Model Summary

A1. Background on you/your team

Competition Name: Prostate cANcer graDe Assessment (PANDA) Challenge
Team Name:Save The Prostate
Private Leaderboard Score: 0.93768
Private Leaderboard Place: 2

Name: DrHB (Habib S. T. Bukhari)
Location:USA
Email:bibahaba@gmail.com

Name: CatEek
Location:Russia
Email:zubarev.ia@gmail.com

Name: R GUO (Rui Guo)
Location:USA
Email: ruiguo97@126.com

Name: Kuzmin Rinat
Location:Russia
Email:trytolose@gmail.com

Name: Xie29(HSIEH, CHIA-LUN)
Location: Taiwan
Email:r04941087@ntu.edu.tw

A2. Background on you/your team

What is your academic/professional background?

A: Xie29 : I majored in electric-optical engineering and image processing engineer is my daytime job.

A: DrHB: I have PhD in structural Biology, I use to work in the wet lab, but 3 months ago I moved to scientific computing

A: R Guo: I am currently a graduate student in Electrical and Computer Engineering in University of Michigan.

A: CatEek: I have MSc in mechanical engineering, defense related. Spent last several years in finance industry. In the process of shifting careers now.

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

A: Xie29 : The experiences in previous kaggle competitions help me have better insight on data and model/loss function selection. Also my daytime job, image processing engineer helped me deal with image data better

A: DrHB. Only experience from the previous Kaggle Competition.

A: R Guo: Yes, I took some ML and DL courses and I entered several competitions before in Kaggle and got much experience from them.

A: CatEek. Experience from other Kaggle Competitions and a few ML/DL courses

What made you decide to enter this competition?

A: Xie29 : This is my first medical competition, and since there are so many deep learning jobs are related to medical field. So I decided to join this competition to have practical experiences on medical data.

A: DrHB: The problem was very challenging.

A:R Guo: I am interested in DL applications in medical field and want to gain some experiences in this field.

A: CatEek: I'm very interested in AI in medicine/biology, so I'm trying to participate in all related competitions. This particular one also caught my attention thanks to a very challenging problem statement.

How much time did you spend on the competition?
A: Xie29 : Almost three months.

A: R Guo : I entered this competition in May and spent 3-4 hours on this competition every day.

A: DrHB: I have entered the competition since the beginning and spend almost 3-4 hours every day.

A: CatEek. I started working on this competition about a month from the start and spent several hours every day.

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

A: Xie29 : At the early stage, I noticed that ensemble my own models can't bring too much improvement to the result. And since the data is quite noisy, I thought increasing the diversity of ensemble models might be helpful in this competition.

A: R Guo : We have similar single models scores and we want to share ideas and get better scores by ensemble of different methods

A: DrHB: We had similar scores, and we decided to team up.

A: CatEek: We teamed up with DrHb and Trytolose pretty early, others joined later on because we thought it would be a good opportunity to join resources and share ideas.

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

A: Xie29 : I was using TPU on Kaggle/Colab as my training machine. In my part of the solution, I used different loss functions on different center data, due to Radboud data being more noisy than Karolinska data. And I extracted the tiles from slides and glued them into a big image as my training data. Here is my solution summary: https://www.kaggle.com/c/prostate-cancer-grade-assessment/discussion/169303. Basically we all have different models for increasing the diversity of ensemble models. So we all trained our individual models by our own resources. And during the time, we will share the discoveries/findings to help each team member know each of our models more.

A: R Guo: We solve this problem through different ways. We together made our final solutions.

A: DrHB: We had a lot of discussion and everyone was pursuing and testing different ideas.

A: CatEek: Each of us had a different approach and we tried to make our solution as diverse as possible.

A3. Summary

We used Convolutional Neural Networks to solve this problem. Instead of using the entire whole slide image as input, we first crop the whole image into tiles and take tiles as input. During the competition, we developed several models to tackle this problem, using different input structures and different network design.

The tool(s) you used
Packages : Tensorflow/Keras/Numpy/Pandas/skimage/sklearn/Pytorch/Apex/OpenSlide/FasiAI

machine : TPU/ GPU (2080Ti, RTX Titan, 1080ti)

How long it takes to train your model (single model)
A: Xie29: 5~6 hours on TPU

A: DrHB: two stage model around 16 hours.

A:R Guo: models on level 1 need around 10-12 hours and models on half scale level 0 need 15 hours.

A: CatEek: model with 1 image per batch, around 16-18 hours in total

A4. Features Selection / Engineering

Did you make any important feature transformations?

A: Xie29 : I selected the image feature by the pixel value of patches/tiles. Since most areas of slides are blank, I extracted the patches/tiles with lower average pixel value and glued them into one big image as my image feature.

A: R Guo: I have a two stage model. In the first stage I choose tiles according to their pixel values. Darker tiles are preferred to lighter ones. After training the first stage on median resolution. I select high resolution tiles by attention calculated in median level. The attention tends to select tiles that are cancerous. I used separate tiles as input.

A: DrHB: I have a two stage model. In the first stage I used 49 tiles (sampled based on tissue presence) and second stage 81 tiles. I used separate tiles as input.

A: CatEek: I wanted to extract as much information from each slide as possible. I used 49 tiles (sampled based on the amount of tissue and randomly). I used separate tiles as input.

Did you find any interesting interactions between features?

Using both kinds of input (separate tiles/ glued tiles) can reach similar results.

Did you use external data? (if permitted)

Only ImageNet pretrained weights are used.

A5. Training Method(s)

What training methods did you use?
A: Xie29 :
One cycle cosine annealing learning rate scheduler with initial learning rate 5e-4 and minimum learning rate 5e-6. The best checkpoints will be saved separately depending on their improvement on individual validation loss of different data centers. And only use the best checkpoint of the Karolinska data center in inference time.
I was using noisy-student pretrained backbones and a simple regression head. And instead of using separated tiles for model training, I glued the tiles into a big image. And the tiles were collected from lower average pixel value patches from original slides. At first, I only used 128 tiles to glue into big images due to some memory usage concern. But after a while, I thought I could use more tiles for better feature extraction from original slides, so I changed the tiles number to 144 for gluing into a big image. And since the data from both centers are slightly different, especially colors. So I applied H&E color jitter in my data pipeline for data augmentation, since I hope the model can have better generalization ability. Apart from H&E color jitter, I also performed random contrast/brightness/saturation for data augmentation, and the reason is the same as I performed H&E color jitter. For transformation data augmentation, I performed shift, rotate, scale, h/v flip and transpose, these are some basic data augmentation which were used in every competition. And the purpose of these augmentation is for increasing more diversity for my training data and letting models have better generalization ability. For my model, I was using efficientnet series networks for my backbone, since I thought it has better balance on performance and training efficiency. After the backbone I used GeM pooling since it is similar to Global average pooling and Global Max pooling but it has weights in the pooling process, so should be better than simple GAP/GMP. After GeM pooling, I simply add a fully-connected layer with 128 units and dropout with 0.3 dropout rate. And the last layer is a fully-connected layer with 1 unit for regression. And the most important part of my solution should be loss function, I used different loss functions for data from different centers. Since data from Radboud is more noisy, so I used huber loss with 1 delta for the instances from Radboud. Also the data from Karolinska is relatively clean, so I want models to learn data from Karolinska data as much as possible, so I used MSE loss for instances from

Karolinska. Basically preparing the mask of each data center can easily do this. (If you skip the instances with mae loss bigger than 4 on Radboud instances can have better private board score.)

A: R Guo:
I used ImageNet pretrained backbones and designed customized heads. Instead of gluing tiles, I send separate tiles to CNN and have an attention and max pooling layer after backbone, which intends to gather information from tiles in the same slide. My network had three heads, one for whole slide isup_grade regression, one for whole slide isup_grade classification and last one for tile level binary classification (benign or cancerous). The structure is shown in Figure 1.

I trained my models for two stages, the first stage is trained on median level, with simple color based tile selection. During training, I do tile selection online, and add random offset to the grid. I used 48 192x192 tiles from median level. I used smoothed labels for both classification and regression to reduce label noise. To generate these labels, I first train a 5-fold model with ground truth, take its out-of-fold prediction and calculate a weighted sum with ground truth(0.3 * oof prediction + 0.7 * ground truth). I used MSE for regression, cross entropy for classification. I used seresnext50 as my backbone.

In the second stage, I used attention calculated on the median level to select high resolution tiles, in order to reduce tile numbers while keeping decent performance. I saved three groups of cached tiles with different tiling offset to disk to reduce overfitting. I randomly selected 32 tiles according to their attention during training. The other setting is very similar with median level.

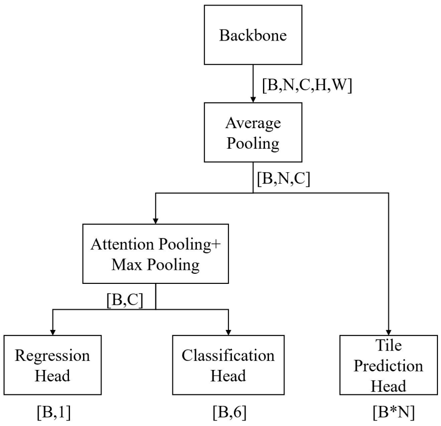

Figure 1. Model Structure

A: DrHB.

I trained in two phases. In the First phase he was trained with 49 tiles and later fine tuned with 81 tiles. Both phases were using standard one cycle.
I decided to use a very simple model resnet 34 but throughout competitions ended up doing few modifications:
1) Making square features:
After resenet encoder we reshape features to look like a square in a following way: x = x.view(x.shape[0], x.shape[1], x.shape[2]//int(np.sqrt(N)), -1) . Here N represents Number of Tiles. After this we pass all the features to SqueezeExcite Block
2) SqueezeExcite block
After reshaping features, we added 1 SE block to enable the network to learn features for individual slides based on tiles.
4) Final Head
I used two heads. One head was for classification, the second was for regression. I noticed that training with two losses makes training much smoother (with sigmoid trick below) and yields higher local CV (0.88 -> 0.90). In the final prediction, I use output only for the regression head.
One small modification that I did before calculating loss is that the regression head used sigmoid to scale outputs between (-1. 6.). This enables much smoother training without bumps and faster convergence.

A: CatEek
I decided to use a model with group normalization, which allowed me to use batch size 1. Each WSI slide was randomly cropped to 95% of the original size and randomly rotated with a 5 degrees limit, thus ensuring tiles would be slightly

different for each epoch. Those parameter were very carefully selected in order not to introduce too much noise into the model. Medium resolution was used. I extracted 49 tiles from each slide. First ⅓ of the tiles were extracted from an array sorted by amount of tissue present in each tile. The rest were taken from a randomly shuffled array. Extensive augmentation were applied to each tile, including cutouts, I used ResNeXt50 model pretrained on ImageNet, with group normalization layers instead of conventional batch normalization and weight standardization for each convolutional layer. Model input was of shape (B, N, C, H, W), where N is the number of tiles. To allow input of that shape, feature tensor was reshaped by stacking the tile dimension with batch. Backbone output features were reshaped again by stacking tiles into H and W dimensions, producing a conventional sized tensor (B, C, H, W). SqueezeExcite block was added right before the regression head. Reduce on plateau learning rate scheduler with patience 5 was used.

Did you ensemble the models?

Yes

If you did ensemble, how did you weight the different models?

By checking CV and LB.

A6. Interesting findings

What was the most important trick you used?

A: Xie29: I used different loss functions on instances from different data centers. Since the data from the radboud center is more noisy than the karolinska center. So I choose huber loss with delta 1 as loss function for radboud instances and MSE on karolinska instances.

A: R Guo: I used smoothed labels generated by oof prediction and groundtruth. This reduces the effect of label noise in training data. On the other hand, in order to reduce GPU memory requirements, I used gradient checkpoints and apex mixed precision training. By using these methods, I made my batch size 3 times bigger and improved nearly 0.01 in CV.

A: DrHB. Modification of architecture, using double losses. And ensembling with Convolution networks trained using different methods.

A: CatEek. Group normalization with weight standardization allowed training to be stable with very small batch size. Using an optimized ETL pipeline for large images. With a slightly more powerful CPU there will be no necessity to save preprocessed slides.

What do you think set you apart from others in the competition?

We all removed slides with pen marks and some suspicious slides with inconsistent labels and masks, which makes train images closer to test images

A: Xie29 : I will assume most of the people in this competition we're using MSE on all data, which might bring high overall cv but low LB. Since the model already overfitted on noisy labels of radboud instances.

A: R Guo: We tried to deal with noisy labels.

A: DrHB, we had a very diverse idea. Double Losses, Different Architectures and Different Features representations.

A: CatEek, We realized that noisy labels are our main concern quite early. As a team, we tested a lot of different approaches to address this issue and found a very stable setup, which allowed us to choose the right ensemble for final submission.

A7. Simple Features and Methods

Is there a subset of features that would get 90-95% of your final performance?

A: Xie29 : Yeah, I was using an image composed of 144 tiles with 128x128 tile size as training / inference data for competition. But I also tried 121 tiles with 128x128 tile size at the earlier stage of competition, it also can get pretty decent results with the same model and training strategy.

A: R Guo: The tile number in high resolution model can be further reduced. I once trained with 24 tiles and got nearly the same result. I also checked the attention values, it gives decent weights to fewer than 10 tiles. Reducing tile numbers to 12 or 16 can give very close performance to my current model.

A: DrHB: Yes, just using 49 tiles with two loses and resnet modifications that I mentioned above.

What would the simplified model score?

A: XIe 29 : With efficientnetb0 and lower tile amount, it should be able to get 0.915~092 private board score quite easily.

A: R Guo: With lower tile numbers, It should score around 0.92 in private.

A: DrHB: I was using a simple model already resnet34. It scores around 0.911~0.92 in private

A8. Model Execution Time

How long does it take to train your model? (single model)

A: Xie29: 5~6 hours on 8 cores TPU

A: R Guo: 15 hours for efficientnet-b0 on RTX Titan

A: DrHB: 16 hours on RTX Titan

A: CatEek: Approx. 16 hours on 4x1080ti

How long does it take to generate predictions using your model?

On kaggle’s P100 GPU and 2 cores CPU, it takes 3.5 hours to run all models, but this time is limited by CPU.

How long does it take to train the simplified model (referenced in section A6)?

A: Xie29: Around 3 hours on 8 cores TPU

A: R Guo: Around 10 hours on RTX Titan

A: DrHB: Around 16 hours

How long does it take to generate predictions from the simplified model?

Less than 1 hour. Can be faster with more CPUs.

A9. References

[1] Wouter Bulten, Hans Pinckaers, Hester van Boven, Robert Vink, Thomas de Bel, Bram van Ginneken, Jeroen van der Laak, Christina Hulsbergen-van de Kaa, and Geert Litjens. Automated gleason grading of prostate biopsies using deep learning. *arXiv preprint arXiv:1907.07980,* 2019.1

[2] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning, 2018.

[3] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight Standardization, 2019.

[4] Peter Str ¨om, Kimmo Kartasalo, Henrik Olsson, Leslie Solorzano, Brett De-lahunt, Daniel M Berney, David G Bostwick, Andrew J Evans, David JGrignon, Peter A Humphrey, et al. Pathologist-level grading of prostate biopsies with artificial intelligence. *arXiv preprint arXiv:1907.01368,* 2019.

[5] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946,* 2019.

[6] David Tellez, Maschenka Balkenhol, Nico Karssemeijer, Geert Litjens,Jeroen van der Laak, and Francesco Ciompi. H and E stain augmentation improves generalization of convolutional networks for histopathological mitosis detection. In John E. Tomaszewski and Metin N. Gurcan, editors,*Medical Imaging 2018: Digital Pathology,* volume 10581, pages 264 – 270. International Society for Optics and Photonics, SPIE, 2018.