HuBMAP
Human BioMolecular Atlas Program

Hacking the Human Body

HuBMAP + HPA

CODEnter

INDEX

Target

[TEAM MAIN TASK : SEGMENTATION] ------------------------ [KAGGLE COMPETITION 선정 기준]

- Segmentation 으로 Car Damage Detection 한 경험
- 이를 발판삼아, 다른 TASK 를 진행해 보고자 하였다.
- 팀원들의 선호도 : CV task

- Sementic Segmentation
- 현재 진행중인가?
- CV 계열인가?

HuBMAP | Human Bio Molecular Atlas Program

HPA | Human Protein Atlas

Competition Goal —— Python | Deep-Learning Model

- 여러 장기의 생검 슬라이드에서 각 기능 조직단위 [FTU] 의 위치를 식별하는 것
- 분자 구조에서 신체 추측 | 인체를 구성하는 세포 사이의 기능 및 관계를 파악 | 5개의 장기에 걸쳐 기능 조직 단위(FTU)를 식별하고 분류
- 현 의료 데이터들의 다양한 이미지 사이즈에서 보다 효율적으로 분류할 수 있는 일반화 된 모델 개발이 개최 목표

Project Overview

Overview

Competition Rule

6 / 22	9 / 15	9 / 22
대회 개최	참가신청 / 팀 합병 마감	최종 제출 마감

Submit File Shape

- rle Format - 'submission.csv' 이름으로 제출
- KAGGLE 업로드 시, 주최측의 비공개 데이터셋으로 TEST가 진행
- 해당 결과값을 기반으로 점수가 산출

DEV - ENV

python SOFTWARE FOUNDATION, colab, amazon web services, Visual Studio Code, Weights & Biases, MMSegmentation, papermill, ubuntu, ANACONDA, fast.ai, PyTorch, Linux

Expected Effect

- 세포 구성을 더 잘 이해하면 사람들이 더 건강하고 장수하는 데 도움이 될 수 있다.
- 세계적으로 세포와 조직간의 관계에 대한 이해를 가속화할 수 있다.
- 인간의 건강에 영향을 미치는 세포의 기능에 대해 더 많은 통찰력을 얻을 수 있다.

Team Member & Role

Member & Role

윤소미	AWS dev-env manage, Model Training , Writing Down Hyper-Parameter from Training Code, Inference, kaggle upload
장주찬	Dataset Refine, Dataset Creating, Dataset Pre-Processing, Augmentation, Inference, kaggle upload, Model info
전규원	Model Training, EfficientNet Thesis & Hyper Parameter Information Search, PPT, Inference, kaggle upload, Reference Modify
한정현	AWS dev-env manage, Model Training , Trying Max Batch-Size and Fixed, Inference, kaggle upload, Competition info
허 권	Model Training, Dataset Refine, Dataset Creating, Inference, kaggle upload, Reference Modify, Mosaic Dataset create
박정현	PM: Issue Tracking & Resolving, Role Assignment, Guidance for Project Planning & Model Training, Project Pipeline Set up

Project Timeline & Workflow

Timeline

7월 4주 ~ 8월 2주 [총 3주] : MMDET , MMSEG Study / 8월 3주 ~ 8월 5주 [총 3주]

SEARCH [8월 3주차]

Competition

- DACON
- Kaggle
- 국내 공모전
- 해외 공모전

Select

- Kaggle
- HuBMAP & HPA

Competition Analysis

- Competition Rule
- Reference Search

DATA [8월 3 - 4주차]

Analysis

- Dataset Shape
- Data Type
- Train Image Unbalanced
- Submit Shape

Pre-Processing

- Original Image Converting
  - 256 * 256
  - 512 * 512
  - 768 * 768
- Custum size Converting
  - 256 * 256
  - 512 * 512
  - 768 * 768
- Mosaic

MODEL [8월 4 - 5주차]

Modeling

[MMSEG]

- Study about MMSEG
- Various Model Select
- Trying Modelings
- Outputs Analysis
- Recording Try List
- Submit Kaggle

[FastAI - EfficientNet]

- Rererence in Kaggle
- Trying Modeling
- Modify Inference Code
- Outputs Analysis
- Submit Kaggle

DEVELOP [8월 5주차 ~]

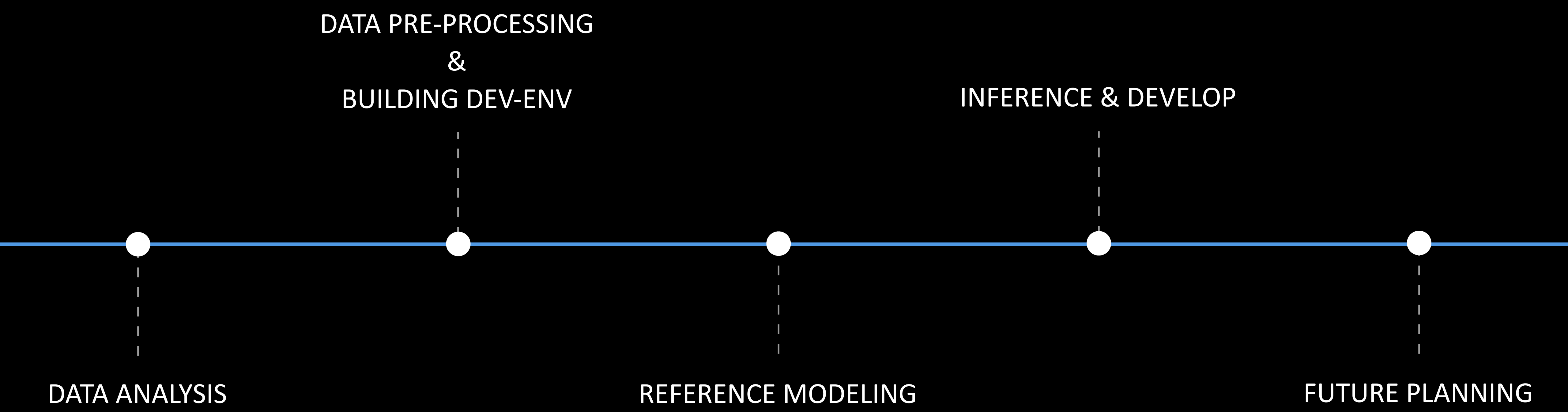
Development

- Augmentation Search
- Hyper Parameter
- Error and low score cause analysis
- Using AWS
- AWS DEV-ENV Build

Future Planning

- Ongoing Competition

Workflow


DATA ANALYSIS
DATA PRE-PROCESSING
&
BUILDING DEV-ENV
REFERENCE MODELING
INFERENCE & DEVELOP
FUTURE PLANNING

Project Detail & Result

Project Detail

✓ Data EDA [Exploratory Data Analysis]

	id	organ	data_source	img_height	img_width	pixel_size	tissue_thickness	rle	age	sex
0	10044	prostate	HPA	3000	3000	0.4	4	1459676 77 1462675 82 1465674 87 1468673 92 14...	37.0	Male
1	10274	prostate	HPA	3000	3000	0.4	4	715707 2 718705 8 721703 11 724701 18 727692 3...	76.0	Male
2	10392	spleen	HPA	3000	3000	0.4	4	1228631 20 1231629 24 1234624 40 1237623 47 12...	82.0	Male
3	10488	lung	HPA	3000	3000	0.4	4	3446519 15 3449517 17 3452514 20 3455510 24 34...	78.0	Male
4	10610	spleen	HPA	3000	3000	0.4	4	478925 68 481909 87 484893 105 487863 154 4908...	21.0	Female

< train.csv >

- **id** : 고유번호
- **organ** : 장기 명칭
- **data_sorce** : HPA
- **img_height / img_width** : 이미지 사이즈
- **pixel_size** : 1픽셀 사이즈
- **tissue_thickness** : 조직 두께
- **rle** : 마스크 정보
- **age / sex** : 나이 / 성별

	id	organ	data_source	img_height	img_width	pixel_size	tissue_thickness
0	10078	spleen	Hubmap	2023	2023	0.4945	4

rle value ?

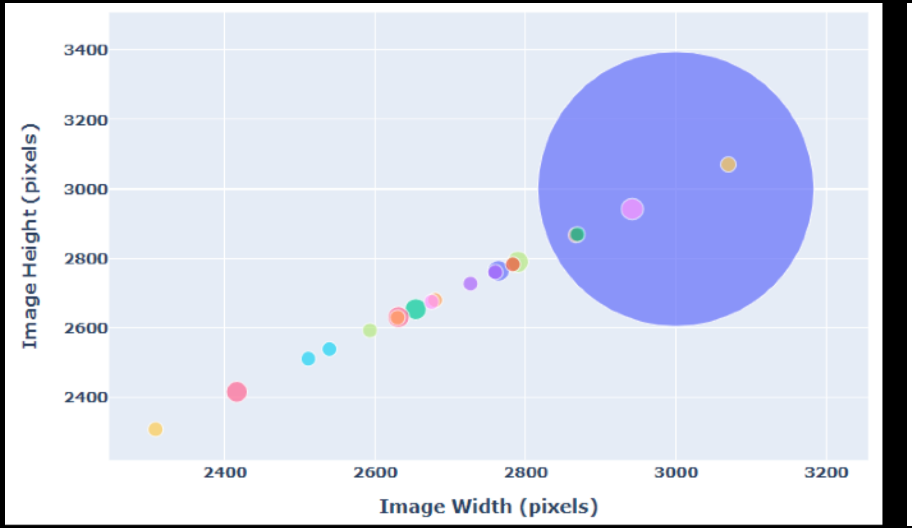
< test.csv >

- train 후, **rle** 값 예측

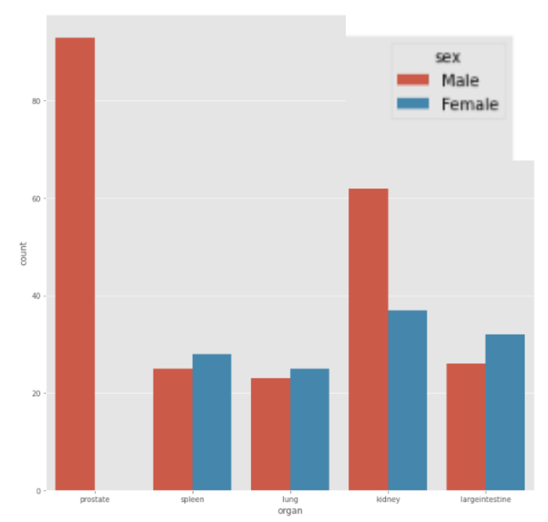
Project Detail

✓ Data EDA [Exploratory Data Analysis]

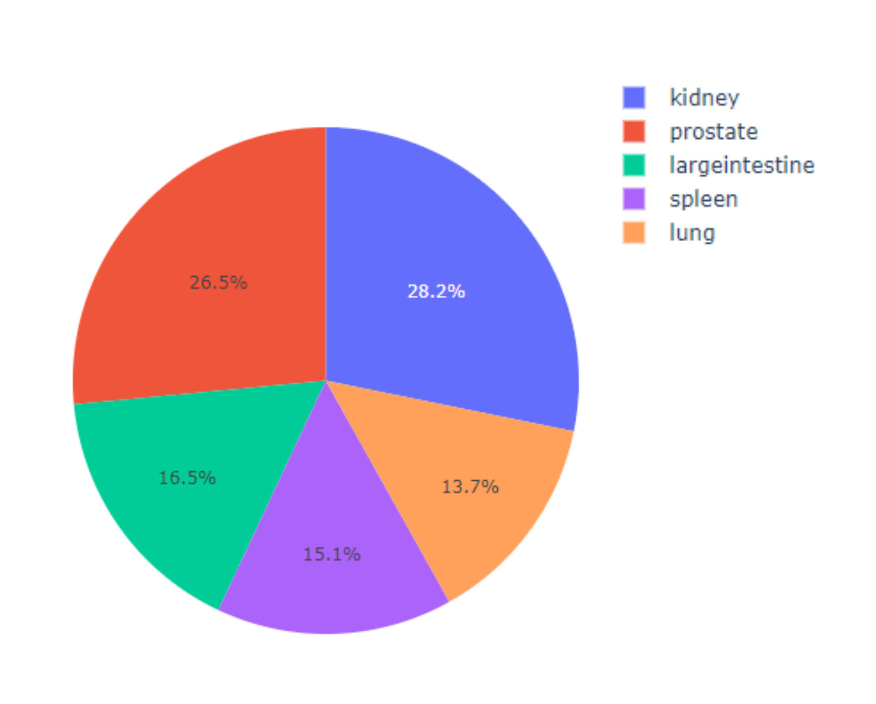
Dataset Image Info.



Sex & Organ Dataset



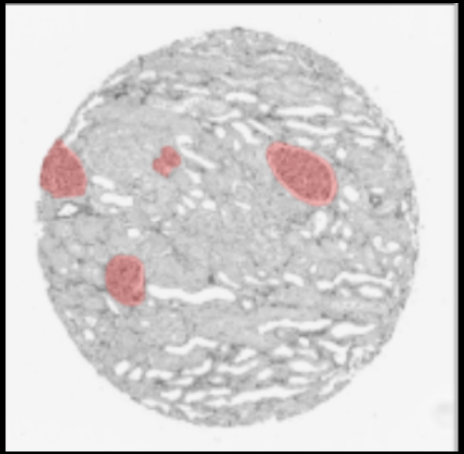
Organ Dataset Statistics



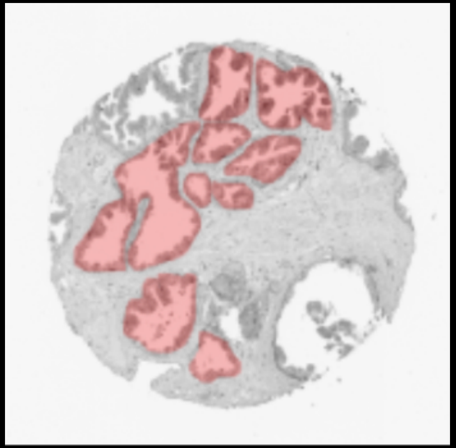
- kidney [신장] : 28.2%
- prostate [전립선] : 26.5%
- largeintestine [대장] : 16.5%
- spleen [비장] : 15.1%
- lung [폐] : 13.7%

Project Detail

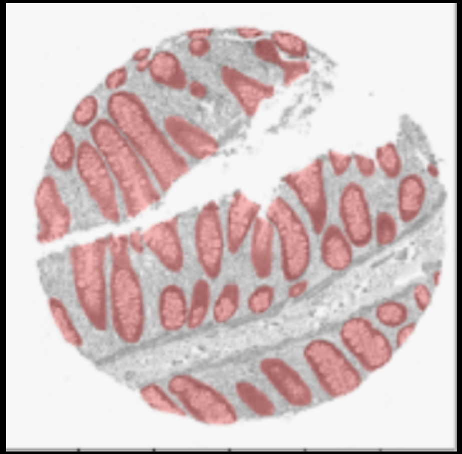
✓ Mask Shape [Each 5-Class]



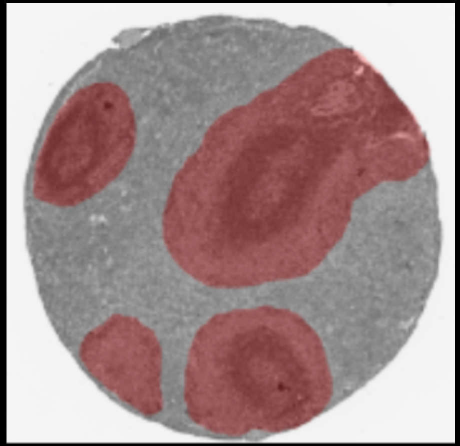
kidney [신장]
Count : 99



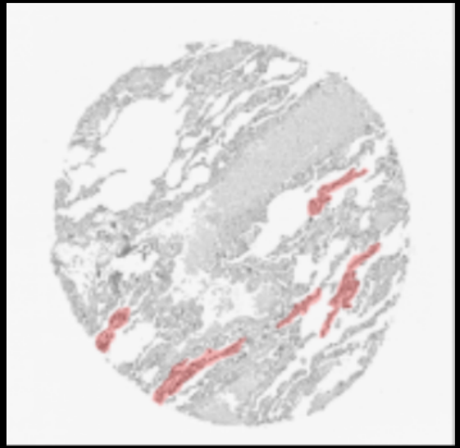
prostate [전립선]
Count : 93



largeintestine [대장]
Count : 58



spleen [비장]
Count : 53



lung [폐]
Count : 48

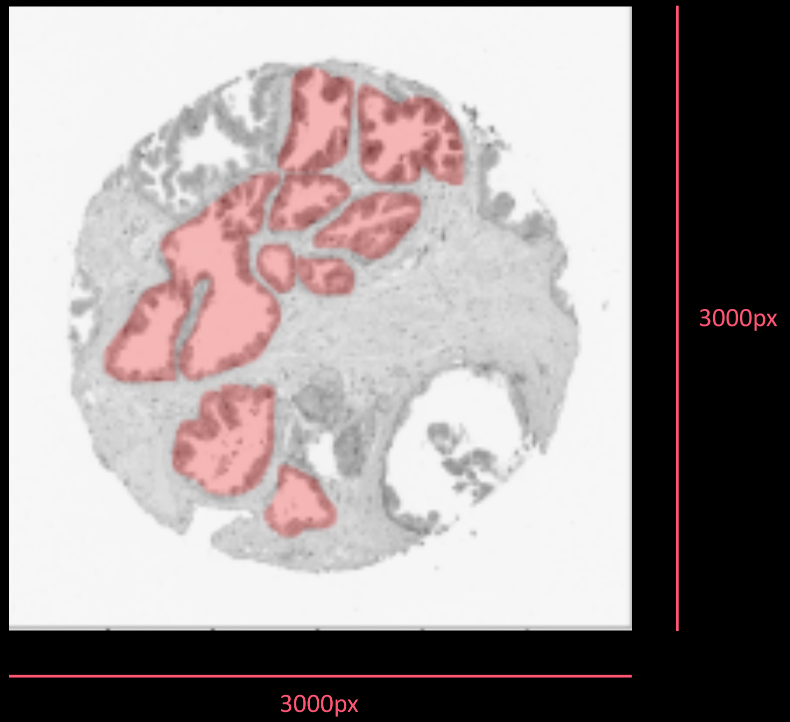
Lung Dataset

- 가장 적은 Dataset 수
- 5가지 class 중, 가장 작은 영역의 Mask 형태를 보유

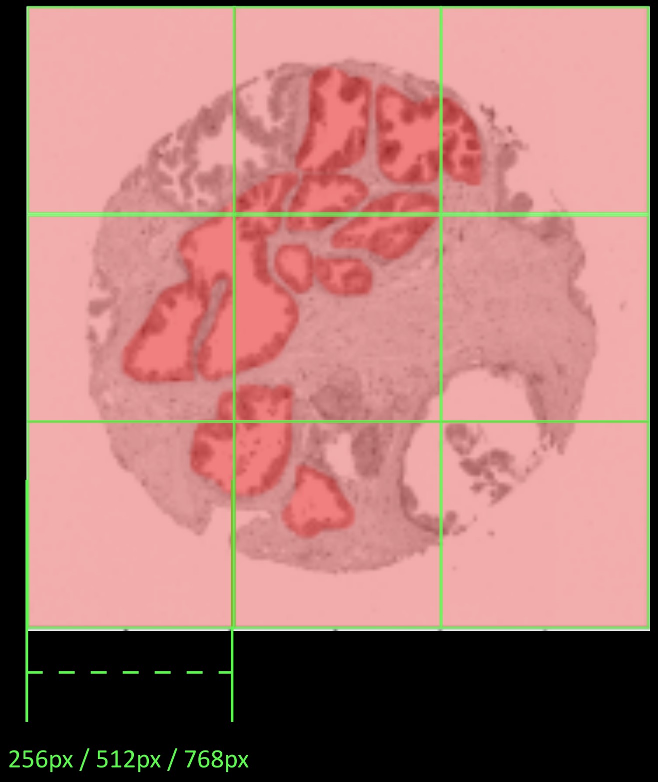
Project Detail

✓ Converting Original Dataset

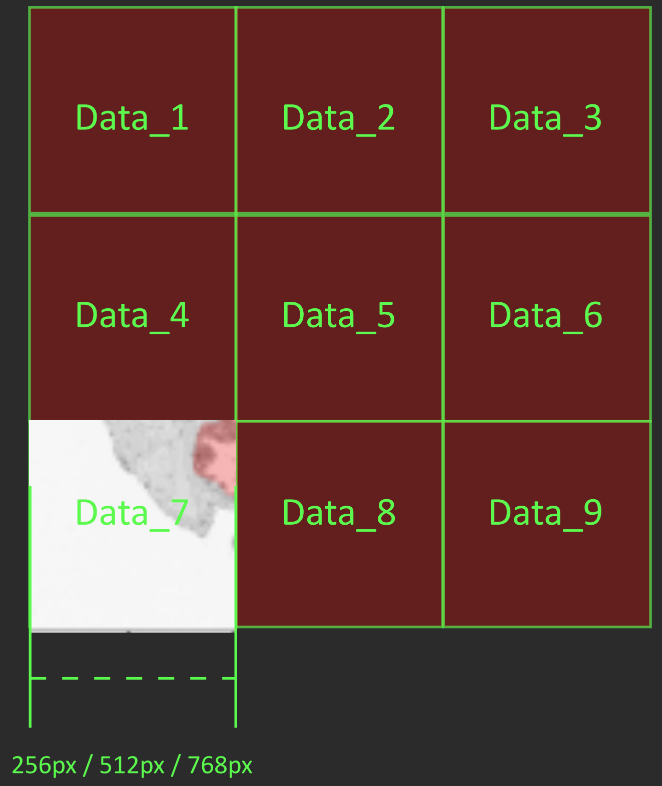
Sample Original Image
[3000*3000 image]



Converting Original Dataset



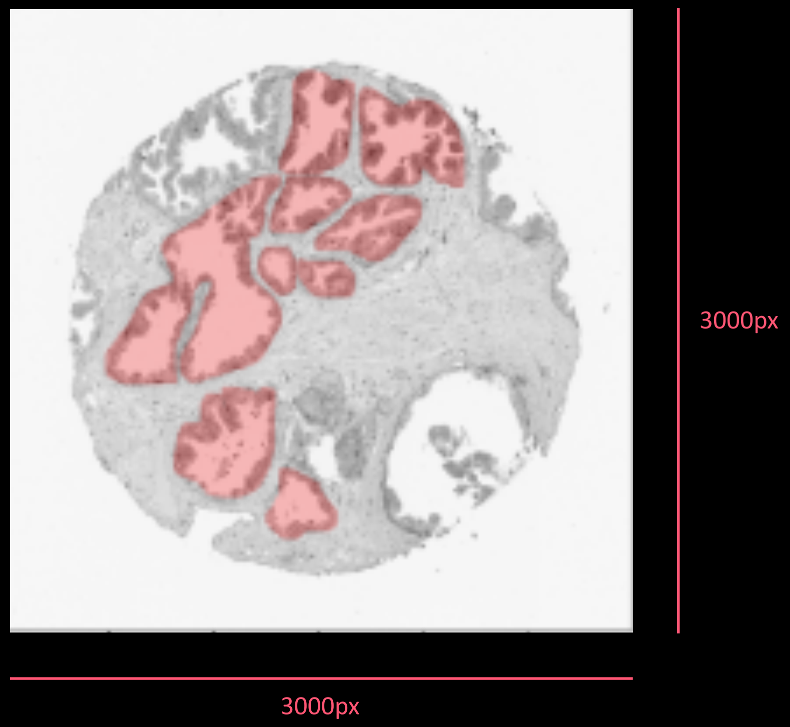
Converting Various Size
[256 / 512 / 768 px]



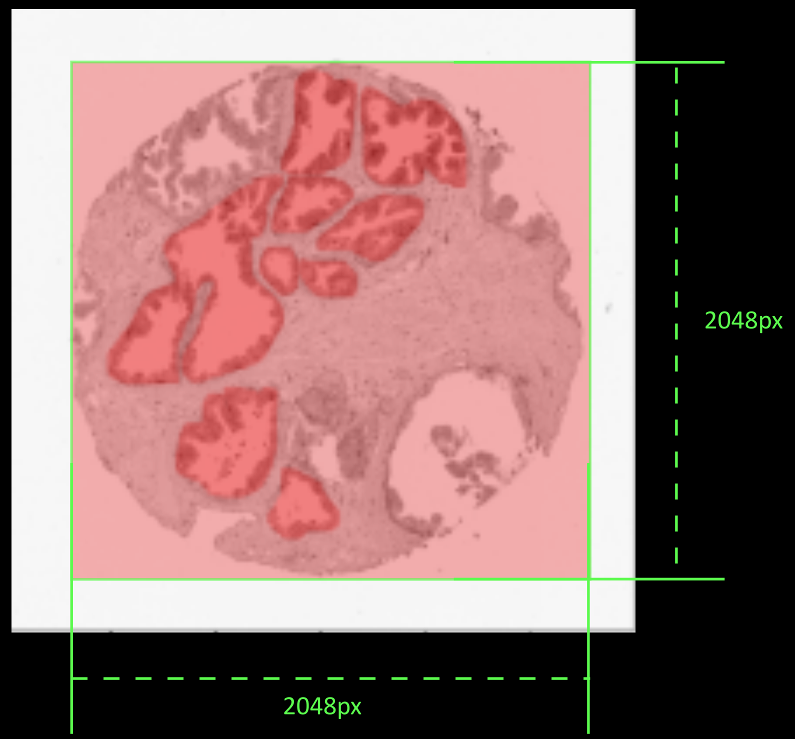
Project Detail

✓ **Remove** Unnecessary Background Area

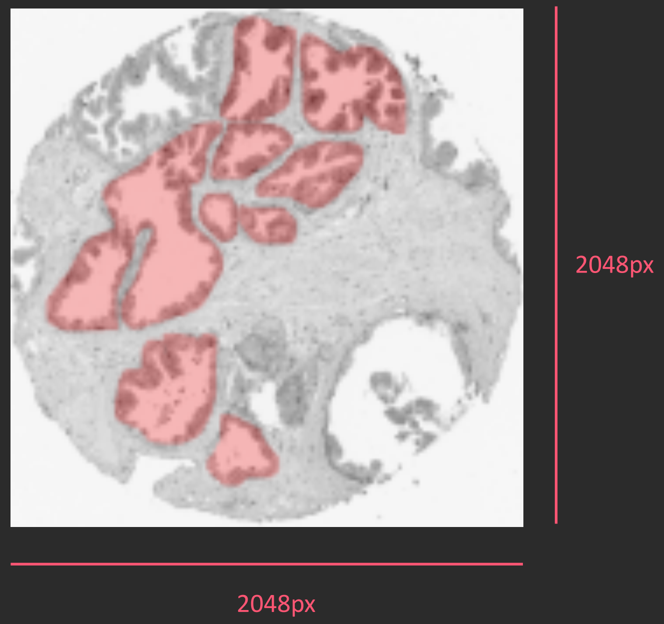
Sample Original Image
[3000*3000 image]



Remove Unnecessary BG Area



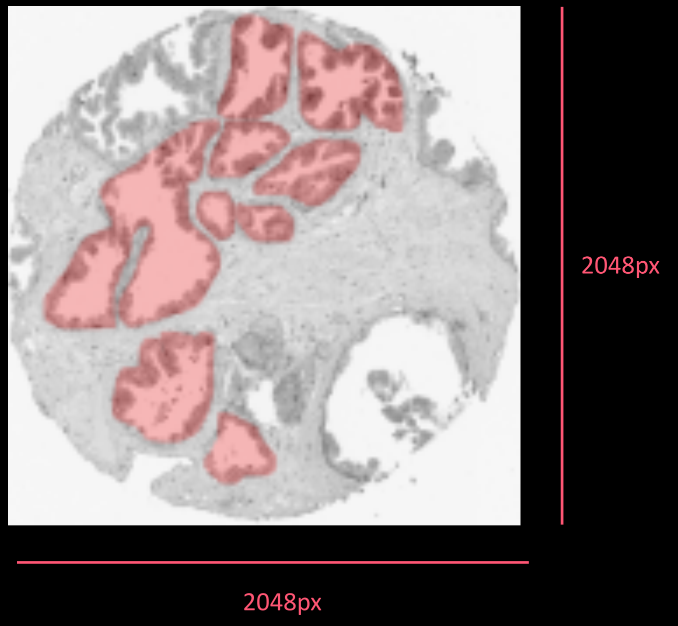
New Converting Meterial



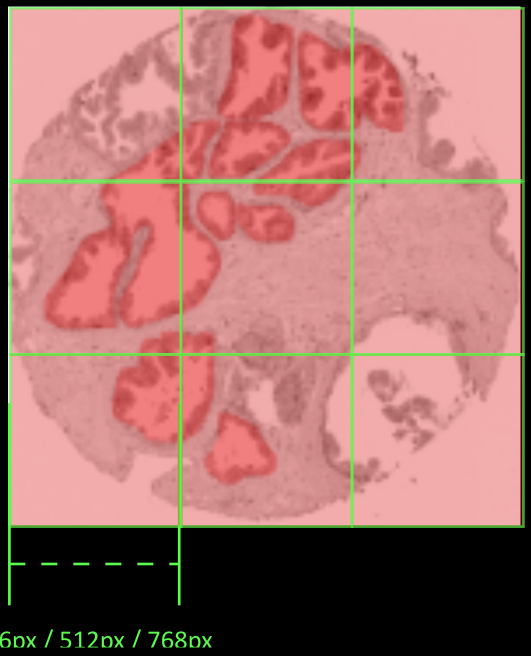
Project Detail

✓ **Converting** Removed BG Area Dataset

New Converting Meterial
[2048 *2048 image]



Converting 2048 Dataset



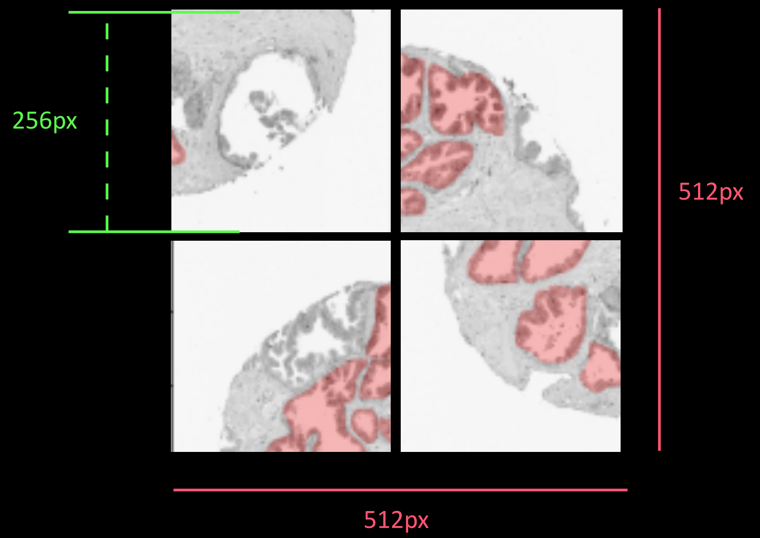
Converting Various Size
[256 / 512 / 768 px]

Project Detail

✓ Mosaic - 256px [256 size Converting Image Augmentation] — OFFLINE AUGMENTATION

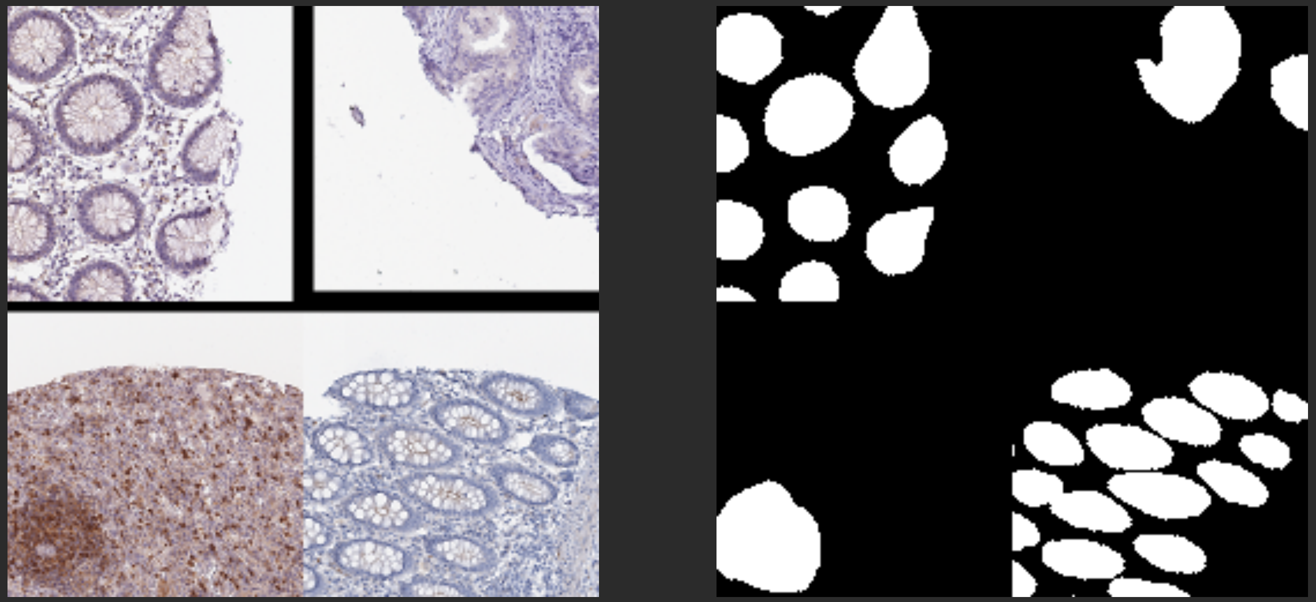
256 ▶ **512**

[Using Mosaic Method]



[Image Data]

[Mask Data]



Project Detail

✓ **Griddropout - 256px** [256 size Converting Image Augmentation] — ONLINE AUGMENTATION [Transform]

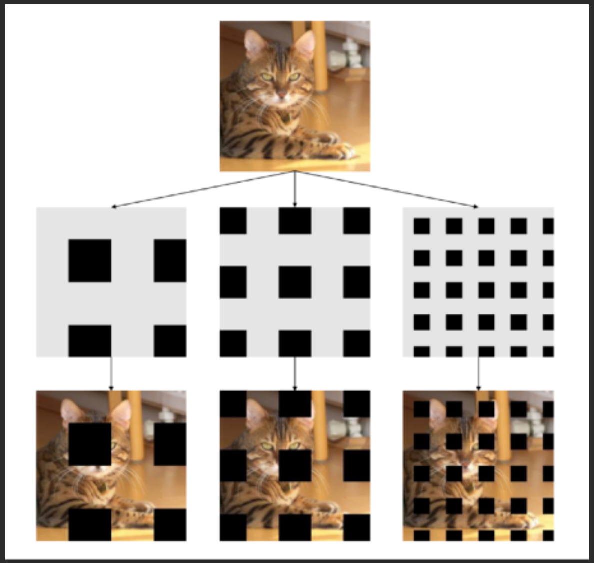
[Using Online Augmentation]

- Rotate [Random / Shift-Scale]
- Flip [Horizontal / Vertical]
- Gaussian Blur
- Random Brightness Contrast
- HueSaturationValue
- GaussNoise

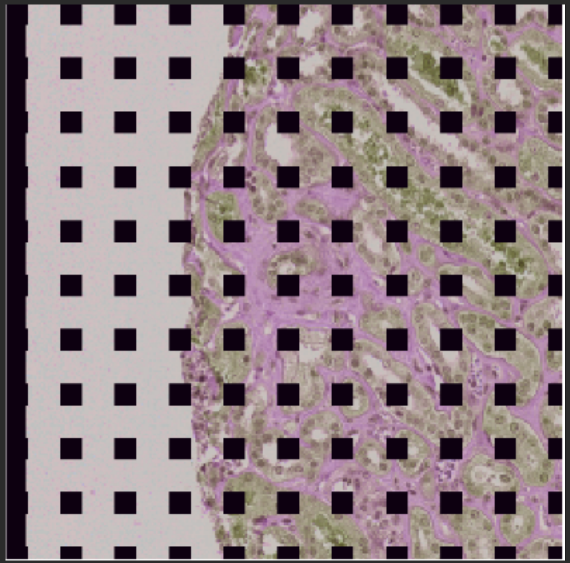
+

[Griddropout Method]

훈련 이미지의 일부 영역을 무작위로 Dropout / Blackout Method



[Sample Image]



[Our Data Making]

Project Detail

✓ Dataset Count [Current Using]

Count	Original Dataset [3000 * 3000]	Custom Dataset [2048 * 2048]
Converting 512 [Reduce : 1]	10271	5599
Converting 512 [Reduce : 2]	3115	1404
Converting 768 [Reduce : 1]	5373	3157
Converting 768 [Reduce : 2]	1404	-

Project Detail

Building DEV-ENV

✓ AWS ec2

Visual Studio Code

ANACONDA

ubuntu

Linux

papermill

ENV.	GPU	RAM	NOTES
Kaggle	tesla P100	16gb	1주일 40시간 / 하루 12시간
Colab	tesla P100 / T4	16gb	Colab Pro 기준 24시간 (보장 x)
AWS	A10G	24gb	T4보다 3.3 배 빠름

 # nohup — Terminal session이 종료되어도, 지속적으로 동작하게 하는 명령어

 # papermill — Jupyter Notebook으로 ML 모델링을 할 때, 원본 ipynb파일을 보존
[Copy 후, 특정값만 변경하여 실행할 수 있는 툴]

 # nvidia-smi — GPU 의 상태를 Monitering 할 수 있는 명령어

Project Detail

MMSegmentation **Trying Various Model**

✓ MMSEG - 1st Try [Using Base Model Only]

https://www.notion.so/MMSEG-3afb1a38ed904aaa8db5c0ce408a3038

Model	Dataset	Hyper Parameter	mDice	score
Segmenter (VisionTransformer)	Resize 512x512 Multi dataset	optimizer : AdamW, runner : EpochBasedRunner, lr : 1e-4, epoch : 300	44.72	-
Segmenter (VisionTransformer)	Resize 512x512 Multi dataset	optimizer : AdamW, eps : 1e-8, lr : 1e-6, weight decay : 1e-2, size : 512, epoch : 500	39.98	0.14
Unet (backbone-FCN)	Grid_512_BG_0%t Multi dataset	optimizer - Adam, lr - 1e-4, iter : 10000	72.94	-
Unet (backbone-FCN)	Convert_512 Multi dataset	optimizer - Adam, lr - 1e-4, iter : 20000	72.79	-
Unet (backbone-FCN)	Competition Original Dataset Resize 512	optimizer - Adam, lr - 1e-4, iter : 50000	74.16	-
DeepLabv3(encoder-Unet)	Resize 512x512 Multi dataset	epoch : 500	80.97	0.27
DeepLabv3+ (backbone-ResNetV1c)	Grid_512_BG_0% Multi dataset	iter : 80000	69.32	-
Segformer (Mix Transformer)	Grid_512_BG_0% Multi dataset	epoch : 300	70.38	-
Segformer (Mix Transformer-base 5)	Grid_512_BG_0% Multi dataset 512 to 256 resize	epoch : 500, pretrained-model using	62.03	-
UPerNet (SwinTransformer)	Grid_512_BG_0% Multi dataset	epoch : 20	77.45	-
UPernet (Vision Transformer-base 16)	Grid_512_BG_0% Multi dataset	epoch : 50	77.45	-
UPerNet (backbone-ResNet 50)	Grid_512_BG_0% Multi dataset	pretrained='open-mmlab://resnet50_v1c'	78.95	0.25
ERFnet (backbone-FCN)	Grid_512_BG_0% Multi dataset	pipeline = auto-augmentation(padding, resize) skip	72.6	0.18

✓ About FastAI

- 기본적으로 SOTA급 성능의 모델을 손쉽게 학습 될 수 있도록 도와주는 라이브러리 [ex. learning rate을 찾기위한 시스템적인 방법을 제공]

- fastai라는 이름대로 빠른 속도가 장점

- Vision / NLP / Recommend System 등에 사용하면 좋을만한 Default-Model 제공

- 단순 모델의 구현체가 아니라, 여러가지 테스트를 거쳐 최적의 파라미터로 구성된 모델

- 데이터 전처리 등에 있어서, 일반적으로 많이 사용되는 전처리를 손쉽게 수행해 주는 기능 일부 탑재

- 기존 구현한 딥러닝 코드를 단지 fastai로 컨버팅시, 트레이닝 속도 개선과 정확도 상승도 기대할 수 있다.

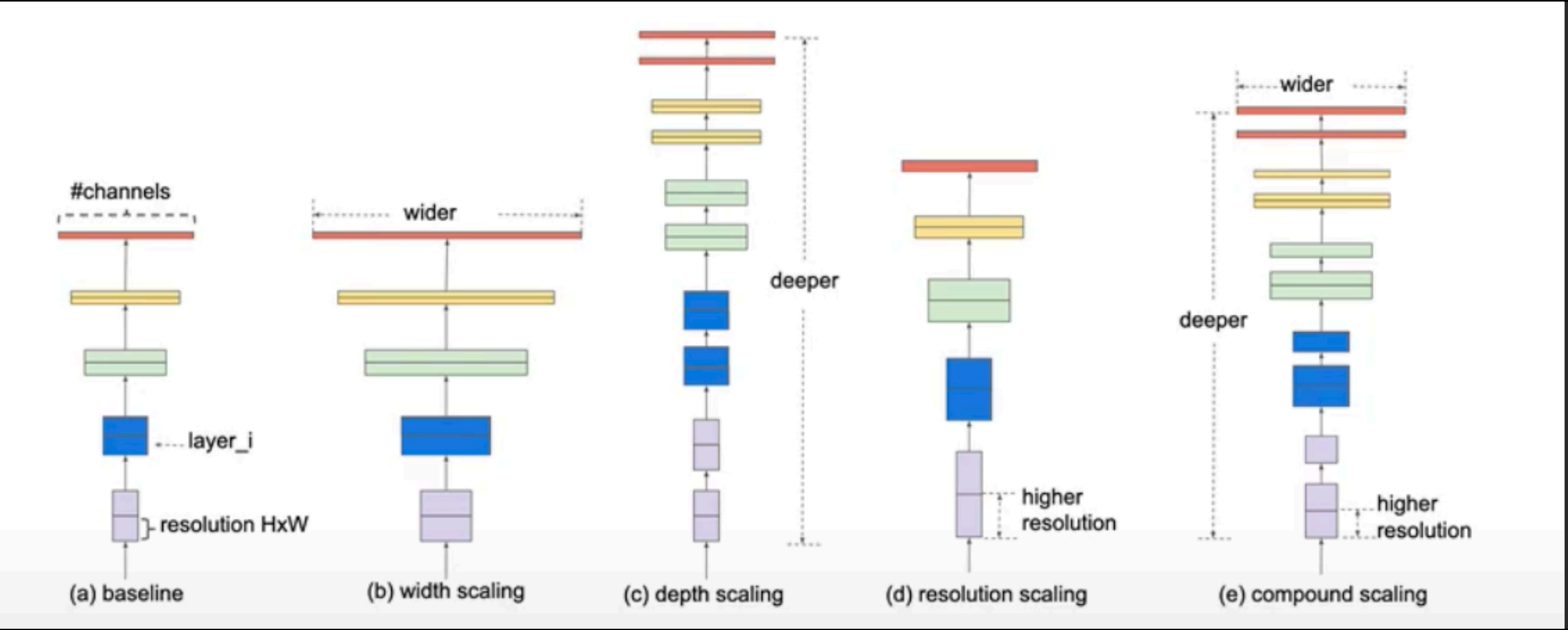
- Version 2까지 출시

- 각 Version별로 지원하는 함수가 다를수도 있다.

Project Detail

fast.ai Trying Various Model

✓ FastAI - EfficientNet



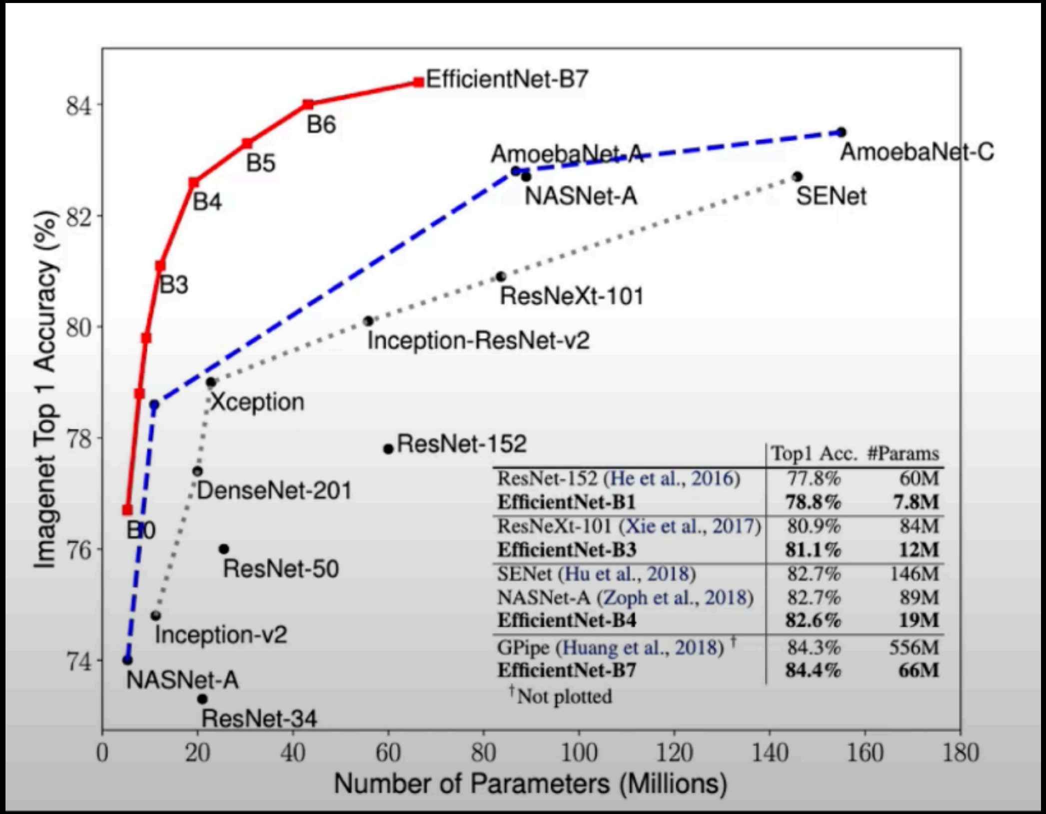
Project Detail

fast.ai Trying Various Model

✓ FastAI - EfficientNet

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	**77.1%**	**93.3%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.1%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**80.1%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.6%**	**95.7%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.9%**	**96.4%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.6%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.3%**	**97.0%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).



https://visionhong.tistory.com/19

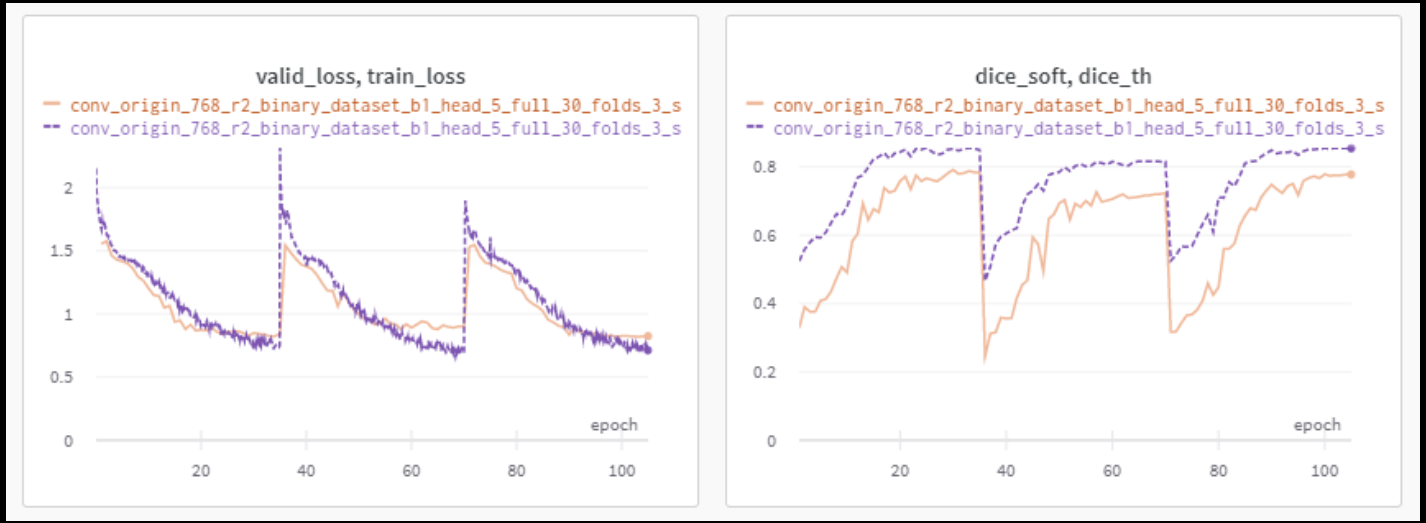
Result

fast.ai Try Info. Record Model

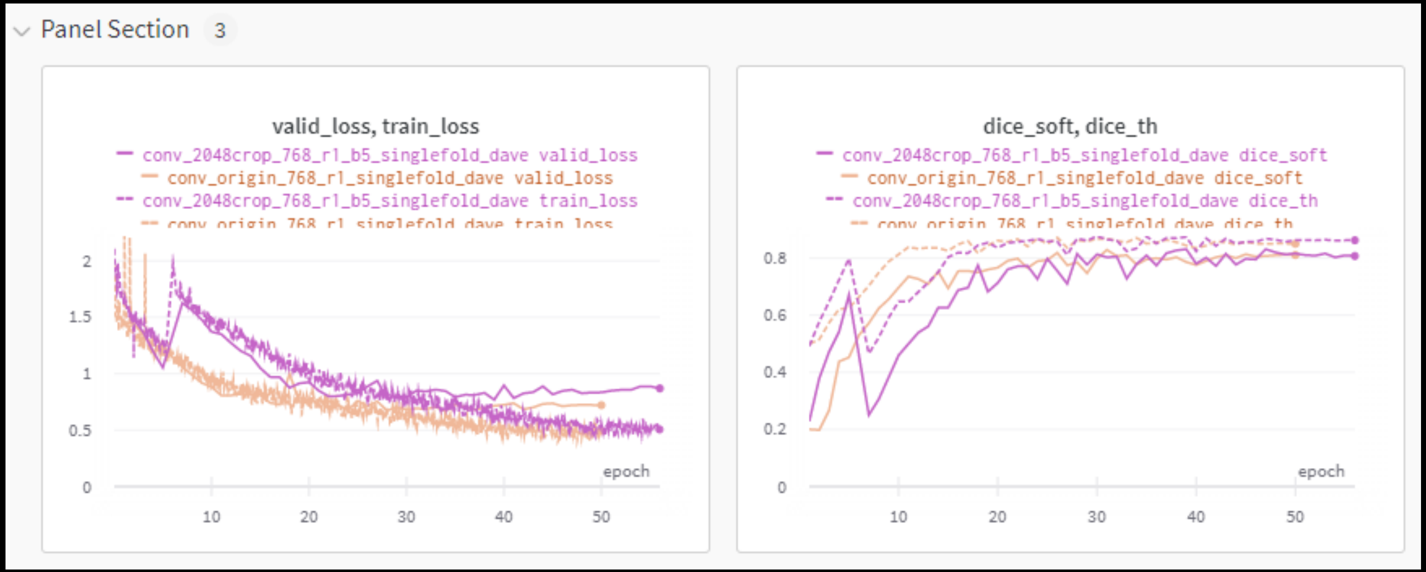
Model	Dataset	Setting	score
EfficientNet [b1]	Grid_256_BG30_Binary	batch_size = 64 size = 256 reduce = 4	0.30
EfficientNet [b1]	Mosaic_256		0.44
EfficientNet [b0]	Converting_256, Delete Albumentation(Augmentation)		0.51
EfficientNet [b5]	52 Grid_256 >>> Resize_512		0.52
UneXt50	Converting_256		0.56
EfficientNet [b3]	Converting_256		0.58
EfficientNet [b1]	Converting_256 [Try in Colab, Test]		0.59
EfficientNet [b0]	Converting_256		0.60
EfficientNet [b1]	Converting_256 + Griddropout = albumentation, save_model monitor = ‘dice_th'		0.61
EfficientNet [b1]	Converting_256 + Griddropout = albumentation, save_model monitor = ‘valid loss'		0.62
EfficientNet [b2]	Converting_256		0.63
EfficientNet [b1]	Converting_256		0.64
EfficientNet [b1]	Converting_origin_to_512_r1_binary_dataset	batch_size = 16 size = 512 reduce = 1	0.66

Result

Weights&Biases Try Info. Record Model



Original Data [3000 px]
Converting_768
reduce _ 2



Custom Data [2048 px]	Original Data [3000 px]
Converting_768	Converting_768
reduce _ 1	reduce _ 1
Efficient_b5	Efficient_b5
Singlefold	Singlefold

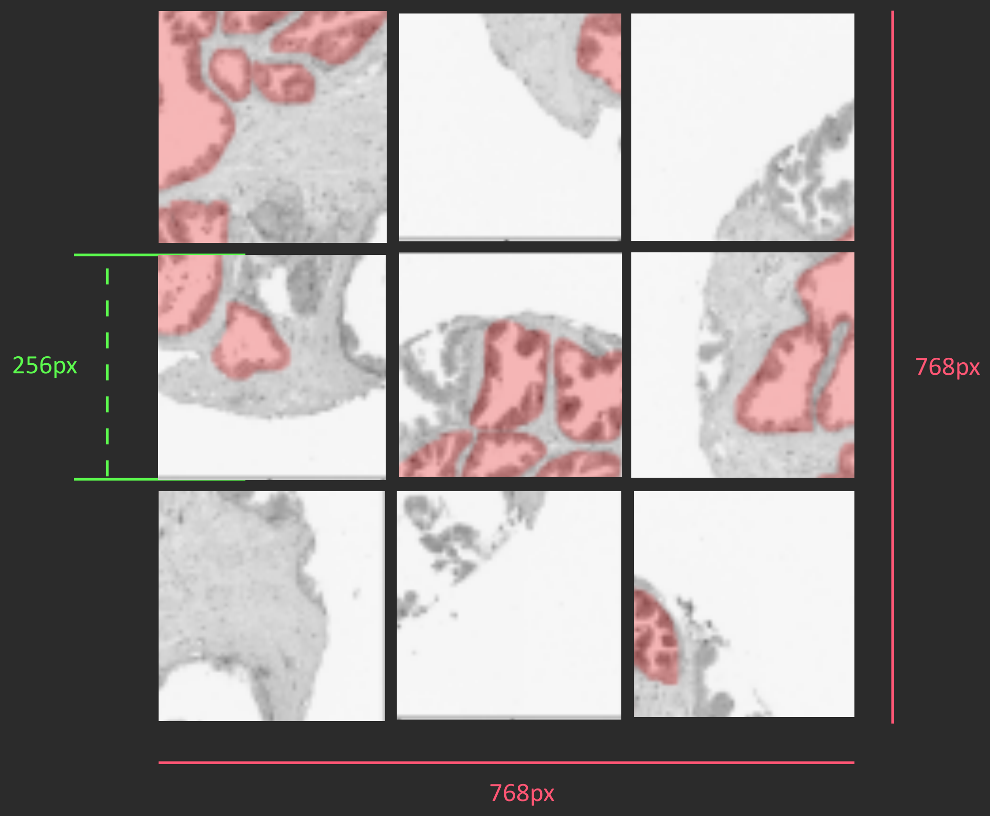
Future Plan & Opinion

Future Plan

- ✓ Training Mosaic Dataset 256 to 512 [FastAI]
- ✓ Making Mosaic Dataset 256 to 768 [Augmentation]
- ✓ CutMix Method [Augmentation]
- ✓ Copy & Paste Method [Augmentation]
- ✓ Hyper-Parameter [Develop Model]
- ✓ Result Analysis & Increasing Score [Kaggle LB]
- ✓ Try 2nd MMSEG Training & FastAI
- ✓ Making Padding Dataset [Pre-Processing]

256 to 768

[Using Mosaic Method]



Self-Assessment Opinion

1. 공모전 참가 이전 Segmentation Project [CAR DAMAGE DETECTION] 을 기반으로 좀 더 다양한 Task를 수행

2. 차량 Segmentation Task 와 차이점

   - CAR DAMAGE DETECTION : 차량 파손 데이터 / Multi Class + Multi Label / Binary Class
   - HuBMAP : 의료데이터 / Multi Class + Binary Classifier

3. CAR DAMAGE DETECTION 프로젝트 경험을 바탕으로 도전하고있는 공모전이지만, 방법과 종류의 차이

   - 다양한 모델 튜닝 방법 고안
   - 각 Model, Dataset, Hyper-Parameter등 실험 결과 정보 정리와 분석 및 계획 필요

4. 제출 점수 중 가장 높은 score 는 66점으로, 1차 목표인 상위 5-10% 안에 들어가기 위해 다양한 방법으로 접근 방법 필요

CODEnter

장주찬 | https://github.com/jcjang1

윤소미 | https://github.com/YoonSoM

전규원 | https://github.com/GUUNNIA

한정현 | https://github.com/Riverskin1

허 권 | https://github.com/heokwon

박정현 | https://github.com/dotsnangles