

PANDA competition writeups

poteman & arutema & fam_taro(1st /1010teams)

Agenda

1. About our team
2. Challenges
3. Our solution
 1. Summary
 2. Denoising method
 3. Data preparation
 4. Model setup
 5. Inference pipeline



Team PND

- Public: 22nd (0.910)
- **Private: 1st (0.940)**
- Members:
 - arutema47, Japan, kyoshioka47@gmail.com
 - famtaro, Japan,
 - poteman, China,

Our slack icon



Background of our team members..

- **arutema47**

What is your academic/professional background?

Ph.D EE.

Did you have any prior experience that helped you succeed in this competition?

Bengali-cv19

What made you decide to enter this competition?

- Computer vision.
- Was interested in medical microscope images

How much time did you spend on the competition?

- About 2 hours per day * 4 weeks

If part of a team, how did you decide to team up?

- I knew famtaro from twitter and got an e-mail from poteman. Wanted to team up with Kaggle Master

If you competed as part of a team, who did what?

- Worked on denoising the labels, read papers, trained model and discussed on slack.

Background of our team members..

- **Famtaro**

What is your academic/professional background?

Master of CS.

Did you have any prior experience that helped you succeed in this competition?

<https://www.kaggle.com/c/siim-acr-pneumothorax-segmentation>

What made you decide to enter this competition?

- CV
- Medical
- Academic

How much time did you spend on the competition?

- About 240 hours

If part of a team, how did you decide to team up?

- I knew arutema on twitter.

If you competed as part of a team, who did what?

- fam_taro: Make model, Survey of papers about denoise, discussion on slack.

Background of our team members..

- **Poteman**

What is your academic/professional background?

Master of CS.

Did you have any prior experience that helped you succeed in this competition?

no

What made you decide to enter this competition?

- This is a cv problem.
- I want to get the gold medal and to be a grandmaster.

How much time did you spend on the competition?

- About 2-3 hours per day * 3 weeks

If part of a team, how did you decide to team up?

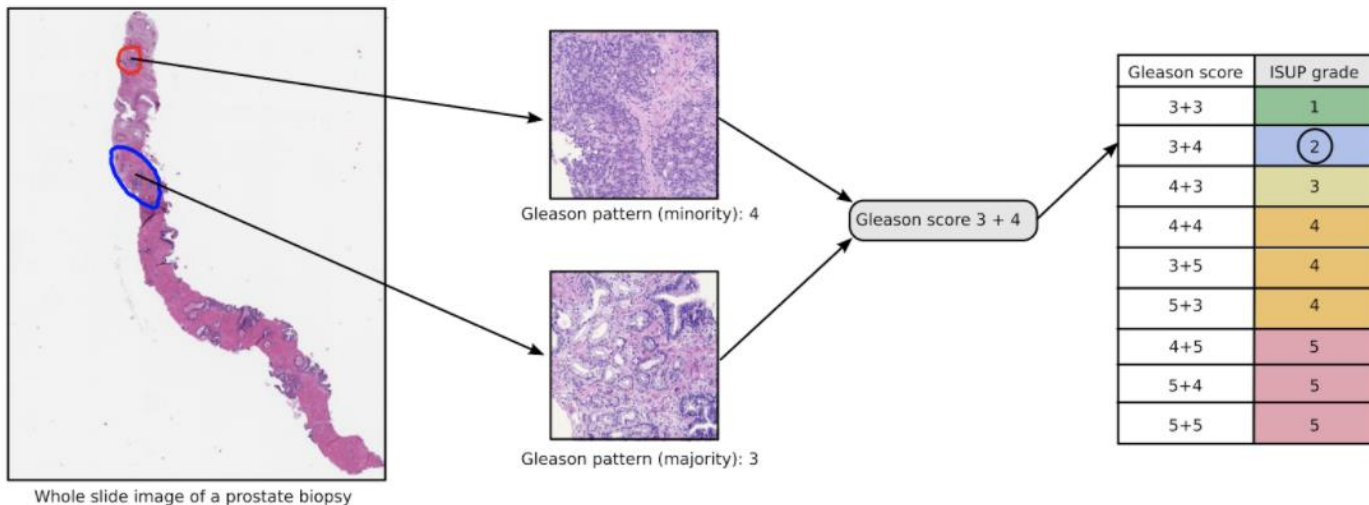
- I contacted arutema with email.

If you competed as part of a team, who did what?

- Process data, make model, read some papers, and discussion on slack with teammates.





Competition Challenge : WSI images

- Predict ISUP grade score from WSI(Whole Sliding Image)
 - ISUP grade $\hat{=}$ Risk of prostate cancer
 - (No cancer) 0 \leftrightarrow 5 (High risk cancer)
- WSI from prostate tissue biopsies
- Raw WSI is too big for humans to see. (10,000 x 10,000 ~)



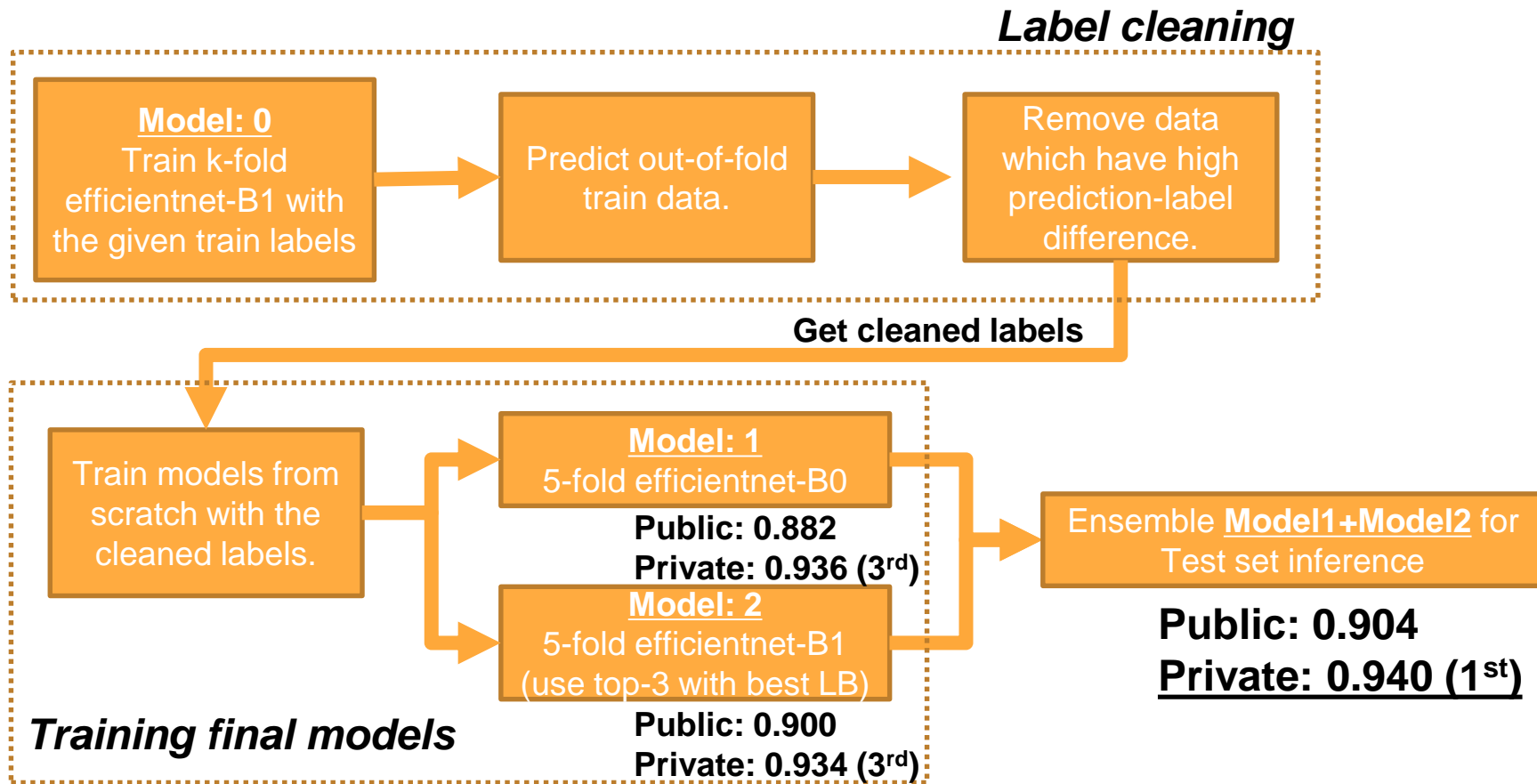
Competition Challenge: Labeling Noise

- Annotator for this competition data (written on official document...)

	Data Provider	
	Karolinska	Radboud
Train	1 Expert 	Noisy labels are the largest challenge in this competition. Trained students judge from diagnostic report (I don't know how many students...) 
	3 Experts (and 1 expert same at train) 	3 Experts 
Test		

At Radboud, if students **predict test data**,
Acc : **0.720**
QWK: **0.853**
→ **Radboud train label noise may be larger than Karolinska's**

Our Solution summary



Our Label Cleaning method

- Simple, yet effective label cleaning method
 - **Sets us apart from others!**
- Remove data based on the gap between the hold-out prediction results and the given label
 - **Idea: Large prediction gap mean: 1) wrong label, 2) difficult data**
 - This method excludes both (1)+(2), the model will be weak against difficult data, but strong against easy data.
 - E.g. Predicted ISUP = 4.1, Label ISUP = 4 **gap = 0.1** and data is kept
 - E.g. Predicted ISUP = 0, Label ISUP = 4 **gap = 4** data is removed



Our Label Cleaning method

- Remove data based on the prediction and the label gap and get cleaned labels.
 - Gap Threshold = 1.6
 - remove ratio[%]: 5.614167294649586
 - total number of removed data: 596
 - total number of removed Radboud : 445
 - total number of removed Karolinska : 151

```
# Base arutema method
def remove_noisy(df, thresh):
    gap = np.abs(df["isup_grade"] - df["probs_raw"])
    df_removed = df[gap > thresh].reset_index(drop=True)
    df_keep = df[gap <= thresh].reset_index(drop=True)
    return df_keep, df_removed

df_keep, df_remove = remove_noisy(df, thresh=1.6)
show_keep_remove(df, df_keep, df_remove)
```

5.6% of training data was removed.

More Radboud data removed

-> matches that Rad. has more label noise (students labeled)!

Our Label Cleaning method

- Remove data based on the prediction.
 - Gap Threshold = 1.6
 - remove ratio[%]: 5.614167294649586
 - total number of removed data: 596
 - total number of removed Radboud : 445
 - total number of removed Karolinska : 151

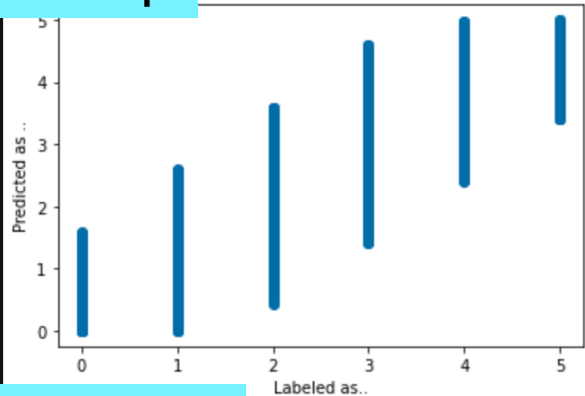
5.6% of training data was removed

More Radboud data removed

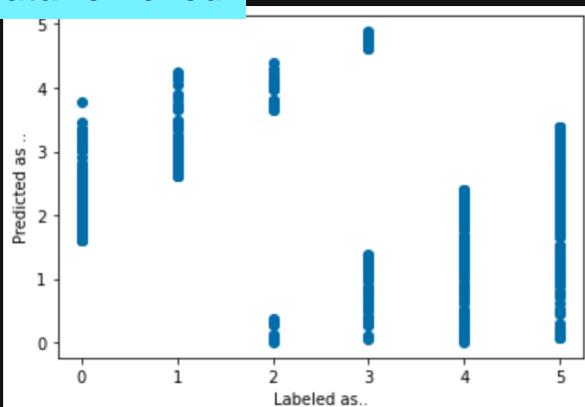
-> matches that Rad. has more label noise (s

```
remove ratio[%]: 5.614167294649586
number of reduced: 596
number of reduced radboud : 445
number of reduced karolinska : 151
```

Data kept



Data removed



Further cleaning (used for Model:2)

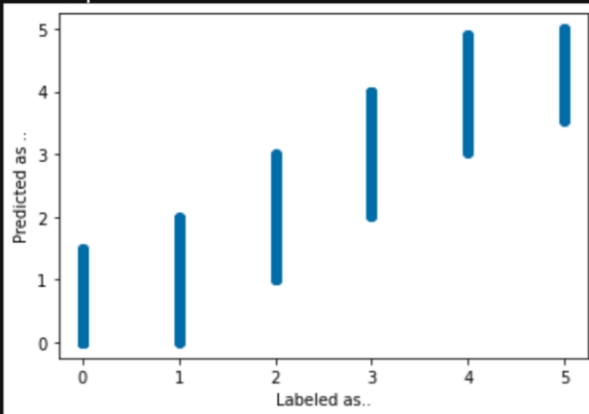
- Remove noise

- Change gap threshold for each label for each data provider.
- Threshold was set to remove 20% of Radboud data

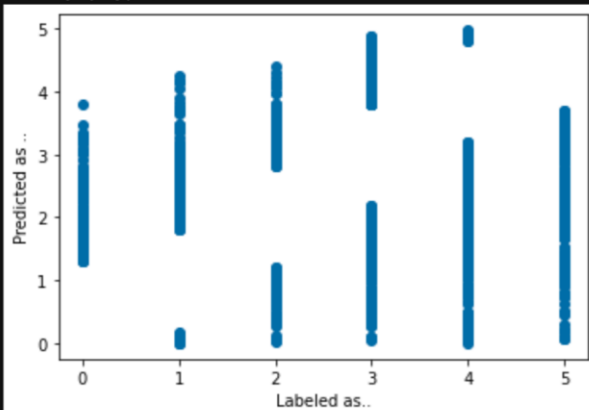
```
def remove_noisy3(df, thresholds_rad, thresholds_ka):  
    print(f' threshold_rad: {thresholds_rad}')  
    print(f' threshold_ka : {thresholds_ka}')  
    df_r = df[df.data_provider == "radboud"].reset_index(drop=True)  
    df_k = df[df.data_provider != "radboud"].reset_index(drop=True)  
  
    dfs = [df_r, df_k]  
    thresholds = [thresholds_rad, thresholds_ka]  
    df_keeps = list()  
    df_removes = list()  
  
    for df_tmp, thresholds_tmp in zip(dfs, thresholds):  
        df_keep_tmp, df_remove_tmp = remove_noisy2(df_tmp, thresholds_tmp)  
        df_keeps.append(df_keep_tmp)  
        df_removes.append(df_remove_tmp)  
  
    df_keep = pd.concat(df_keeps, axis=0)  
    df_removed = pd.concat(df_removes, axis=0)  
    return df_keep, df_removed  
  
# Change thresh each label each dataprovider  
thresholds_rad=[1.3, 0.8, 0.8, 0.8, 0.8, 1.3]  
thresholds_ka=[1.5, 1.0, 1.0, 1.0, 1.0, 1.5]
```

```
remove ratio[%]: 14.016578749058025  
number of reduced: 1488  
number of reduced radboud : 1153  
number of reduced karolinska : 335
```

**** keep ****



**** removed ****



Ablation study of label cleaning

Model 2-like performance trained with / without cleaned labels
(Final version has slight modifications)

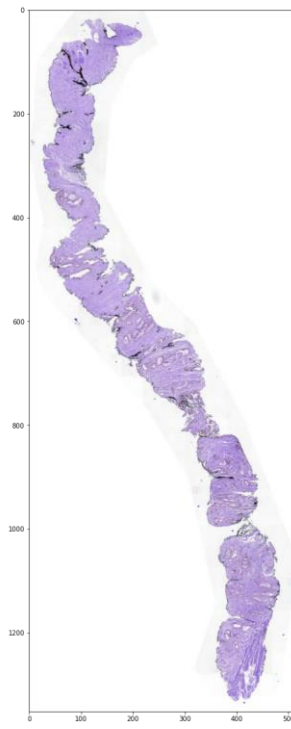
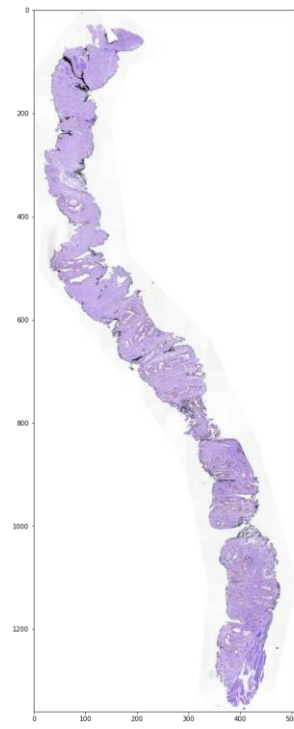
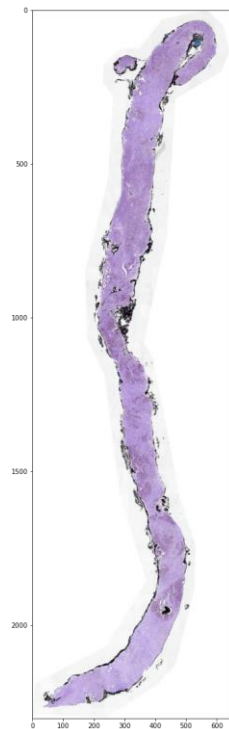
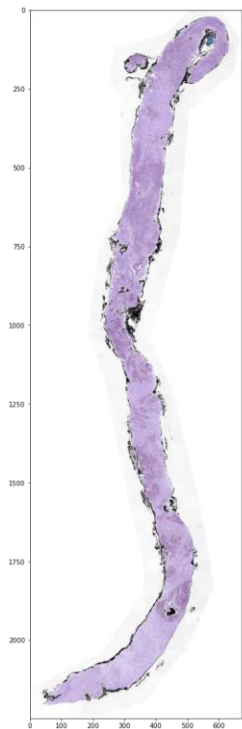
	Public	Private
Without Cleaned Labels	0.892	0.916
With Cleaned Labels	0.901	0.932

Dataset setup

- **Tiling method based on iafoos public kernel**
 - We generate [32x256x256] and [64x256x256] tiled images with this.
 - <https://www.kaggle.com/iafoos/panda-16x128x128-tiles>
- **Data split based on image hash**
 - Careful so that duplicated images will not be spread across validation sets.
 - This will makes the CV more stable.

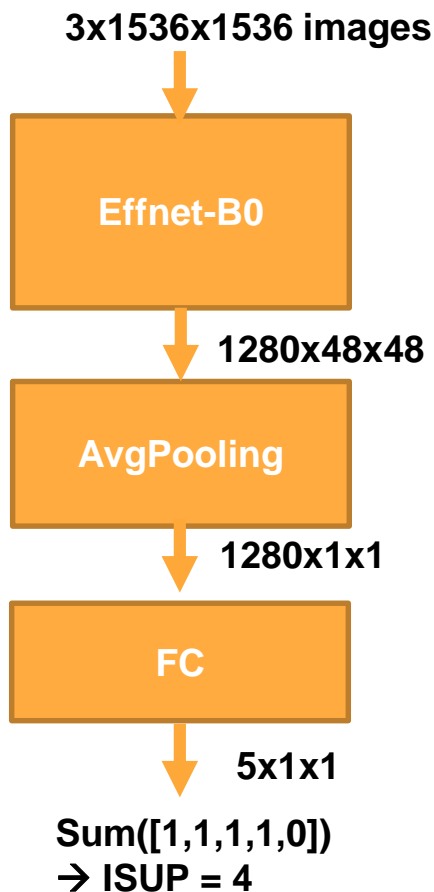
Additional Challenges: Duplicated Data

- Same biopsy, but a different slice. (Estimated 500-1,500 duplicate images)
 - Can detect duplicate data by image hash
 - <https://www.kaggle.com/c/prostate-cancer-grade-assessment/discussion/155954>



Model setup (1/2)

- Our model structure and loss is based on public kernels
 - <https://www.kaggle.com/haqishen/train-efficientnet-b0-w-36-tiles-256-lb0-87>
- **Model 1:**
 - Backbone: EfficientNet-B0, pooling: avg_pooling
 - Tile: 32x256x256
 - Augmentations (e.g. cutout, mixup) used for generalization
 - Cosine annealing schedule for 20 epochs
- Larger backbones introduced overfitting (e.g. resnext...)



Model setup (2/2)

- Model 0: Before denoise, used for denoising
- Model 2: After denoise
- Some parts that differ from Model 1
 - Tilesize 192, Tilenum 64 (image size: $192 \times 8 = 1,536$)
 - Model: EfficientNet B1 + GeM at head
 - Predict ISUP + first gleason score during training (10 dimension outputs)
 - Predicting gleason score enables faster training and some improvements in LB.
 - Note that **only predicted ISUP** is used for test inference.

Common config

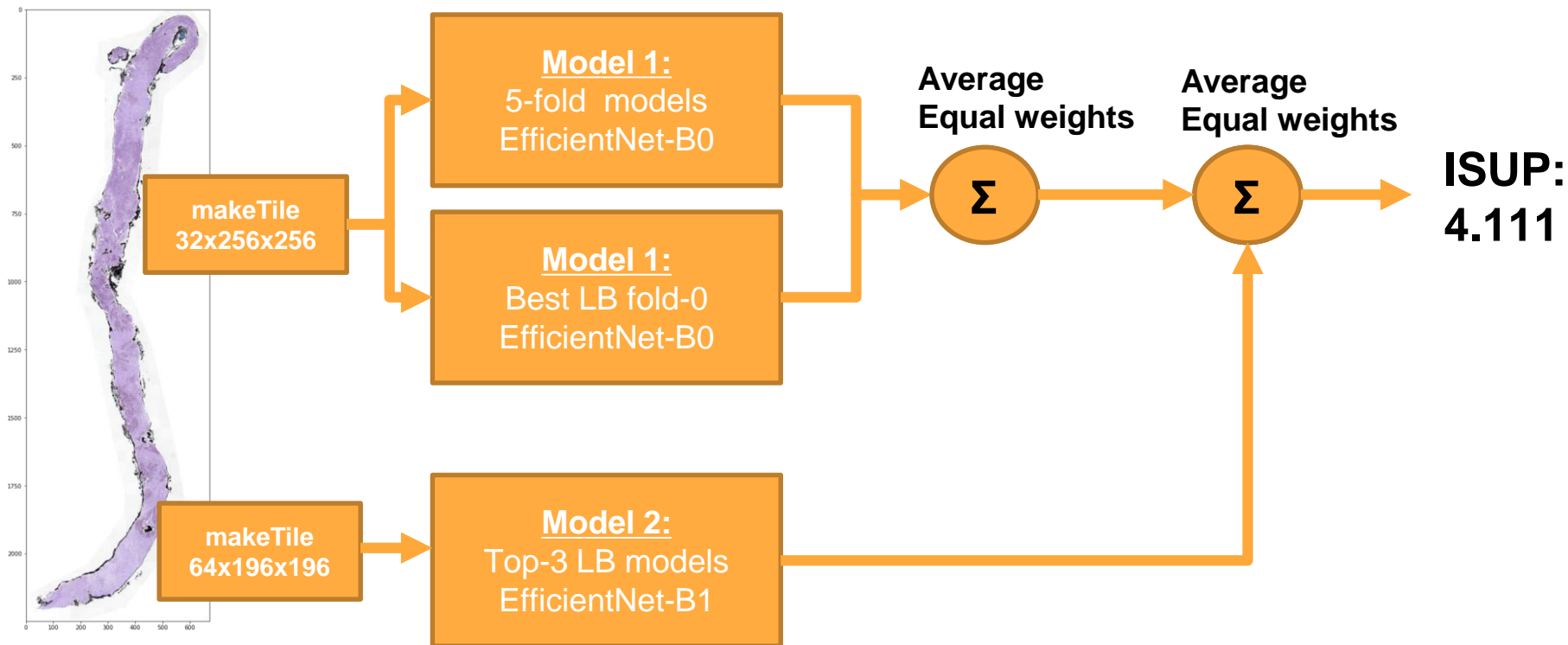
```
General:
  fp16: True
  amp_level: 01
  multi_gpu_mode: ddp
  epoch: &epoch 30
  grad_acc: 2
  frozen_bn: False

Data:
  data_loader:
    batch_size: 6
    num_workers: 4

Optimizer:
  optimizer:
    name: Adam
    params:
      # 10 times on epoch 0 by warmup scheduler
      lr: !!python/float 3e-5
      amsgrad: False
  lr_scheduler:
    name: CosineAnnealingLR
    params:
      T_max: *epoch
      last_epoch: -1

Loss:
  base_loss:
    name: BCEWithLogitsLoss
```

Inference pipeline ~Ensemble~



Model execution time

- Training
 - Model 0: 75 hours (5 fold) @TitanRTX
 - Model 1: 24 hours (5 fold) (using png tiled images) @RTX2080ti
 - Model 2: 65 hours (5 fold) @TitanRTX
- Inference
 - ~4 hours on Kaggle Notebook +GPU

Discussions: Why did we win?



- Assumption 1: **Private data contain more easy data than Public**
 - We could get good score because our model is strong to easy data (but weak to difficult data)
 - Cons: Our denoise removes difficult data as well
 - Our model accuracy is better than the student annotator!
 - Similar story.. “Are we done with ImageNet?” <https://arxiv.org/abs/2006.07159>
 - ImageNet labels have lots of noise (5-10%): actually the model prediction was right than the annotator in most cases!
- Assumption 2: **Splited kfold with imghash (considering duplicates)**
 - Having duplicate images across folds causes leak in the cross-validation set.
 - We treat these duplicate images properly and place them under the same folds.
 - Some people in the discussions said that the score changes largely by their “random seed”, this is because of this data leakage!
 - Our pipeline can reproduce 1st place score with different seed settings.