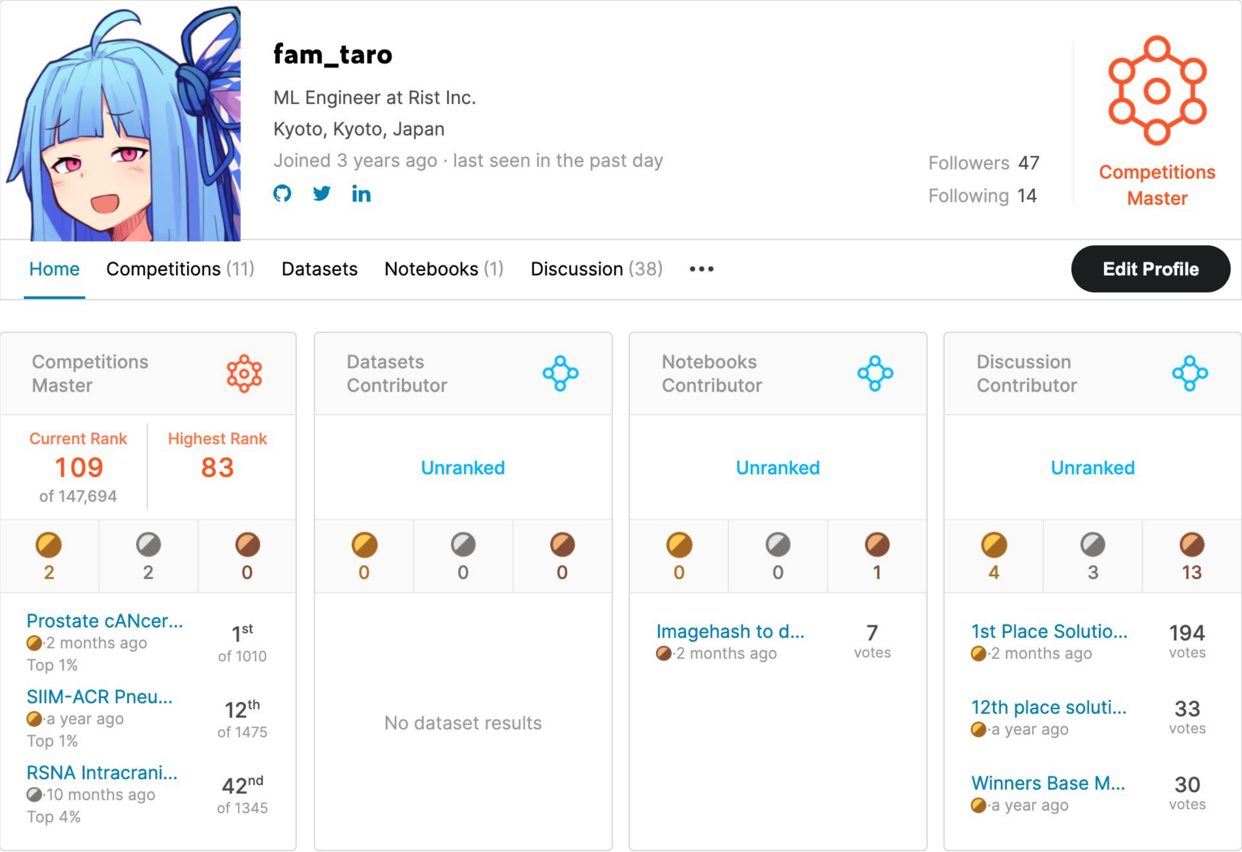
PANDA competition 1st solution

fam_taro(1st / 1010teams)

About me


fam_taro
ML Engineer at Rist Inc.
Kyoto, Kyoto, Japan
Joined 3 years ago · last seen in the past day
Followers 47
Following 14
Competitions Master
Home
Competitions (11)
Datasets
Notebooks (1)
Discussion (38)
Edit Profile
Competitions Master
Current Rank
109
of 147,694
Highest Rank
83
2
2
0
Prostate cANcer...
·2 months ago
Top 1%
1st
of 1010
SIIM-ACR Pneu...
·a year ago
Top 1%
12th
of 1475
RSNA Intracrani...
·10 months ago
Top 4%
42nd
of 1345
Datasets Contributor
Unranked
0
0
0
No dataset results
Notebooks Contributor
Unranked
0
0
1
Imagehash to d...
·2 months ago
7
votes
Discussion Contributor
Unranked
4
3
13
1st Place Solutio...
·2 months ago
194
votes
12th place soluti...
·a year ago
33
votes
Winners Base M...
·a year ago
30
votes

Introduction: Our team

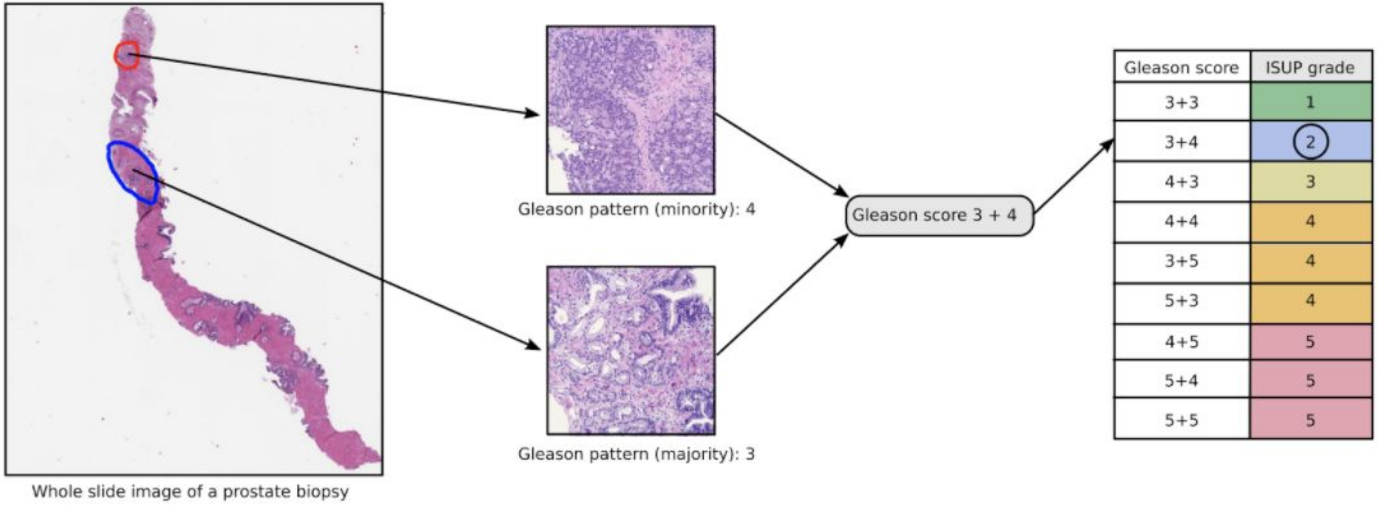
- Result of challenge
  - Public: **22nd (0.910)**
  - Private: **1st (0.940)**
- About team PND
  - arutema47, twitter@arutema47
  - fam_taro, twitter@fam_taro
  - poteman

Agenda

1. About Competition
2. Basic approach
3. Why Local CV ≠ Public LB? @ 🐼
4. Our solution
5. Conclusion
6. Appendix: Not work for me

1. About Competition: Task

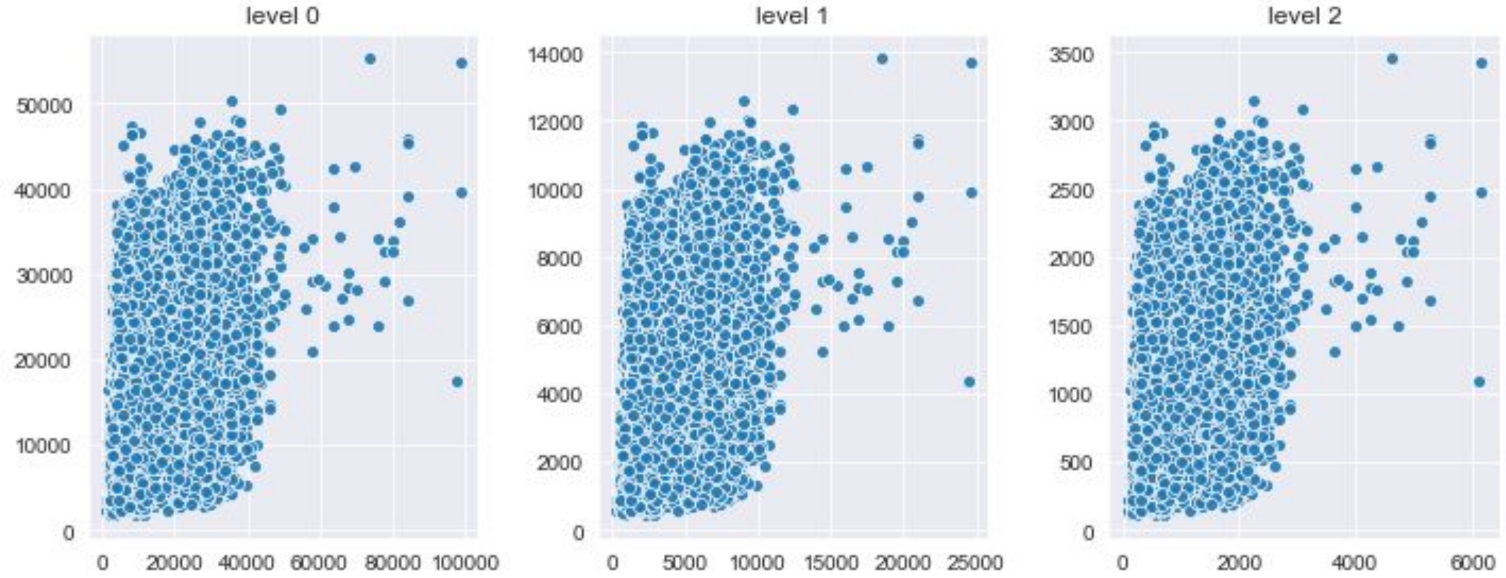
- Predict ISUP grade score from WSI(Whole Sliding Image)
  - ISUP grade ≒ Risk of prostate cancer(前立腺がん)
    - (No cancer) 0 ↔ 5 (High risk cancer)
- WSI from prostate tissue biopsies(前立腺組織生検)
- Raw WSI is too big for humans to see. (10,000 x 10,000 ~)



https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview

1. About Competition: Whole Sliding Image(WSI)

- Each provided WSI file has 3 scales (choose the scale and load image)
    - level 0, 1, 2 (16x, 4x, 1x)
    - x=width, y=height
- Very large
    - Many player used level 1 WSI

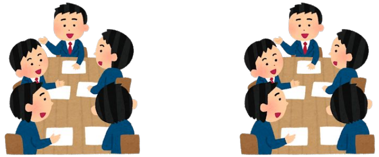
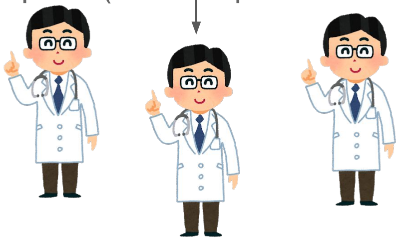
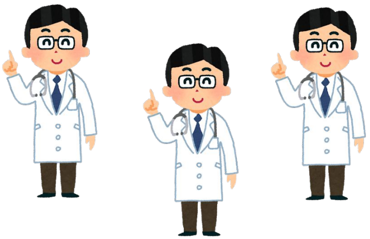


1. About Competition: Data

- Provided data
  - train.csv
    - image_id, data_provider, isup_grade(target), gleason_score
  - train_images(WSIs)
  - train_masks
    - Segmentation masks based on gleason by hosts model for each data provider
  - test.csv
    - image_id, data_provider
- Data Count
  - Train: 10k
    - Data provider(Karolinska:Radboud ≒ 1:1)
  - Test: About 940 ? (Public:Private ≒ 42:58)
    - Private test ≒ 545 (small…😢), Public test ≒ 395 (more small...😭)
      - https://www.kaggle.com/c/prostate-cancer-grade-assessment/discussion/158687

1. About Competition: Data

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

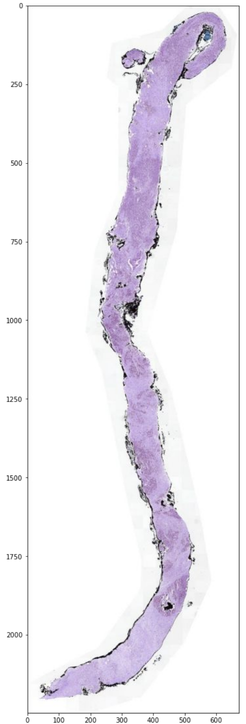
	Data Provider	
	Karolinska	Radboud
Train	1 Expert 	Trained students judge from diagnostic report (I don't know how many students…) 
Test	3 Experts (and 1 expert same at train) 	3 Experts 

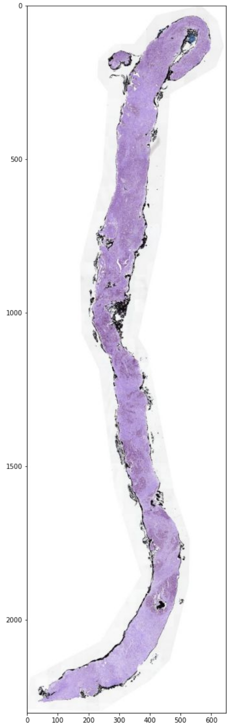
At Radboud, if students predict test data,
Acc : 0.720
QWK: 0.853

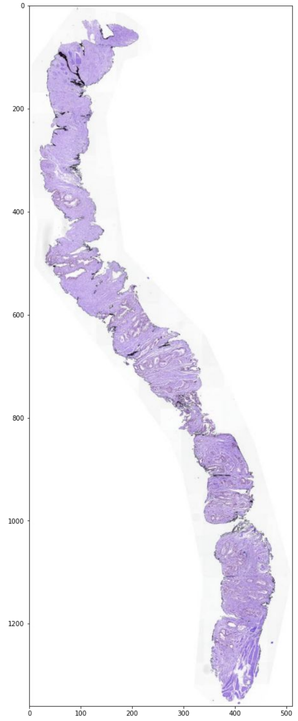
→ Radboud train label noise may be larger than Karolinska's

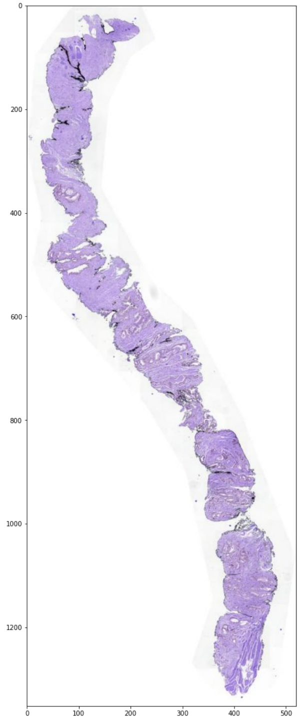
1. About Competition: Duplicated Data

- Same biopsy, but a different slice. (Estimated 500-1,500 duplicate images)
  - https://www.kaggle.com/c/prostate-cancer-grade-assessment/discussion/155954
  - Host say “This only holds for the training set.” (but not give us information about duplicated image ids)



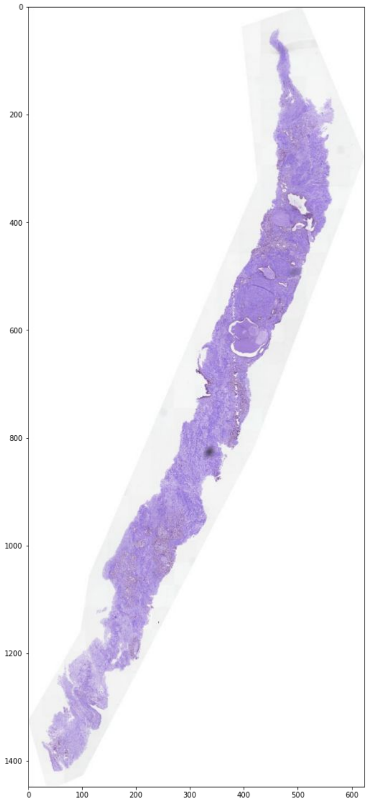


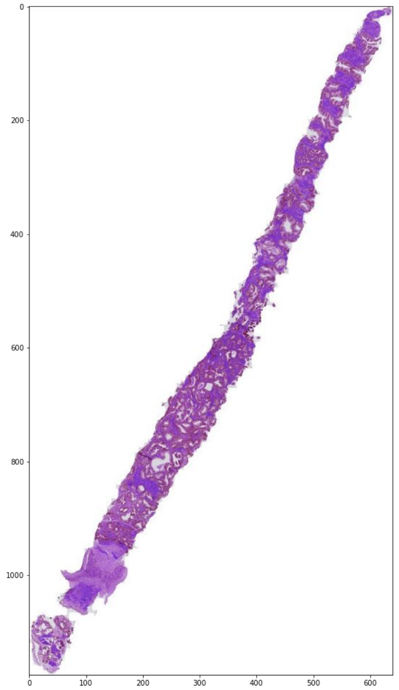


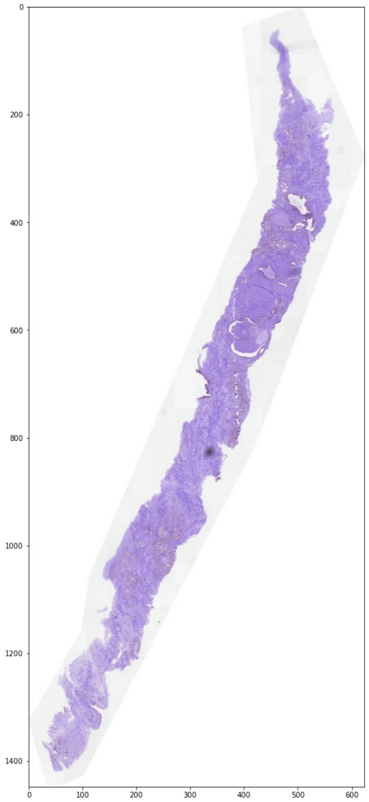


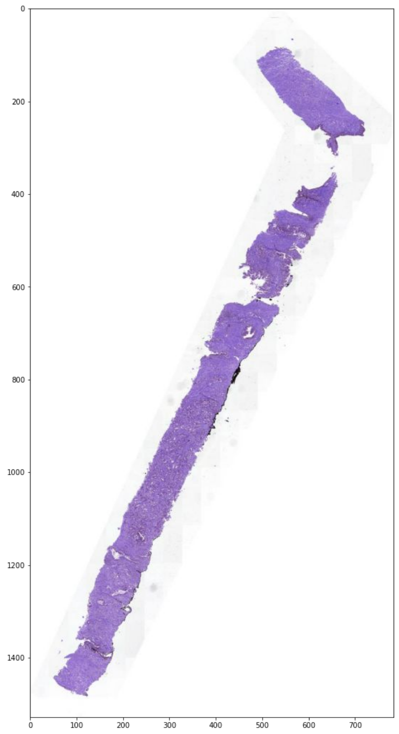
Appendix: Not correct grouping by Imghash

- Imagehash threshold: 0.9
  - https://www.kaggle.com/yukkyo/imagehash-to-detect-duplicate-images-and-grouping









1. About Competition: Metrics

- QWK: Quadratic Weighted Kappa
  - https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview/evaluation

Submissions are scored based on the quadratic weighted kappa, which measures the agreement between two outcomes. This metric typically varies from 0 (random agreement) to 1 (complete agreement). In the event that there is less agreement than expected by chance, the metric may go below 0.

The quadratic weighted kappa is calculated as follows. First, an N x N histogram matrix *O* is constructed, such that *Oi,j*
corresponds to the number of `isup_grade`s *i* (actual) that received a predicted value *j*. An *N-by-N* matrix of weights, *w*,
is calculated based on the difference between actual and predicted values:

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2}$$

An *N-by-N* histogram matrix of expected outcomes, *E*, is calculated assuming that there is no correlation between values.
This is calculated as the outer product between the actual histogram vector of outcomes and the predicted histogram vector, normalized such that *E* and *O* have the same sum.

From these three matrices, the quadratic weighted kappa is calculated as:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}.$$

1. About Competition: LB

- LB Line
  - Public
    - Bronze: 0.892, Silver: 0.901, Gold: 0.914
  - Private
    - Bronze: 0.917, Silver: 0.923, Gold: 0.929
      - Private is assumed to have fewer hard example
- Shake at private…
  - Larger than APTOS2019…
  - Smaller than M5(accuracy)

#	△pub	Team Name	Notebook	Team Members	Score	Entries
1	▲ 21	PND			0.94085	105
2	▲ 3	Save The Prostate			0.93768	263
3	▲ 188	Mikhail Druzhinin			0.93480	14
4	▲ 3	NS Pathology			0.93399	243
5	▲ 42	Kiminya			0.93283	34
6	▲ 11	BarelyBears			0.93260	229
7	▲ 70	ctrasd123			0.93245	131
8	▲ 19	ChienYiChi	</> Tile Model Ensemble		0.93238	159
9	▲ 282	Shelldragoon1104			0.93162	70
10	▼ 8	vanda			0.93032	181
11	▼ 3	lafoss			0.93009	111
12	▲ 71	Manuel Campos			0.92960	45
13	▲ 13	Blue Jeans [ods.ai]			0.92939	63
14	▼ 4	gakki			0.92921	49
15	▲ 90	BabaCondaBoko			0.92857	29
16	▲ 24	IJF			0.92845	211
17	▲ 121	Dmitry A. Grechka			0.92828	14
18	▲ 95	KovaLOVE v2	</> PANDA Inference w...		0.92770	12
19	▼ 16	Aksell			0.92741	260
20	▲ 196	andrekos			0.92732	5

1. About Competition: Data

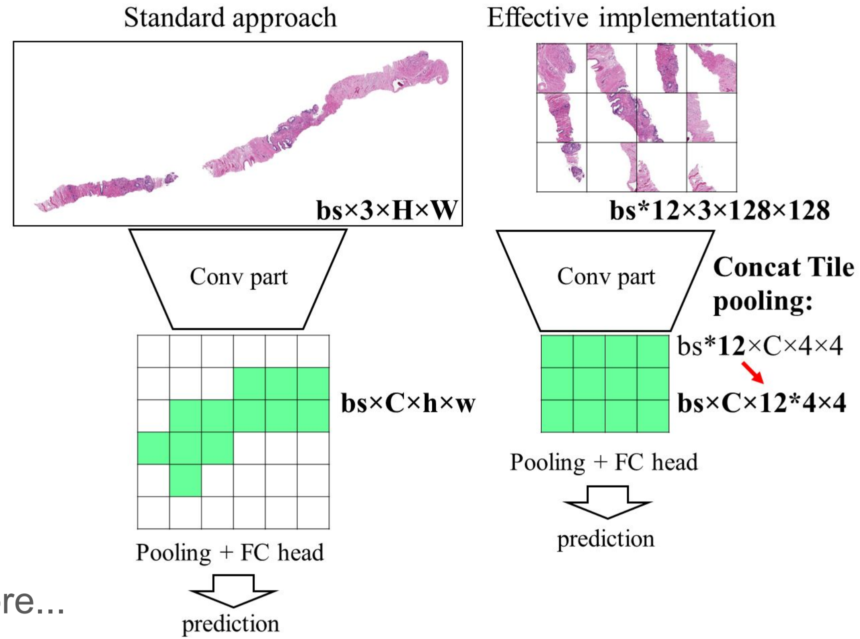
- Data Aspects (We can't take everything into account…)
  - Data provider (Karolinska or Radboud)
  - Noisy label or Not
  - Duplicated image or Not
  - Easy or Hard example (I didn't notice it during the competition…)
    - Perhaps this is what made us big shakeup
      - Private LB scores higher overall than Public LB's
        - Private test hard example may be less than Public test
    - Assumption
      - Our model is strong to easy example, but weak to hard example
        - Because our removing noise method (using gap between pred and original label of oof) is easy to remove hard example
        - This led to a divergence between Public and Private

1. About Competition: Code competition

- https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview/code-requirements
  a. CPU Notebook <= 9 hours run-time
  b. GPU Notebook <= 6 hours run-time
  c. TPUs will not be available for making submissions to this competition. You are still welcome to use them for training models.
  d. No internet access enabled
  e. External data, freely & publicly available, is allowed. This includes pre-trained models.
  f. No custom packages enabled in kernels
     i. 🤔
  g. Submission file must be named `submission.csv`

2. Basic approach

- Iafoss tile method & Qishen Ha bin label
  - Kernels
    - https://www.kaggle.com/iafoss/panda-concat-tile-pooling-starter-0-79-lb
    - https://www.kaggle.com/haqishen/train-efficientnet-b0-w-36-tiles-256-lb0-87
  - How to make tiles
    - Split image by (tile_size, tile_size)
    - Sort by pixel value for each tile
    - Imgsize: ex. tile_size: 256, num_tile: 36
      - imsize: 256 x 6 = 1,536
  - Model: EfficientNet B0-B1 with CELoss
  - Tile mode augmentation (train & test)
  - TTA(tile mode, hvflip, transpose)
  - Convert label to bin
    - ex. 2 → [1, 1, 0, 0, 0], 4 → [1, 1, 1, 1, 0]
  - With any luck, it will exceed 0.87 at PublicLB
    - Many people couldn't reproduce this score...

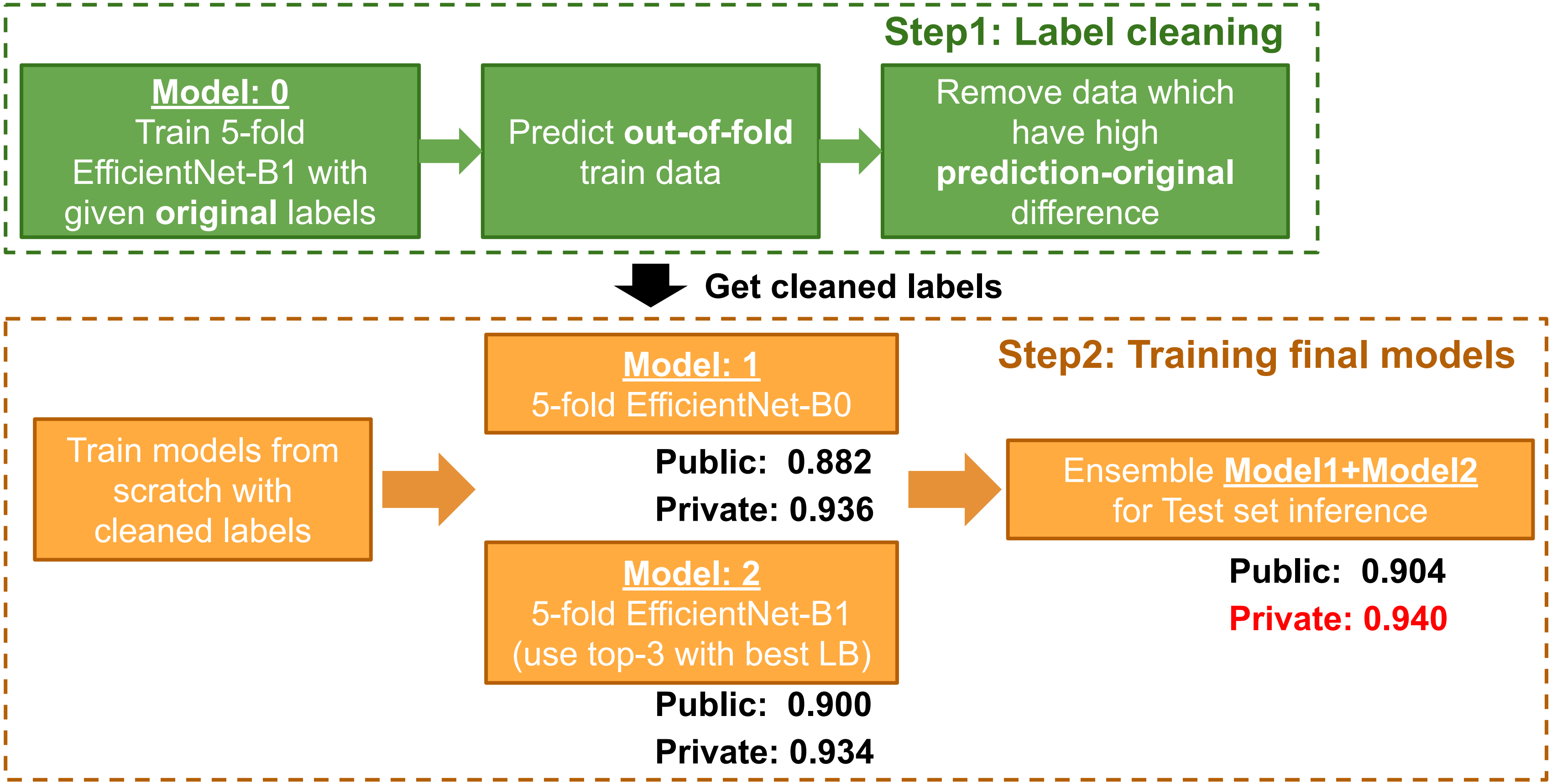


https://www.kaggle.com/iafoss/panda-concat-tile-pooling-starter-0-79-lb

3. Why Local CV ≠ Public LB? @ 🐼

- Many kagglers got high local CV, but low Public LB
- There could be several factors
    - Duplicate images
    - Label noise
        - If we got Radboud CV 1.0, Public LB may only return about 0.85 (on Radboud)
        - Almost solutions that did not have a big shake down were working on noise reduction
            - 2nd, 4th, 6th, 11th...
            - Or the smart ones quit participating😭
    - Small test dataset (public / private)
    - QWK

4. Our solution: Summary



4. Our solution: Each steps

- 1. Split kfold with image similarity
  - stratified k-fold (gleason-score), almost kernel split by isup-grade…
  - imgid (imghash similarity greater than 0.9) in same fold
    - Ofcourse, there were some wrong decisions
      - I didn't want to put the same image in a different fold any more than that
    - https://www.kaggle.com/yukkyo/imagehash-to-detect-duplicate-images-and-grouping
    - networkx was useful for grouping imgids that were determined to be the same
  - In retrospect, this is what we needed for our noise reduction
- 2. Training with original label (with noise)
- 3. Remove noise by prediction and original label gap(out of fold)
- 4. Re-train model without noise
- 5. Ensemble

4. Our solution: Label noise reduction

- Simple, yet effective label cleaning method
- Remove data based on the gap between the **hold-out prediction results** and the **given original 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. threshold: 1.5
    - Predicted ISUP = 4.1, Original ISUP = 4 **gap = 0.1** and data is **kept**
    - Predicted ISUP = 0.5, Original ISUP = 4 **gap = 3.5** data is **removed**

4. Our solution: Label noise reduction (Model 1)

- Remove data based on the prediction and the label gap and get cleaned labels.
  - Gap Threshold = 1.6
  - Remove ratio[%]: 5.614
  - Number of removed data
    - Total: 596
    - Radboud: 445
    - Karolinska: 151

```python
# 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)!

4. Our solution: Label noise reduction (Model 2)

- Remove data based on the prediction
  - Change gap threshold for each label for each data provider.
  - Threshold was set to remove **20%** of Radboud data.
  - **14.0 % of training data was removed.**
    - Number of removed data
      - Total: 1,488
      - Radboud: 1,153
      - Karolinska: 335

4. Our solution: Label noise reduction

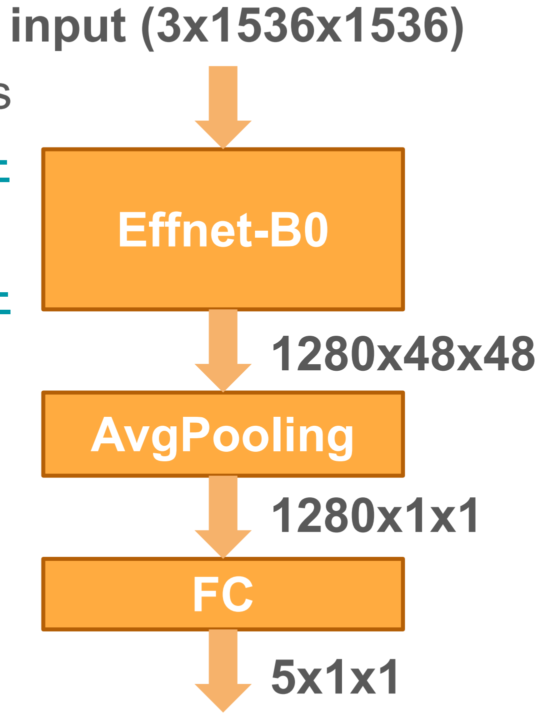
- Ablation study of noise reduction
  - Model 2 threshold (14.0 % of training data was removed)
  - Final model has slight modifications
  - Improved scores on both Public / Private

Model 2-like performance trained Before/After noise reduction

	Public	**Private**
Before noise reduction	**0.892**	**0.916**
After noise reduction	**0.901(+0.009)**	**0.932(+0.016)**

4. Our solution: 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
  - https://www.kaggle.com/iafoss/panda-16x128x128-tiles
- **Model 1: After denoise, arutema47**
  - Backbone:EfficientNet-B0, pooling: avg_pooling
  - Tile: 36 x 256 x 256
  - Augmentations (e.g. cutout, mixup) used for generalization
  - Cosine annealing schedule for 20 epochs
- Larger backbones introduced overfitting (e.g. resnext...)

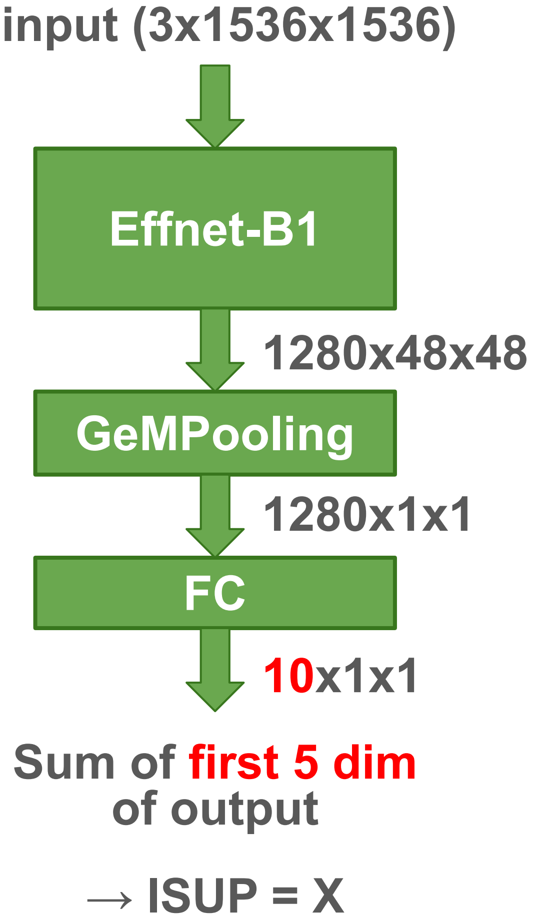


e.g. Sum([1,1,1,1,0])

→ ISUP = 4

4. Our solution: Model setup (2/2)

- **Model 0 and 2 (fam_taro)**
  - Model 0: Before denoise, used for denoising
  - Model 2: After denoise
- Some parts that differ from **Model 1(arutema47)**
  - Backbone:EfficientNet-B1, pooling: GeM pooling
  - Tile: 64 x 192 x 192
  - Cosine annealing schedule for 30 epochs
- Predict ISUP + first gleason score during training
  - e.g. 3+4 → 1st gleason score is 3
  - 10 dimension output
    - e.g. ISUP 3, 1st gleason 4 → [1,1,1,0,0,1,1,1,1,0]
  - Predicting first gleason score enables faster training and some improvements in LB.
  - Note that **only predicted ISUP** is used for test inference.



4. Our solution: Model setup (2/2)

- **Configs**

model 0

```yaml
train:
    - name: Transpose
    - name: HorizontalFlip
    - name: VerticalFlip
    - name: RandomRotate90
    - name: ShiftScaleRotate
      params:
          rotate_limit: 10
          shift_limit: 0.05
          scale_limit: 0.05
    - name: OneOf
      member:
          - name: ElasticTransform
            params:
                alpha: 120
                sigma: 6
                alpha_affine: 3.6
          - name: GridDistortion
          - name: OpticalDistortion
            params:
                distort_limit: 0.1
                shift_limit: 0.1
```

model 2

```yaml
train:
    - name: Transpose
    - name: HorizontalFlip
    - name: VerticalFlip
    - name: ShiftScaleRotate
      params:
          rotate_limit: 10
    - name: RandomBrightnessContrast
      params:
          brightness_limit: 0.2
          contrast_limit: 0.2
    - name: Cutout
      params:
          num_holes: 36
          max_h_size: 128
          max_w_size: 128
          fill_value: 0
```

Common config

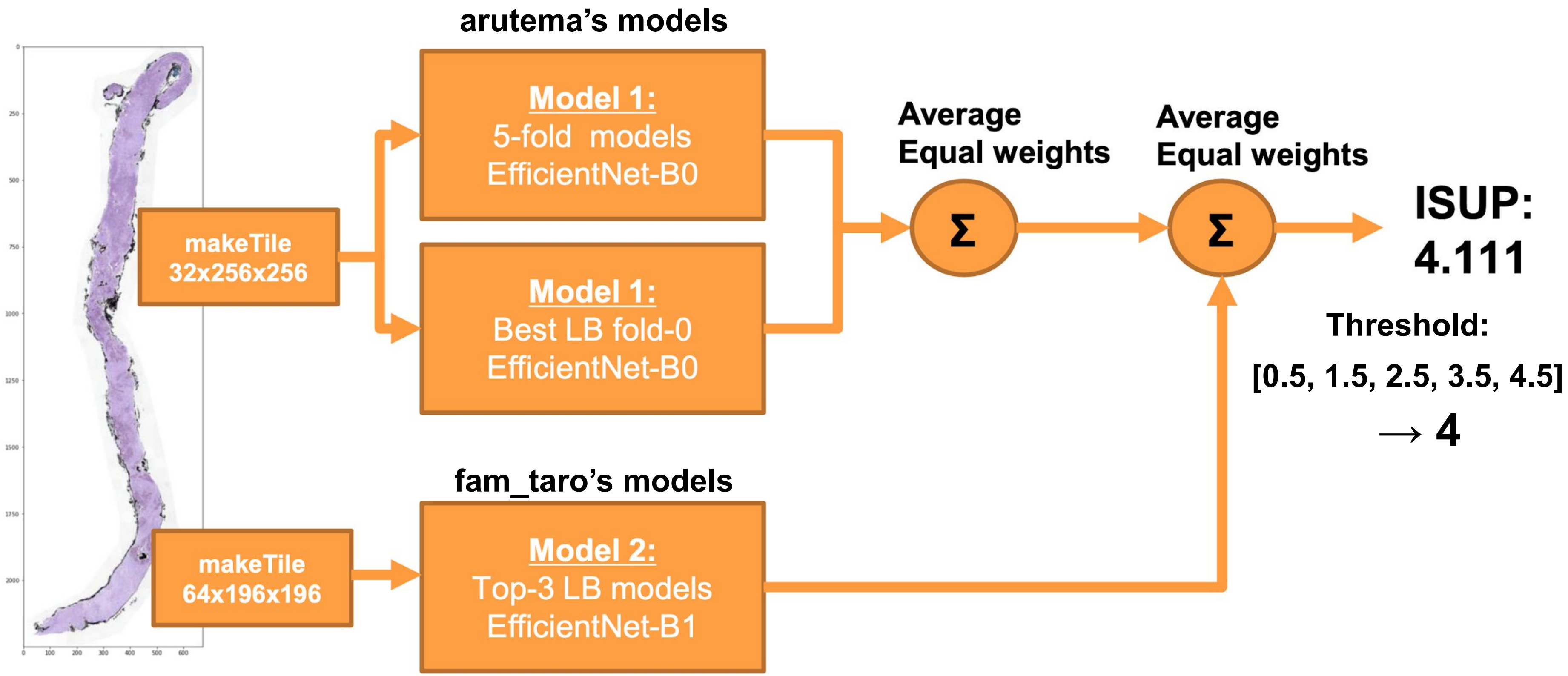
```yaml
General:
    fp16: True
    amp_level: O1
    multi_gpu_mode: ddp
    epoch: &epoch 30
    grad_acc: 2
    frozen_bn: False

Data:
    dataloader:
        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
```

4. Our solution: Inference pipeline



4. Our solution: Why did we win?

- Assumption 1: Private dataset contain more easy data than Public
  - We could get good score because our model is strong against easy data (but weak against difficult data)
    - Cons: Our denoise removes difficult data as well

- Assumption 2: Splitted kfold with imghash (considering duplicates)
  - We've placed duplicate images in the same fold by imghash.
    - We could make the LocalCV and **the noise reduction** more stable.
    - 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.

5. Conclusion

- Our solution point
  - Split kfold with imghash
  - Remove noise by gap between pred and original label
- Impressions
  - It's important to check official documents
  - Is that label correct?
  - Trust CV < Trust LB < Trust Yourself (@ 🐼)
  - Annotator is may be important point for medical image task
    - It is efficient that check how well a Train annotator can answer a Test correctly

Appendix: Not work for me

- Some try before denoise
  - Mixup, CutMix
  - Other tile method
    - NMS based, K-means based, etc…
  - CycleGAN augmentation(Karolinska ↔ Radboud)
  - Segmentation model with classification head
    - My implementation needs FP32...🤔
  - Other loss
    - Class balanced loss
    - Low weight first-gleason score(0.5)
- CleanLab (Confident-Learning)
  - Used for denoising
  - https://github.com/cgnorthcutt/cleanlab
  - Now it is only for classification label…(2020.09.24)