다른
배율 맞추기

Vahadane's Stain Normalization

병원기관별 , 슬라이드별
서로 다른
염색의 정도 맞추기

2. Algorithm

Region of interest extraction | Rescale | Vahadane's Stain Normalization



Whole slide image **(WSI)**

Otsu's thresholding method

Normal : Tumor
1 : 1

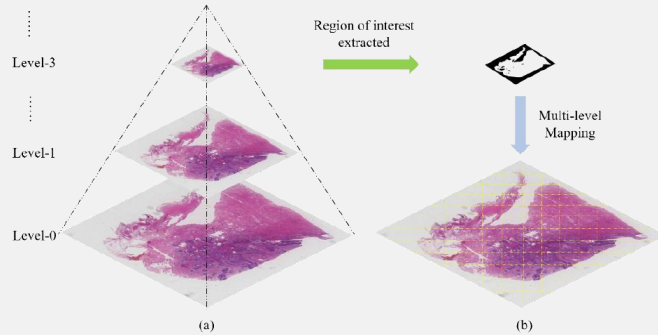
Save img for training

2. Algorithm

Region of interest extraction | **Rescale** | Vahadane's Stain Normalization

➢ Data Scale Matching

- 아산병원AMC 과 서울대병원SNU의 데이터 셋은 다른 resolution을 가지고 있음
- AMC 와 SNU 데이터를 같은 level 4에서 추출하는 것은 정보의 손실 발생할 수 있음
- 데이터의 resolution을 맞춰주기 위해 AMC는 level 4에서 SNU는 level 3을 사용함
- 각각 다른 level에서 추출한 patch의 크기를 256 × 256 으로 맞춰 줌



서울아산병원 Asan Medical Center

SNUH 분당서울대학교병원

Solution ☝

level 4 (256 x 256)

level 4 (145 x 145) → Upscale → (256 x 256)

Solution ✌

level 4 (256 x 256)

level 3 (290 x 290) → Downscale → (256 x 256)

Interpolation 과정 중
손실 발생 가능성을
없애기 위해
Solution 2를 사용

2. Algorithm

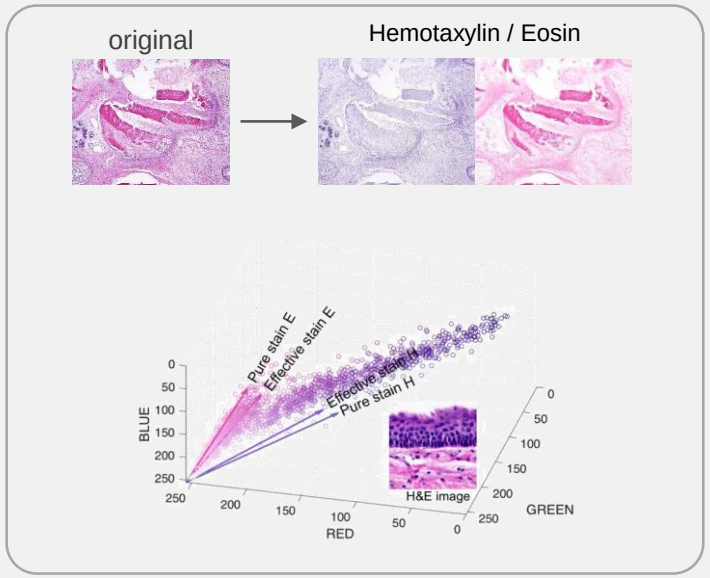
Region of interest extraction | Rescale | **Vahadane's Stain Normalization**

* Structure-Preserving Color Normalization and Sparse Stain Separation for Historical Images, Vahadane(2016)

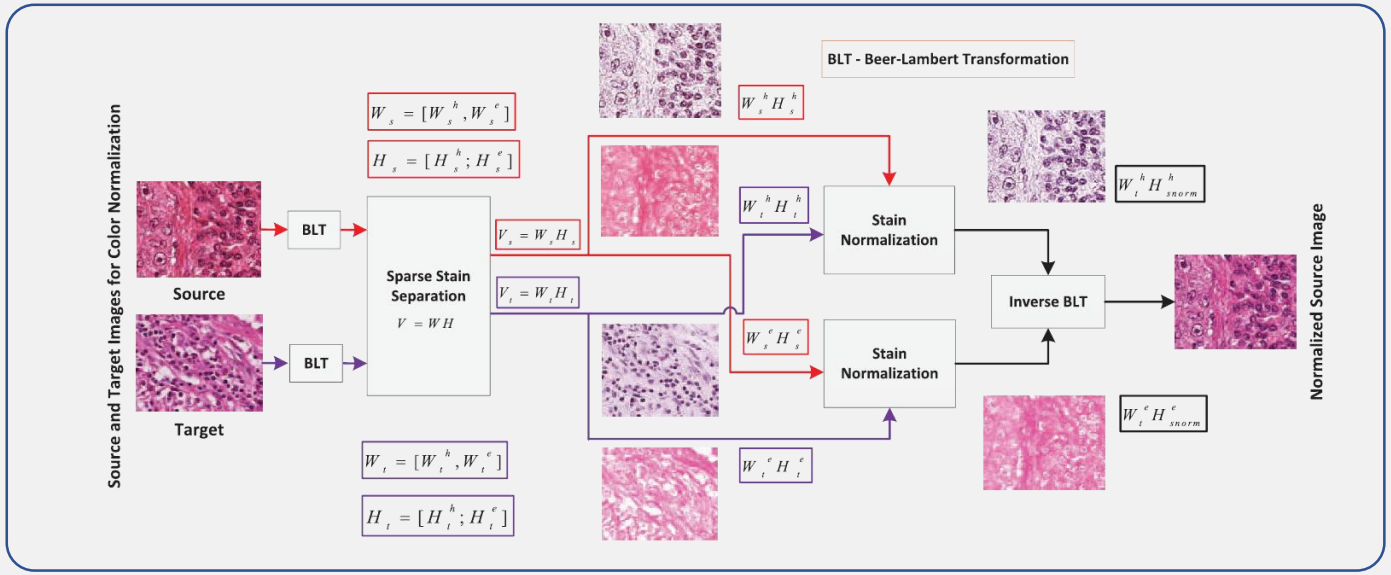
➢ Vahadne's Stain Normalization (Vahadane*, 2016)

- 학습 데이터셋(AMC)과 테스트 데이터셋(SNU)은 frozen pathology의 stain이 다를 가능성이 큼
- 전처리 단계에서 Vahandane (2016)이 제안한 stain normalization을 적용함
  - Source의 구조는 그대로 유지하면서 stain normalization을 해주는 structure-preserving color normalization(SPCN) 방법
  - Source : normalization을 적용하기 위한 입력 이미지
  - Target : source 이미지를 normalize 하여 target의 색상과 맞춰주기 위한 이미지

Stain Separation



Vahadane's Stain Normalization

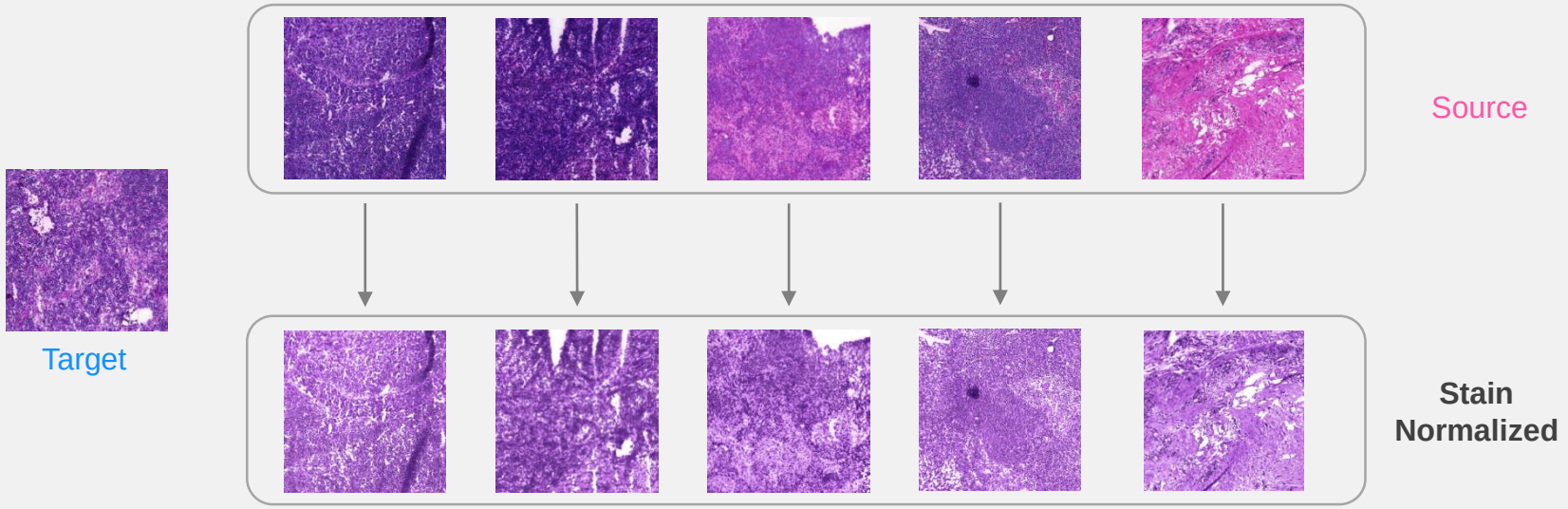


2. Algorithm

Region of interest extraction | Rescale | **Vahadane's Stain Normalization**

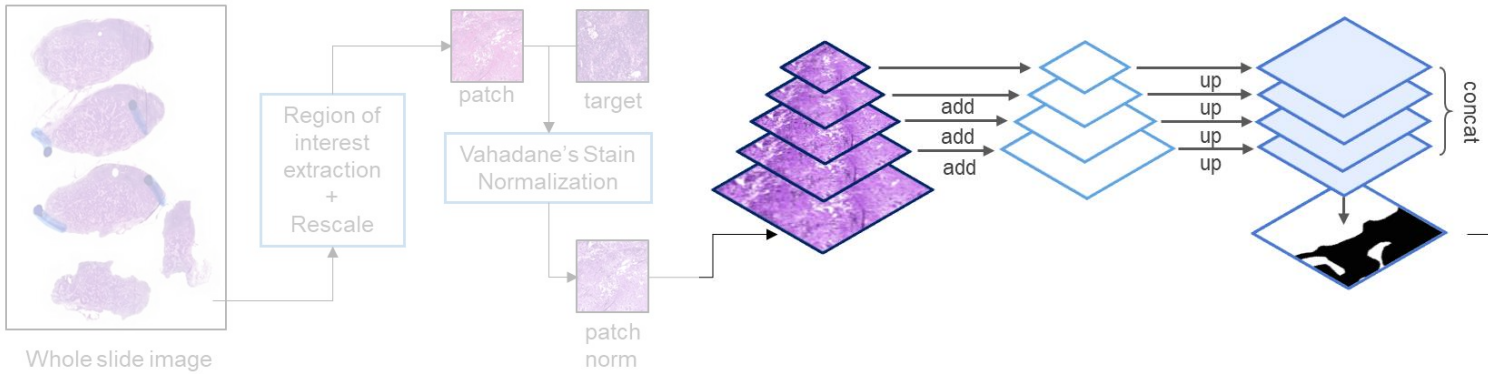
➢ Vahadne's Stain Normalization (Vahadane*, 2016)

- 학습 데이터셋(AMC)과 테스트 데이터셋(SNU)은 frozen pathology의 stain이 다를 가능성이 큼
- 전처리 단계에서 Vahandane (2016)이 제안한 stain normalization을 적용함
  - Source의 구조는 그대로 유지하면서 stain normalization을 해주는 structure-preserving color normalization(SPCN) 방법
  - Source : normalization을 적용하기 위한 입력 이미지
  - Target : source 이미지를 normalize 하여 target의 색상과 맞춰주기 위한 이미지



2. Algorithm

PROCESS



Model Architecture

모델 구조 설명

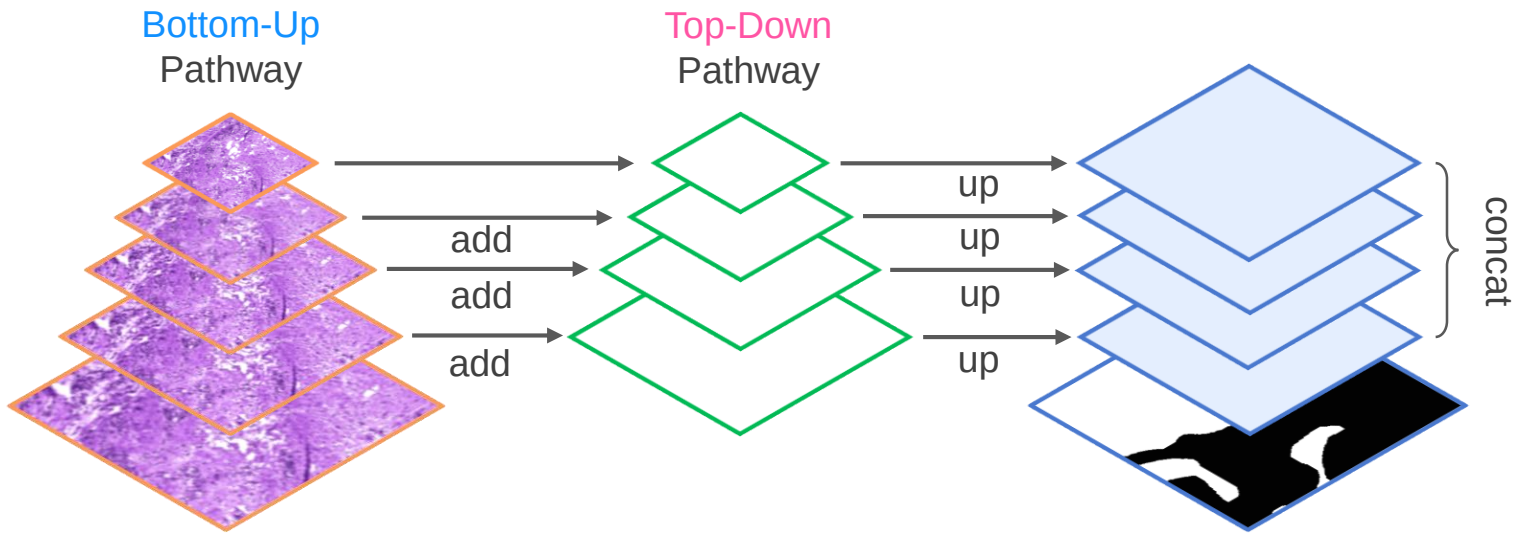
Training detail

학습 세부사항

2. Algorithm

➢ **FPN** (Feature Pyramid Network)

- FPN은 Bottom-Up/Top-Down 2개의 path가 있음 :
  - Bottom-Up은 feature map의 크기size를 조절하며 정보를 추출하는 역할
  - Top-Down은 feature map에 대해 up sampling하여 더 높은 해상도의 이미지를 만드는 역할
- Top-Down path에서, Bottom-Up path에서 추출한 feature map을 더해add 성능을 높임

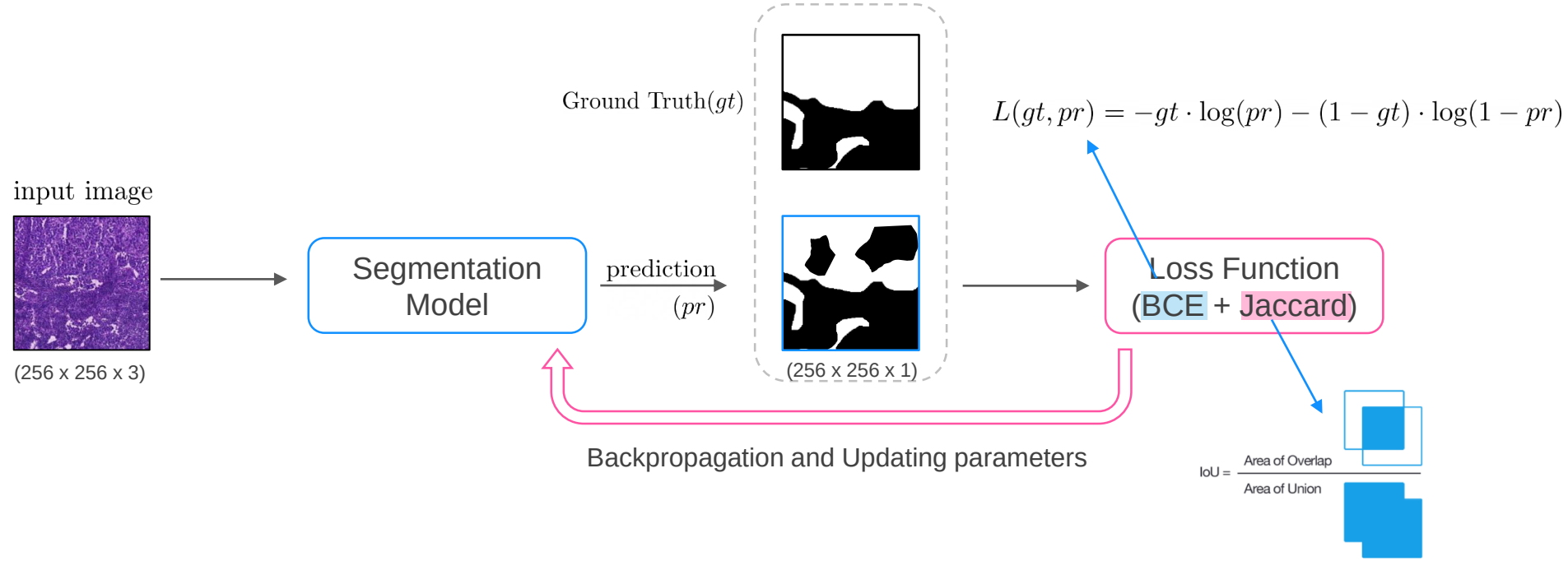


2. Algorithm

* Using Intersection over Union Loss to Improve Binary Image Segmentation, Beers(2018)

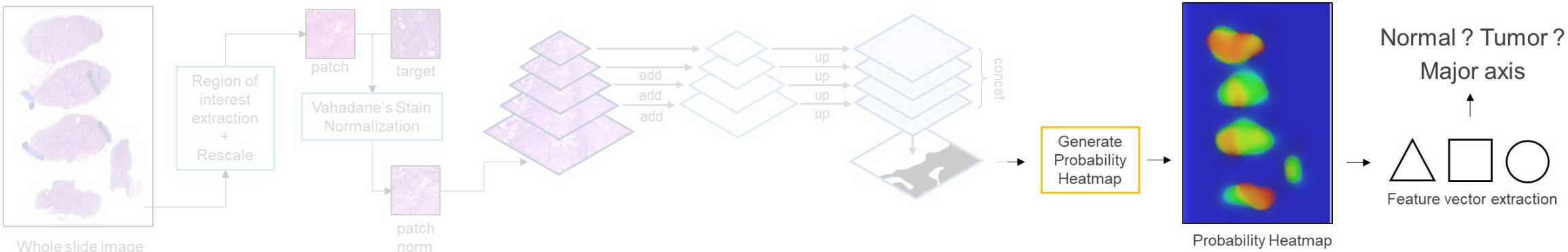
➢ Training detail

- Input : patch (256 x 256 x 3) → Output : probability heatmap of patch (256 x 256 x 1)
- Data augmentation : horizontal flip, vertical flip, rotation randomly
- Loss function
  - Binary Cross Entropy^{BCE} + jaccard loss$^{IOU,\ Intersection\ Over\ Union}$ 를 사용함
  - Beers* (2018) 논문을 참고하여 BCE loss에 jaccard loss를 더한 손실 함수를 사용
- Learning rate scheduler : initial 1e-3, reduced lr when validation loss doesn't improve
- 5-fold cross validation training



2. Algorithm

PROCESS



Generate probability heatmap

전체 슬라이드의
확률 히트맵 생성

Feature extraction

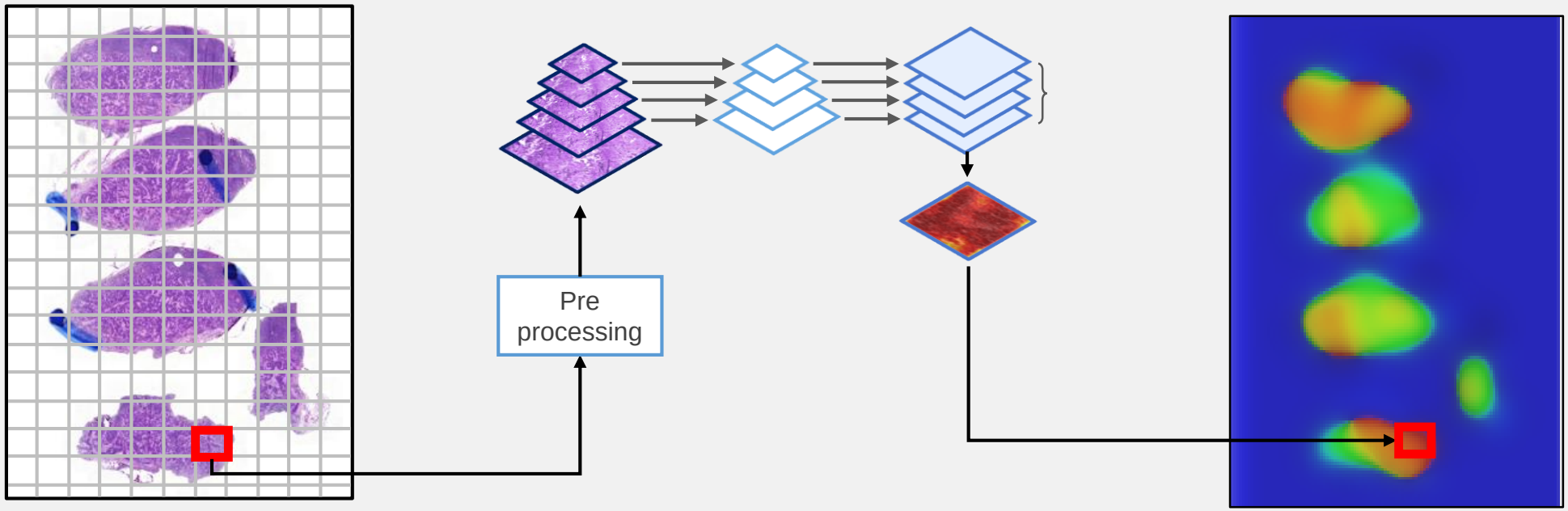
확률 히트맵으로부터
유의미한 특징들 추출

Prediction

특징을 바탕으로
Tumor classification 및
major axis 예측

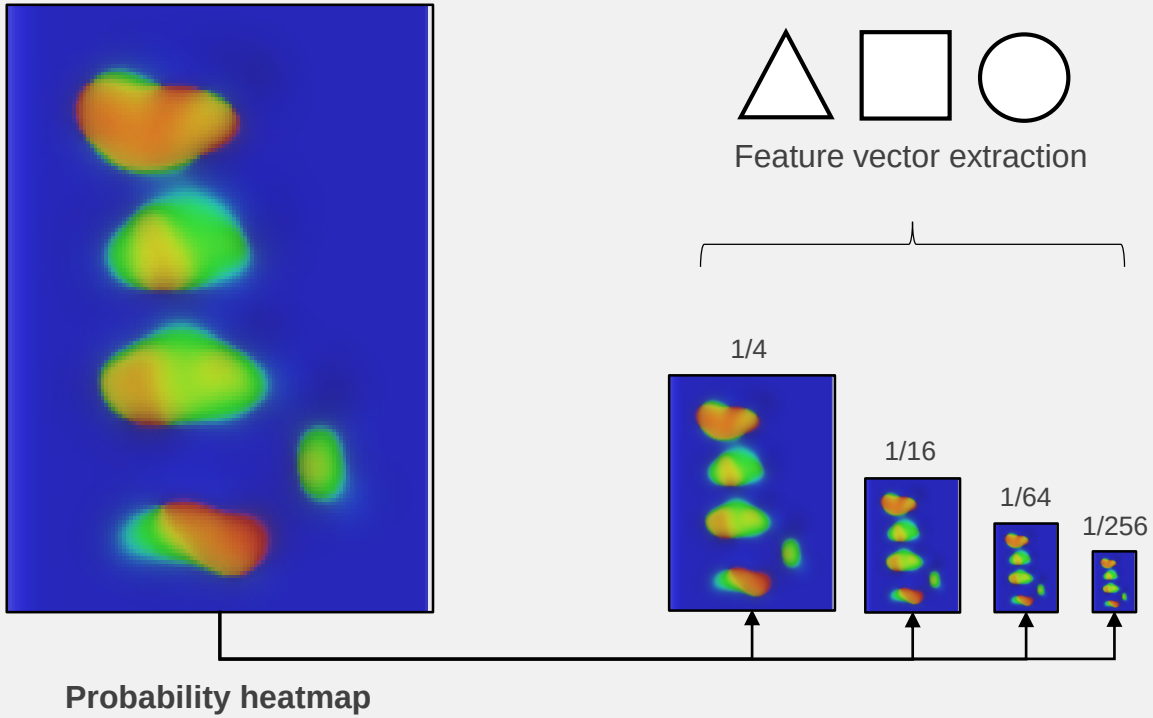
2. Algorithm

Generate probability heatmap | Feature extraction | Prediction



2. Algorithm

Generate probability heatmap | **Feature extraction** | Prediction



Feature list

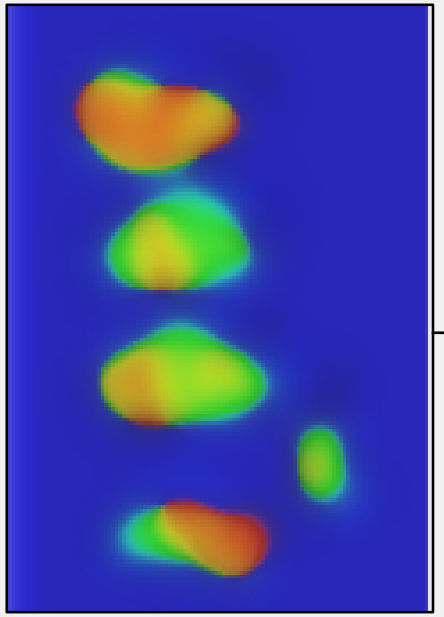
- Major axis with threshold 0.5
- Major axis with threshold 0.9
- Tumor patch ratio with threshold 0.5
- Tumor patch ratio with threshold 0.9
- Max probability of patches
- Mean probability of patches
- Std probability of patches

→ 7 features from different scale probability heatmap
→ 7 * 4 = 28 features

(1/4, 1/16, 1/64, 1/256 x original probability heatmap size)

2. Algorithm

Generate probability heatmap | Feature extraction | **Prediction**



Probability heatmap

Feature vector extraction

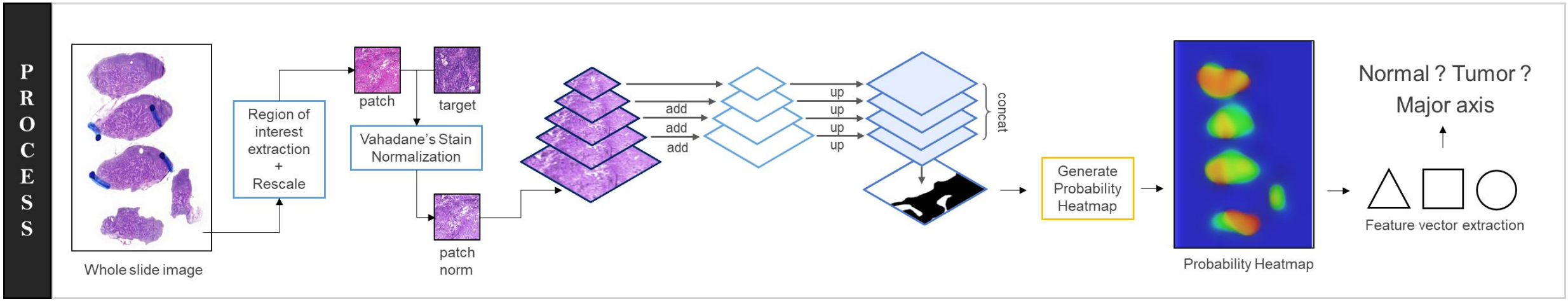
ML regression / classifier

Normal ? Tumor ?

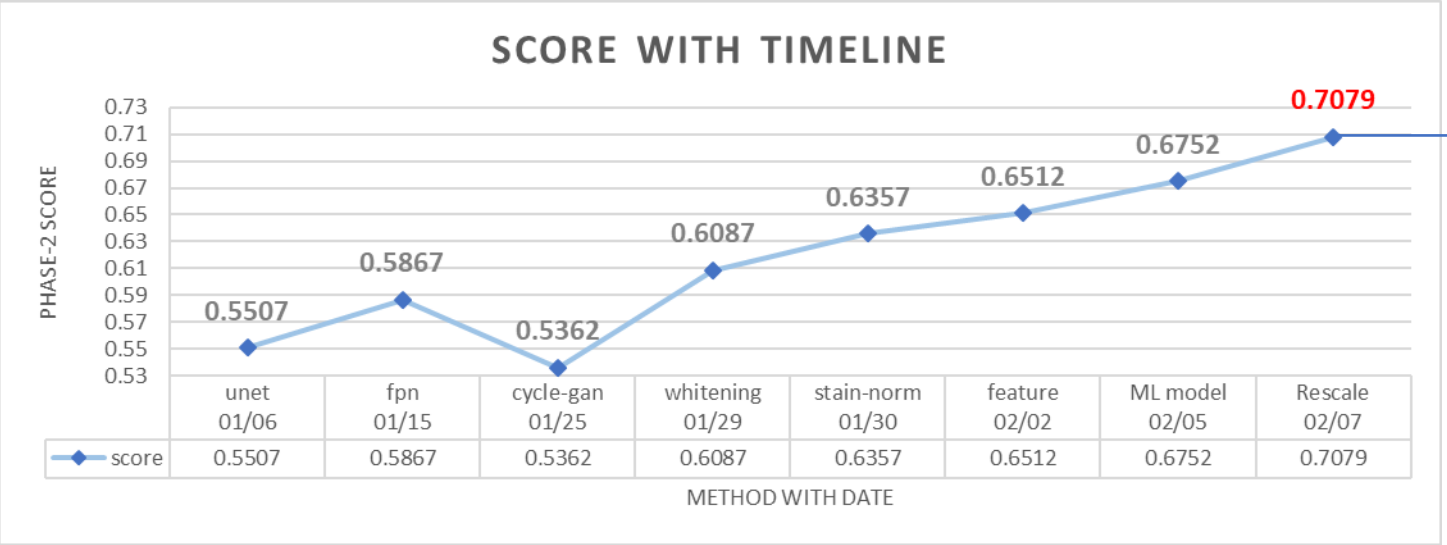
Major axis

- Linear, Ridge, Lasso, KernelRidge, Elastic
- RandomForest, GradientBoosting, ExtraTrees
- XGBoost, LightGBM etc..

2. Algorithm


PROCESS
Whole slide image
Region of interest extraction + Rescale
patch
target
Vahadane's Stain Normalization
patch norm
add
add
add
up
up
up
up
concat
Generate Probability Heatmap
Probability Heatmap
Normal ? Tumor ?
Major axis
Feature vector extraction

2. Algorithm – Experiment result



AUC of slides	Acc of major axis	Total score
0.8908	0.5249	**0.7079**

	ours								will
simple unet?	√								
fpn?		√	√	√	√	√	√	√	√
cycle-gan?			√						
whitening?				√					
stain normalization?					√	√	√	√	√
feature extraction?						√	√	√	√
ML model?							√		√
rescale?								√	√
Phase1 score	0.5869	0.5982	0.6136	0.6095	0.6434	0.6714	**0.7123**	x	
Phase2 score	0.5507	0.5867	0.5362	0.6087	0.6357	0.6512	0.6752	**0.7079**	**more than + 0.04**

2. Algorithm

➢ Average inference time per WSI

- Preprocessing : 3.32 minutes
- Segmentation model prediction with feature extraction : 0.27 minutes
- Total : 3.59 minutes

➢ Things We Tried That Didn’t Work

- Cycle-GAN
- Instance normalization, min max scale, whitening per patch img
- 2nd stage with machine learning model without stain normalization

➢ Future work

- 2nd stage with machine learning model with rescale and stain normalization
- Extract another features
- Another 2nd stage model such as unsupervised algorithms

3. Project Management

➢ Collaboration with Notion

- 자료 공유, 스케쥴 정리, 제출 기록, 진행 상황 등 전체적인 부분에 **Notion**을 활용
- 코드 블록, 표 등 다양한 컴포넌트를 지원하며, 여러 사용자가 동시에 수정이 가능하다는 장점

SCHEDULE

Date	ToDo	caution	비고
12.23 - 12.29	콘테스트 내용 정리, 관련 주제 research, 개발환경세팅 완료, 카브 개발환경 이해	notion에 정리하면서, 지난대회 SotA 와 머신러닝 뒷단붙이는거 반드시 찾기	
12.30 - 1.5	관련 모델 구현과 카브에 직접 올리기	raw 하게 한 번 짜고, 코드 리팩토링	못올림
1.6 - 1.12	성능 내기!!	모델 올리면 세부적인 사항을 꼼꼼하게 기록해서 결과들을 비교할 것	0.587
1.13 - 1.19	`SNU vs AMC` 데이터 최소화 (가장중요) , 다양한 network로 학습	SST Network (대영 , 태우) Color Normalization + (종현)	
1.20 - 1.26 (설날)	`SST` 바탕으로 새로 classifier 학습 `2-stage` 모델 개발 - feature extraction		
1.27 - 2.2	`Phase 2` Validation Set Start `SNU vs AMC` 데이터 차이 최소화!!!	대회 마무리 준비	
2.3 - 2.7		여러 모델 앙상블 준비	

New

3. Project Management

DOCKER 관리

Untitled + Add a View Sort Search New

TASK ID	CREATED AT	내용	비고
16f7cc38-34c0-4061-9a43-b4a410256d61	2020-02-07 23:26	stage 2 meta + major_axis	
6663fd62-3268-41d0-a840-278654db7b03	2020-02-07 23:26	stage 2 only meta	
10524fe9-abbd-4b32-bbb6-d13f93126042	2020-02-07 23:06 ing	[0] *SNU level 3 test 2 feature*	예상 종료시간 : + ~30 mins = **오후 11시 35분**
a20a6934-edf OPEN a5fb-91d611109fc5	2020-02-07 19:19 *Finished*	[0] *SNU level 3 test 2 stain norm img* 저장	if i < 125 : continue 126 ~ 181 까지 img 저장 담당 : Slide 470 ~ 525 **예상 종료 시간 : 오후 10시 40분**
f0e187c8-0112-41e5-84fd-d98268cac1cd	2020-02-07 18:30 *Finished* *0.592*	[3] *Predict Metastasis & Major-axis directly from Feature*	train 에서 best_auc_col, best_acc_col, best_acc_threshold 를 그대로 가져와 test2 feature 에서 best_auc_col → metastasis 예측, best_acc_col + best_acc_threshold → major_axis 예측
f3039f56-8f11-4e53-ad71-e76126fbc8ec	2020-02-07 17:41 *Finished* ***train : 42 mins*** ***test2 : 20 mins***	[2] *Get AMC Feature + More*	thresholds = [0.2, 0.5] / heatmap size = [4, 16, 64] train and test2 only ← feature 설명과 feature path는 **Task ID 클릭** **예상 종료 시간 : 오후 7시**
801a094d-ed14-4853-b5d5-bfc067d26122	2020-02-07 16:51 *Finished*	[2] *Get AMC Feature*	총 5개의 feature 만 뽑음 : 64_major_axis_t0.2, 64_tumor_ratio_t0.2, 64_max, 64_mean, 64_std train : 32mins
215eadbb-4ce4-4b40-886a-6c4c727c33dd	2020-02-07 16:21 ing	[0] *SNU level 3 test 2 stain norm img* 저장	→ 다 되면 test2 에 대해서 기존 모델(fpn_cjh)으로 예측하고 새롭게 feature 만들기 **예상 종료시간 : 오후 11시 (이후에도 모델은 계속 돌아갈 예정 → 도커 종료 요청)**
f7b69153-3cb3-4e3c-9bfe-968b64e6641c	2020-02-07 12:29 *Cancled*	[1] *Rescale_amc - fpn network training 다시*	**MODEL_NAME = 'fpn_cjh_rescale4/'** **처음부터 재학습.... 2fold 까지 학습 f1 score = 0.84 ~ 0.85 / loss = 0.45** no divide 255 **종현님 model setting 기반으로 change!** sm.FPN('resnet34',input_shape = (256,256,3), classes = 1,encoder_weights= None,activation = 'sigmoid')

COUNT 14

3. Project Management

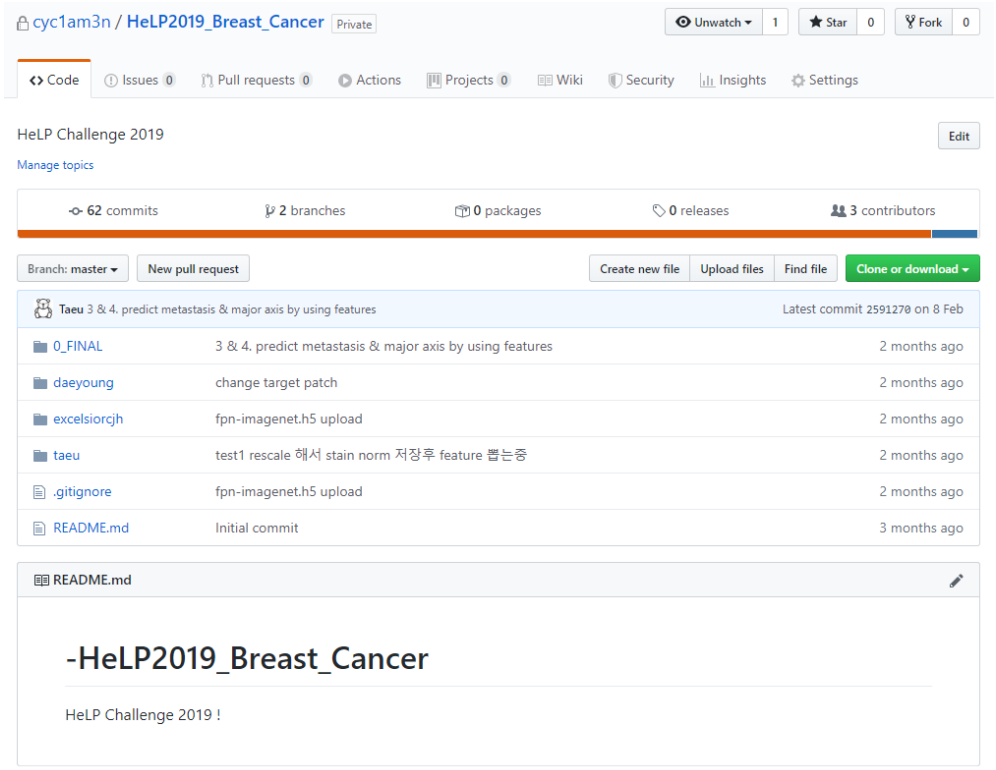
추가 - AMC data set 은 기존 dir에서 가져오기

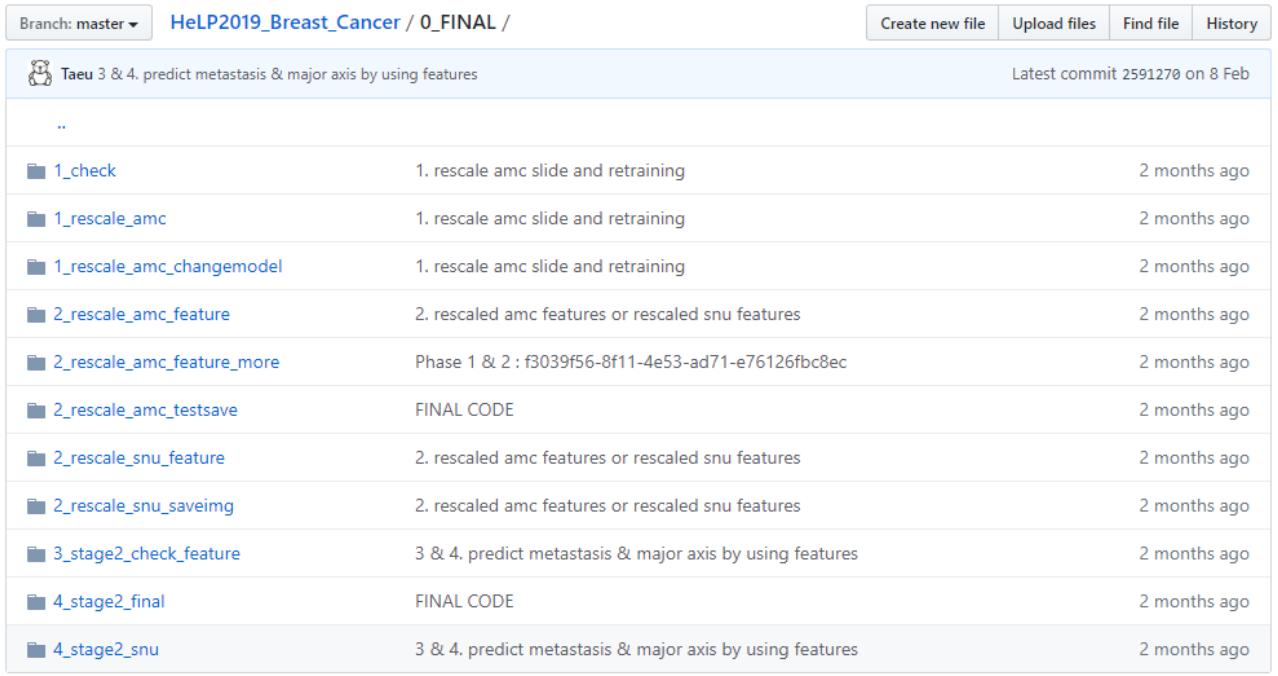
```
if im.size[0] == 256 :
  current_save_dir = '/data/volume/patches/stain/'+phase+'/'+ slide_path[:-4] + '/'

if IS_PREPROCESSED:
  try :
    full_stain_patches_path = current_save_dir + str(idx) + '.png'
    cnt += 1
    img = Image.open(full_stain_patches_path)
    X = np.array(img, dtype =np.uint8)

  except:
    X = np.zeros((256,256,3))
else :
  if img.size[0] == 145: img.resize((256,256)) ## 이부분 추가
      X = np.array(img, dtype = np.uint8)
      try :
          X = staintools.LuminosityStandardizer.standardize(X)
          X = normalizer.transform(X)
          x_img = Image.fromarray(X)
          x_img.save(current_save_dir + str(idx) + '.png')
      except:
          X = np.zeros((256, 256,3))
  else :
      try :
          current_save_dir = '/data/volume/patches/stain/'+phase+'/'+ slide_path[:-4
          full_stain_patches_path = current_save_dir + str(idx) + '.png'
          cnt += 1
          img = Image.open(full_stain_patches_path)
          X = np.array(img, dtype =np.uint8)
      except :
          X = np.zeros((256,256, 3))
```

3. Project Management

➢ Code Managing with Github

- 코드 관리에는 **Github**을 활용
- branch를 각자 만들기 보다는 master에 개인 폴더를 만들어 사용





3. Project Management

➢ Code Managing 2 : modulization

- baseline model부터 용도 별로 코드를 모듈화해 refactoring 진행 *(김보섭님 자료* 참조)*

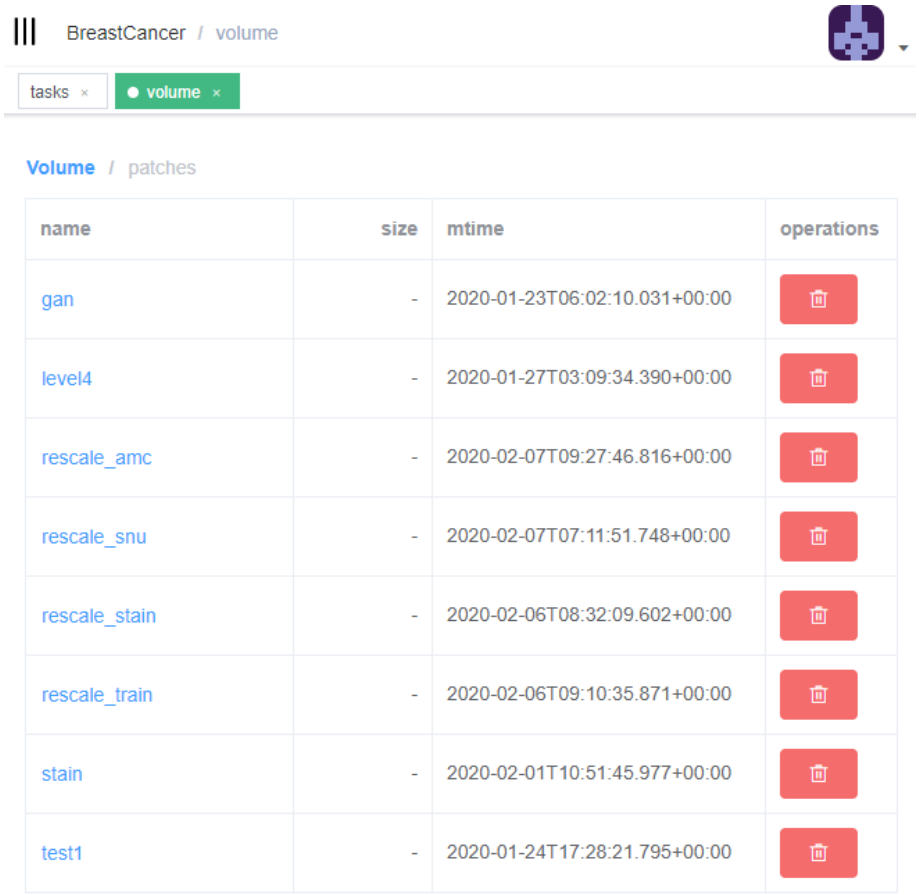
```
preprocess/
    prep.py             # slide -> patch
model/
    weight/
        unet_pretrained.h5
    data.py             # DataGenerator
    net.py              # network architecture
    ops.py              # net에 필요한 operation
    utils.py            # augmentation
train.py                # model 학습
inference.py            # model 평가
utils.py                # 학습 및 평가에 필요한 opeation ex)get_major_axis
config.json             # 아직 안 만들었음
```

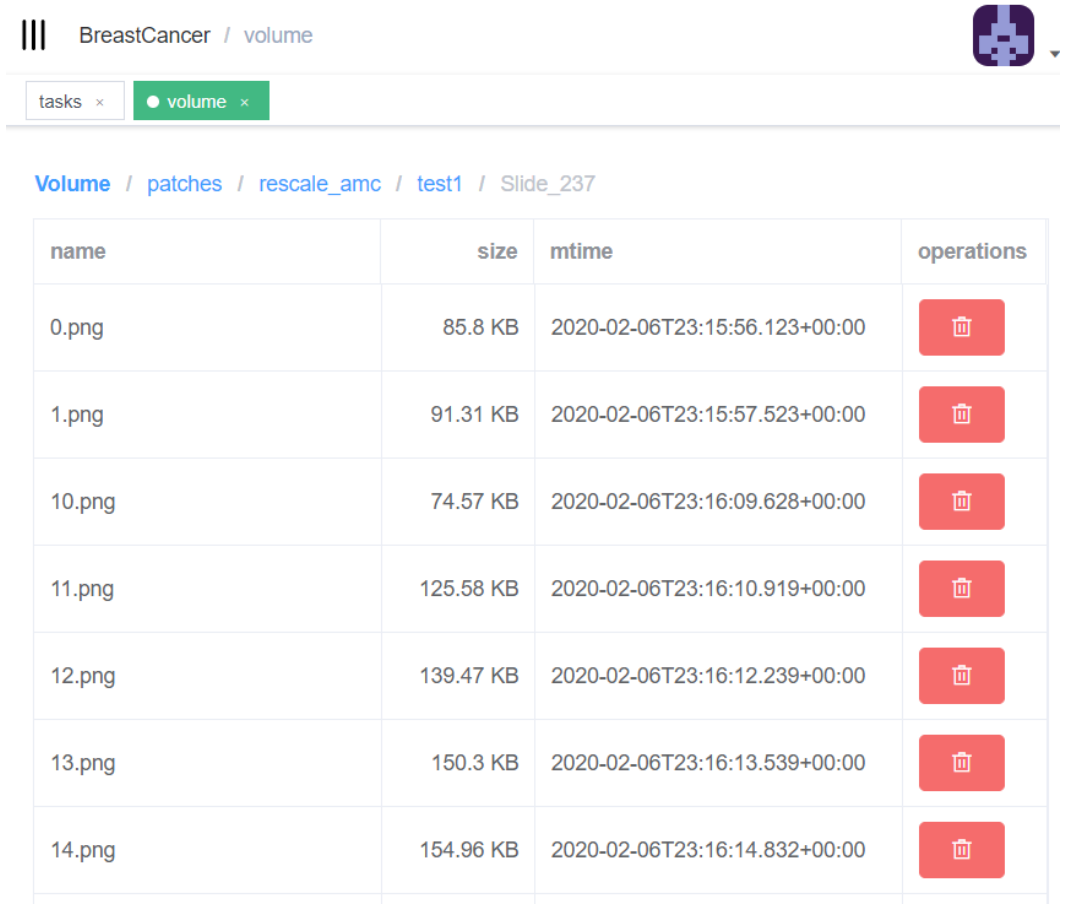
* https://tykimos.github.io/warehouse/2019-7-4-ISS_2nd_Deep_Learning_Conference_All_Together_aisolab_file.pdf

3. Project Management

➢ 기타 노하우

- 각 단계별 전처리 및 결과 파일들 저장
  - 카카오 브레인 클라우드 서버의 Volume 스토리지 활용
- 작은 세팅부터 복잡한 세팅으로 실험 진행





4. Review

➢ 좋았던 점

- 도커에 익숙하지 않은 참가자 팀들을 위해 태스크 관계자인 김성철 연구원님께서 베이스라인 코드를 제공해주신 점
- 카카오 브레인 클라우드 서버 개선
  - 스토리지를 웹 상으로 볼 수 있음
  - 각 로그도 나름 실시간으로 확인할 수 있음
  - 2개의 GPU 사용 가능 등

➢ 개선 사항

- 카카오 브레인 클라우드 관련
  - GPU를 사용하지 않더라도 CPU 따로 돌릴 수 있게 했으면
  - docker image 대신 코드만 업로드를 해 서버 상에서 docker image 생성했으면
- 태스크 관련
  - 암인 부분의 major axis 구하는 metric 관련
    - 오차 범위 5%의 accuracy라 실제 major axis가 50 픽셀일 경우, ±2.5 픽셀 미만의 오차로 맞춰야 함
    - 얼마나 작은 크기의 major axis를 구분 하려는 지 제시하면 더 좋을 것 같음

Thank you

Open Source Code Link
https://github.com/cyc1am3n/HeLP2019_Breast_Cancer_1st_solution

Domain adaptation for histopathological images

Vahadane, Abhishek, et al. "Structure-preserving **color normalization** and sparse stain separation for histological images." IEEE transactions on medical imaging 35.8 (2016): 1962-1971.

Cho, Hyungjoo, et al. "Neural **stain-style transfer** learning using gan for histopathological images." arXiv preprint arXiv:1710.08543 (2017).

Shaban, M. Tarek, et al. "**Staingan**: Stain style transfer for digital histological images." 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE, 2019.

Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using **cycle-consistent adversarial networks**." Proceedings of the IEEE international conference on computer vision. 2017.

Breast cancer metastases detection

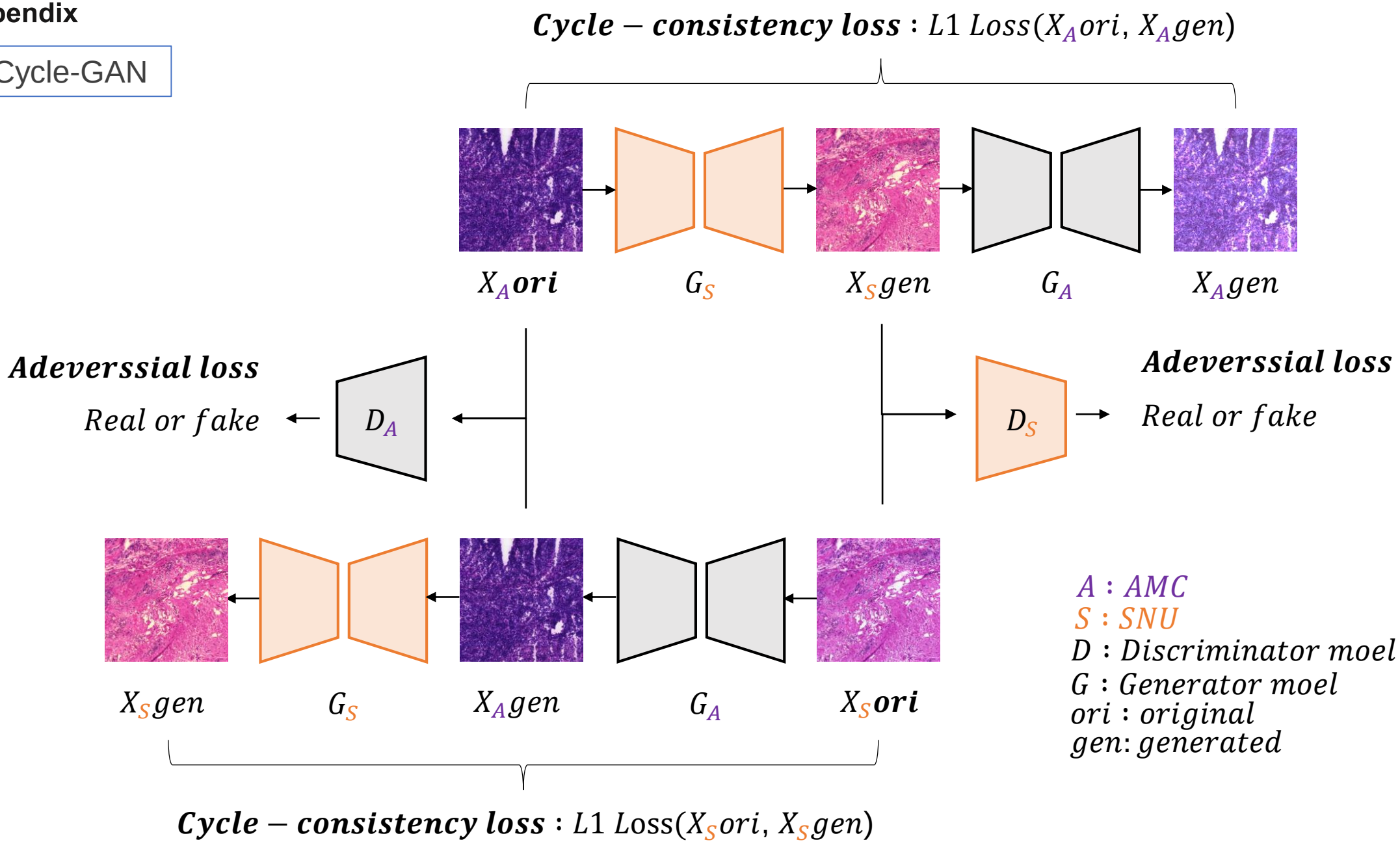
Bandi, Peter, et al. "From detection of individual metastases to classification of lymph node status at the patient level: the **camelyon17 challenge**." IEEE transactions on medical imaging 38.2 (2018): 550-560.

“Breast Cancer stage classification in histopathology images”, Byungjae Lee and Kyunghyun Paeng **Lunit Inc**., Seoul, Korea (2017)

Automatic classification on patient-level breast cancer metastases Sanghun Lee, Joonyoung Cho, Sun Woo Kim **Deep Bio Inc**. (2019)

5. Appendix

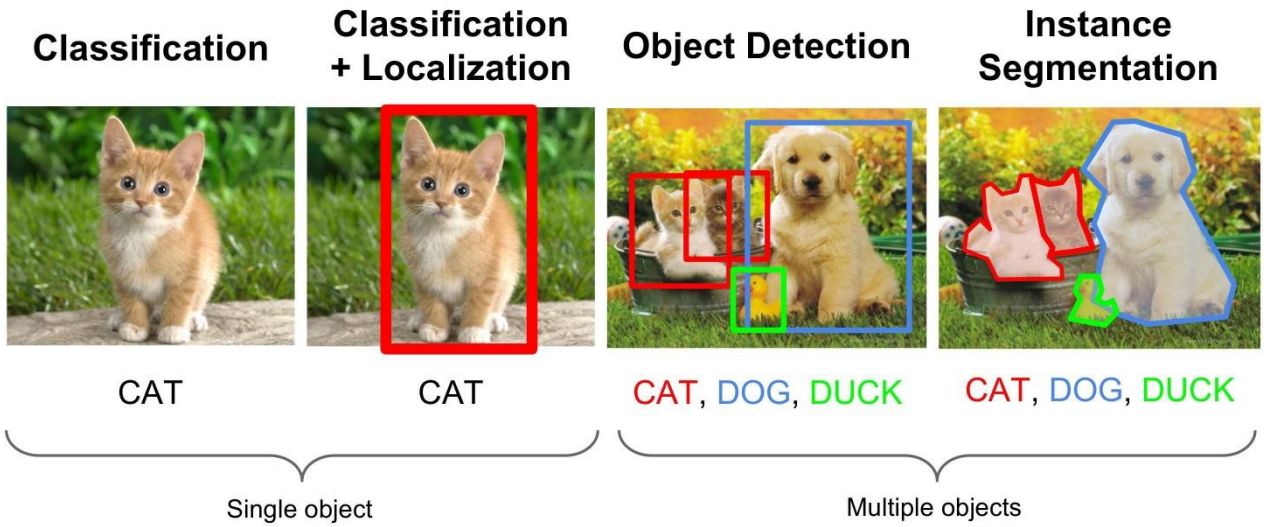
Stain-Cycle-GAN



A : AMC
S : SNU
D : Discriminator moel
G : Generator moel
ori : original
gen: generated

➢ Segmentation Model

- 컴퓨터 비전computer vision, CV 에서 많이 다뤄지는 Task의 유형 3가지 중 하나
  - Classification : 입력input 에 대해서 하나의 레이블을 예측하는 작업
  - Localization/Detection : 물체object의 레이블을 예측 + 물체의 위치 정보 제공
  - Segmentation : 모든 픽셀의 레이블을 예측
- 이미지에 있는 모든 픽셀에 대한 예측을 하므로, **dense prediction**이라고도 함



➢ U-Net

- Convolutional Networks for Biomedical Image Segmentation model
- U-Net은 contraction/expansion 2개의 path가 있음 :
  - Contraction path는 이미지 feature에 대한 정보를 추출하는 역할
  - Expansion path는 transpose convolution을 통한 localization 역할
- Expansion path에서, contraction path에서 추출한 feature map를 합쳐concatenate localization 성능을 높임