Solution for RSNA Intracranial Hemorrhage Detection

Keep Digging Gold (5th place)
November 24th, 2019

Contents

1. Overview 1
2. Summary 2
   - Image Preprocessing 2
   - Data Preprocessing 2
   - Modelling 3
   - Post-processing 4
   - Stacking 5
3. Important and Interesting Findings 5
4. Best single model: EfficientNet-B5 6
5. Hardware 6
6. Code 6

1. Overview

Competition name: RSNA Intracranial Hemorrhage Detection
Team name: Keep Digging Gold
Private Leaderboard score: 0.04561
Private Leaderboard place: 5th

Team members:
1/ Nguyen Xuan Bac (kaggle's nickname: Bac Nguyen)
Location: 77 Yongbong-ro, Yongbong-dong, Buk-gu, Gwangju, South Korea.
Email: ngxbac.dt@gmail.com
2/ Tien-Dung Le (kaggle's nickname: tarobxl)
Location: Brussels, Belgium
Email: tiendzung.le@gmail.com
3/ Bui Duc Toan (kaggle's nickname: msl23518)
Location: Vietnam
Email: mslskku@gmail.com
4/ Nguyen Tang Tri Duc (kaggle's nickname: KeepLearning)
Location: 54 Quach Dinh Bao Str., Ho Chi Minh City, Vietnam
Email: triducnguyentang@gmail.com
5/ Anjum Sayed (kaggle's nickname: anjum48)
Location: Kuwait
Email: anjum.sayed48@gmail.com

2. Summary

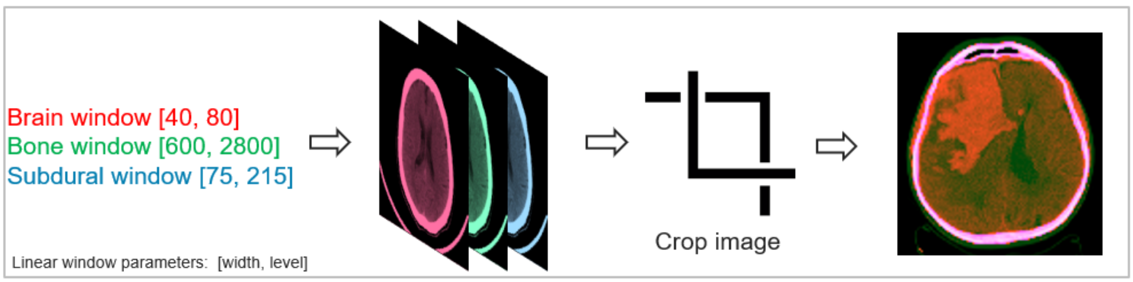
- Image Preprocessing

We have 2 types of preprocessing data.

1) Imaging with multiple windows: We use three windows to construct RGB image. Each channel is corresponded to a window.

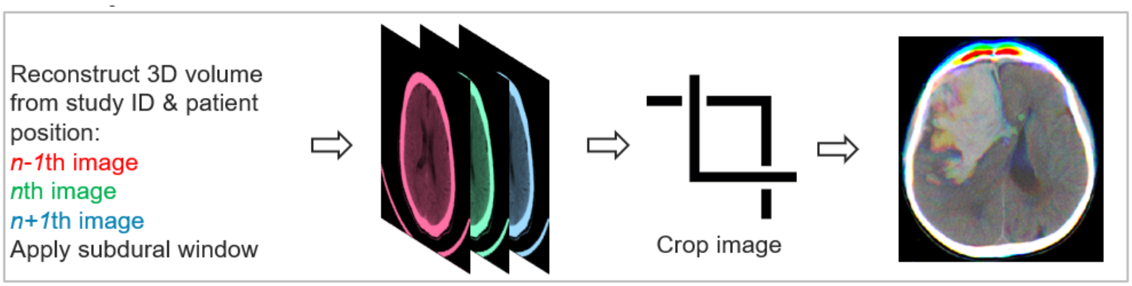
```
'brain': [40, 80],
'bone': [600, 2800],
'subdual': [75, 215]
```

1*) Imaging with multiple windows then crop.
Same as (1), we crop and keep the only informative part.



2) Imaging with spatially adjacent.
We use only one window [40, 80] for preprocessing. To construct RGB images, we use metadata to know the spatially adjacent. Let say to construct RGB of slice St, we take:
R = St-1, G = St, B = St+1.

Finally, we crop and keep only informative parts as same as (1*). Refer this kernel for more detail: https://www.kaggle.com/anjum48/preprocessing-adjacent-images-and-cropping



- Data Preprocessing

1) First approach

   5 folds grouped by patients

2) Second approach

Similar to the 1) approach except

a) We remove the overlapped patients between train and test. This part may be the reason for the shakeup since we estimate that the shakeup score is in a range of 0.001 - 0.002.

b) In each fold, we do random sampling such that the number of positive patients is balanced to the number of negative patients. This step helps to have the correlation between CV and LB, and stable as well.

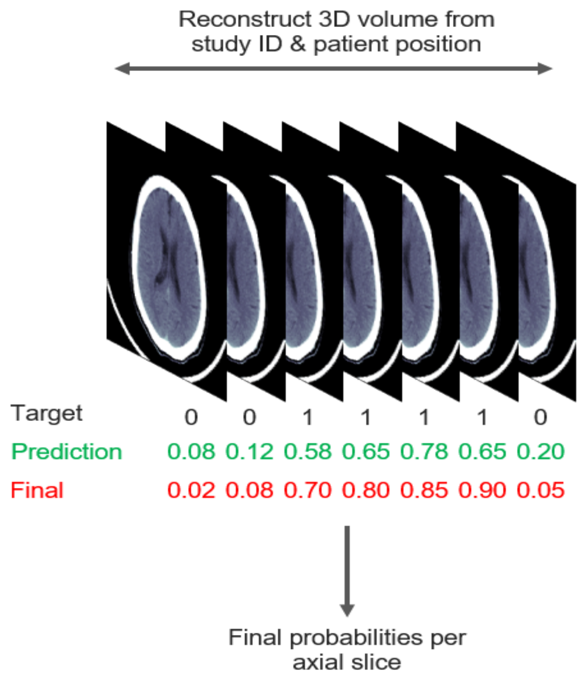
- Modelling

The models and their performance on stage 2:

Preprocessing Method	Network Architecture	Image Resolution	Score Before Post processing	Score After Post processing
1*	ResNet-18	512x512	0.06	0.054
1*	ResNet-34	512x512	0.058	0.052
1*	ResNet-50	512x512	0.058	0.052
1*	Inception-V3 & Deep Supervision	512x512	0.063	0.053
1	DenseNet-169	512x512	0.055	0.049
1	EfficientNet-B3	300x300	0.055	0.050
2	ResNet-50	512x512	0.054	0.051
2	EfficientNet-B0	224x224	0.054	0.051
2	EfficientNet-B5	456x456	0.048	0.048

- Post-processing

It is a simple solution to leverage a sequential model in order to exploit the 3D information.



For each class in 6 groups ['any', 'epidural', 'intraparenchymal', 'intraventricular', 'subarachnoid', 'subdural'], form a binary classification problem.

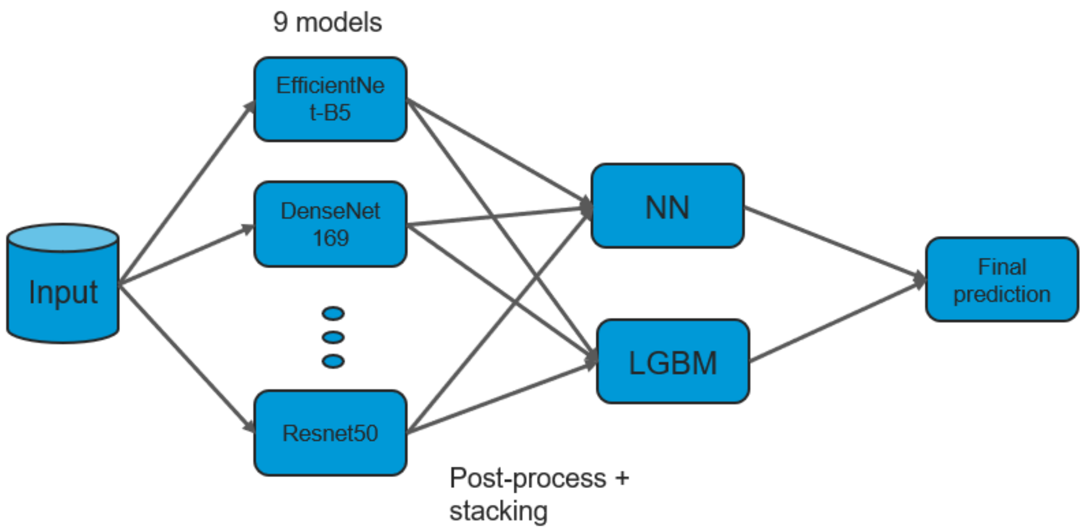
Sort images in each study_instance_uid by image_position_patient_2 then for each image, create the following features

- The original prediction p0
- The predictions of the previous image p_prev and the next image p_next
- The statistical features for all images before and all the images after the image in the same studyID: number of images, mean/std/skew of the original predictions.

Binary classifiers built using H2O H2OGradientBoostingEstimator model.

- Stacking

1/ For CNN model, use a window size (NUMMODELS=9, 1) to learn the correlation between 9 models, and a window size (1,NUMCLASSES=6) to learn the correlation between 6 classes.



2/ Mix-up augmentation is applied.

3/ For lgbm, 6 separate models are built for each class.

More details are shown at https://www.kaggle.com/mathormad/5th-place-solution-stacking-pipeline.

3. Important and Interesting Findings

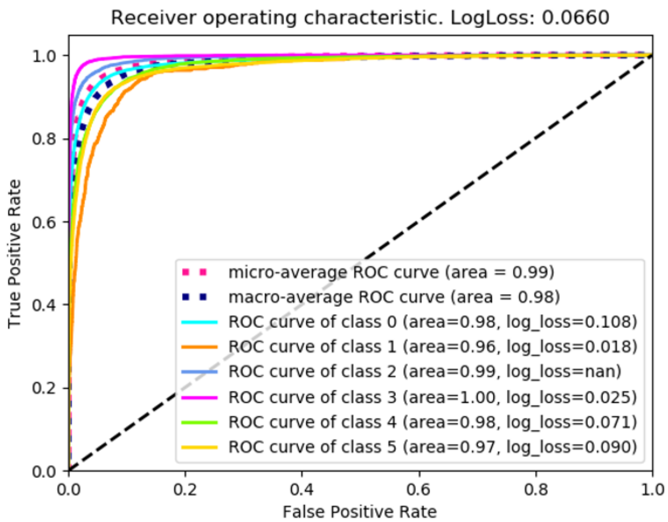
1. Overfitting: This issue is limited by these techniques bellow
    a. Split images by patients to avoid
    b. Early stopping was essential. Most models only needed approx. 5-7 epochs to reach the best validation score

2. Sequence learning or 3D-information modelling:

    Treating this as a 3D problem gives the best results. Pre-processing method 1) benefitted most from the post-processing as the pre-processing method 2) already captured some of the spatial detail but could only utilize one CT window

4. Best single model: EfficientNet-B5

Our best single model is based on EfficientNet-B5 model. Results from validation of fold 1 is shown as bellow.



Results from validation of fold 1 from the EfficientNet-B5 model

Classes:

0. Any
1. Epidural
2. Intraparenchymal
3. Intraventricular
4. Subarachnoid
5. Subdural

Similar study achieved average ROC-AUC scores in the region of 0.93. See figure 3d of “Precise diagnosis of intracranial hemorrhage and subtypes using a three-dimensional joint convolutional and recurrent neural network, Hai et al. April 2019”.

5. Hardware

DGX Workstation, 256G RAM, 4 x V100 (16G) GPU from Bac Nguyen
DGX Workstation, 256G RAM, 4 x V100 (16G) GPU from Taroxl
2x (Intel Core i7, 64G RAM, 4 x 1070 Ti (8G) GPU) from KeepLearning
AMD Threadripper (16 cores), 128GB RAM, 2 x RTX 2080Ti from datasaurus
2 x Titan Xp from Toan

6. Code

[1] Bac Nguyen’s source code
https://github.com/ngxbac/Kaggle-RSNA
[2] Nguyen Tai Tri Duc 's pipeline to train InceptionV3 + Deepsupervision
https://github.com/triducnguyentang/RSNA
[3] Anjum 's pipeline
https://github.com/Anjum48/rsna-ich
[4] Tien Dung LE ’s post-processing
https://github.com/tiendzung-le/Kaggle-RSNA-5th-place-Solution/