Deep learning in medical imaging:
Prostate cancer grade assessment challenge

Mehdi Zemni & Hamdi Bel Hadj Hassine
mehdi.zemni@student-cs.fr & hamdi.belhadjhassine@ensae.fr

Code: https://github.com/zmehdiz97/Kaggle-DLMI

April, 2022

Abstract

In recent years, deep learning technology has been used for analysing medical images in various fields. The use of artificial intelligence (AI) in diagnostic medical imaging is undergoing extensive evaluation. AI has shown impressive accuracy and sensitivity in the identification of imaging abnormalities and promises to enhance tissue-based detection and characterisation.

Prostate cancer is one of the most common types of cancer for men. Usually prostate cancer grows slowly and is initially confined to the prostate gland, where it may not cause serious harm.

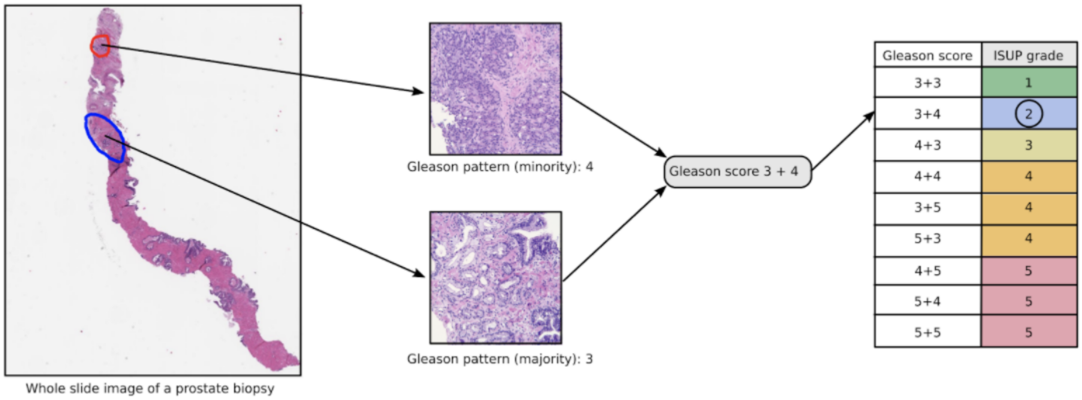
The key to decreasing mortality is developing more precise diagnostics. For this reason, several researches have focused on the use of deep learning to detect prostate cancer.

Diagnosis of PCa is based on the grading of prostate tissue biopsies. These tissue samples are examined by a pathologist and scored according to the Gleason grading system.

In this challenge, we will develop models for detecting PCa on images of prostate tissue samples, and estimate severity of the disease using the most extensive multi-center dataset on Gleason grading yet available.

1 Introduction

The grading process consists of finding and classifying cancer tissue into so-called Gleason patterns (3, 4, or 5) based on the architectural growth patterns of the tumor (fig. 1). After the biopsy is assigned a Gleason score, it is converted into an ISUP grade on a 1-5 scale. The Gleason grading system is the most important prognostic marker for PCa, and the ISUP grade has a crucial role when deciding how a patient should be treated. There is both a risk of missing cancers and a large risk of overgrading resulting in unnecessary treatment. However, the system suffers from significant inter-observer variability between pathologists, limiting its usefulness for individual patients. This variability in ratings could lead to unnecessary treatment, or worse, missing a severe diagnosis.


Gleason pattern (minority): 4
Gleason score 3 + 4
Gleason pattern (majority): 3
Whole slide image of a prostate biopsy

Gleason score	ISUP grade
3+3	1
3+4	2
4+3	3
4+4	4
3+5	4
5+3	4
4+5	5
5+4	5
5+5	5

Figure 1: Gleason grading process.

The goal of this challenge is to predict the ISUP Grade using only Histopathology images. For that, we will need to deal with the process of Whole Slide Images as huge gigapixel images and deal with the limited number of patients provided in the train set.

2 Dataset

The dataset contains 340 whole slide images for training and 86 images for evaluation. Whole Slide Images are a particular format of images : it has a pyramidal structure with several levels called levels of magnification. Each level has its own resolution that represents the level of zoom we go through. We can't use directly the first and second level of resolution without preprocessing because it won't fit in the memory and we need to come up with a proper pipeline to process them.

We are also provided segmentation masks showing which parts of the image led to the ISUP grade (fig. 2). However not all training images have label masks, and mask labels are not always accurate. It should also be noted that the labeling of masks depends on the data provider and test images do not have masks.

For each training image we know its data provider (Radboud and Karolinska), its Gleason score and its ISUP grade. For each test image, we only know its data provider.

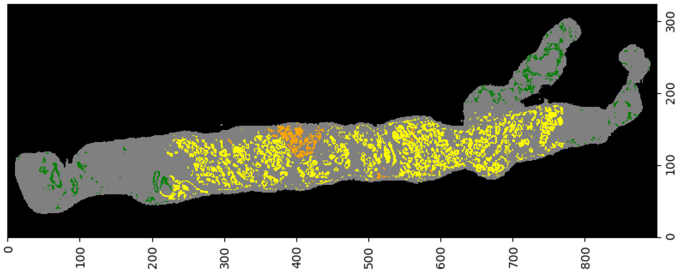


Figure 2: Example of a mask.

3 Proposed method

3.1 Multiple Instance Learning

Multiple Instance Learning (MIL) is a form of weakly supervised learning where training instances are arranged in sets, called bags, and a label is provided for the entire bag, opposedly to the instances themselves. In the standard MIL assumption, negative bags are said to contain only negative instances, while positive bags contain at least one positive instance (Fig. 3). Our problem can be viewed as a Multiple Instance Learning problem where each bag represents a sample (whole slide image) containing tiles of a fixed size extracted from the image. This will simplify a lot the task since we will retain only tiles that contain tissue.

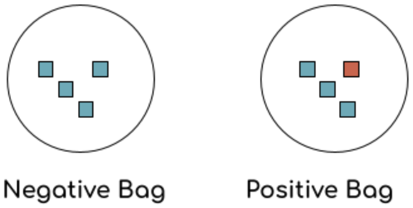


Figure 3: Multiple Instance Learning bags.

3.2 Extracting patches

Whole slide images consist of large empty areas leading to inefficient use of GPU memory and GPU time.

Instead of passing an entire image as an input to a neural network, we select N tiles from each image based on the number of tissue pixels (we retain top N tiles that have the highest value of sum of pixels and discard tiles with large white space). We extract patches of all images and save them beforehand. We use the intermediate resolution of the whole slide images. We didn't manage to use the full resolution of the whole slide images in this competition but in the next sections we propose ideas that may allow using the maximum resolution by exploiting a model trained on a medium resolution. We extract two types of patches:

- 36 patches of size 256x256.
- 128 patches of size 128x128

For a more advanced tiling method we can use masks to select more relevant patches but this gave us worse results on the test set since tiling on the test set was not done with masks.

3.3 Bag level classification

3.3.1 Max/Avg Pooling

With a bag-level label, we can have a latent space containing the probability of each segment (using a sequence-based input). By applying a pooling operator (max/average pooling), there's just a single score associated with a bag (Fig. 4).

So tiles are passed independently through the convolutional part. The outputs of the convolutional part is concatenated in a large single map for each image preceding pooling and FC head. In practice we use both a maxpooling and an average pooling layer and we concatenate their outputs.

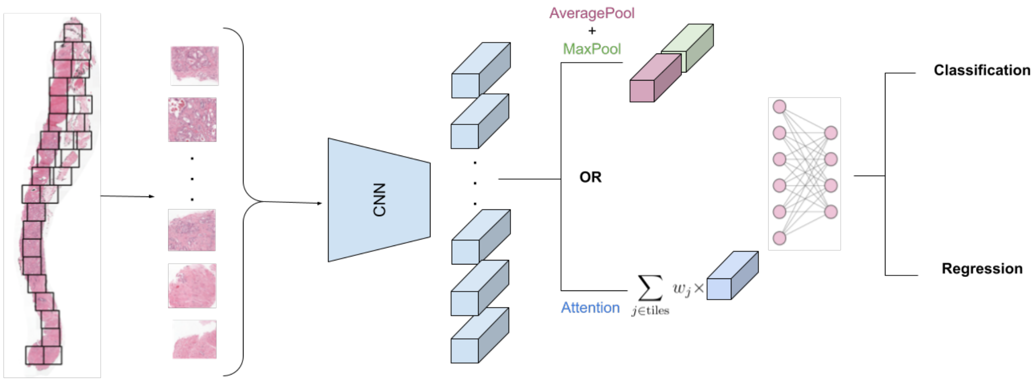


Figure 4: Network architecture.

3.3.2 Attention Pooling

As an alternative to the Pooling layer we can use an attention mechanism. The attention layer will determine how informative a tile is for classifying a certain sample by assigning weights to each one of the tiles.

This method is of great interest compared to the pooling head. First, it is learnable layer that will be optimized during training. The idea behind this attention mechanism is that we wanted at first to train a segmentation neural network to predict masks but this was not straightforward because we were manipulating whole slide images and mask labels differed from one data provider to another. Our intuition about this attention layer is that it will assign weights to patches in a similar way to masks. Fig. 5 shows the distribution of weight magnitude in each patch. To further improve tile extraction and selection, we can train a model on intermediate resolution tiles and use this model to extract high resolution tiles that have a high attention weight. By doing this we can remove unnecessary tiles and therefore decrease memory usage.

This idea was not tested but could be a solution to work with high resolution.

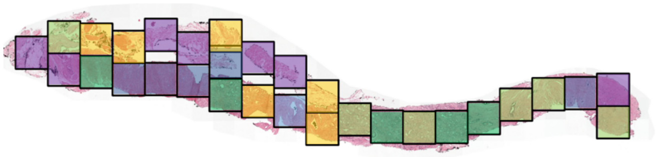


Figure 5: Weights assigned by attention layer to patches.

3.3.3 Classification layer and loss function

After the pooling layer, features will be flattened and passed through a fully connected head. During several days, we considered this challenge as a classification task only and we used the standard cross-entropy loss. We didn't manage to get good results using the cross-entropy loss but we found out that there is problem with our evaluation metrics. We were evaluating Accuracy and F1 score but scores of our submission were much higher than the accuracy and F1 score that we found on the validation set. Finally we realized that the evaluation metric is the Quadratic Cohen Kappa score which measures the agreement between two ratings and not the AUC (Area Under Curve) as stated in the challenge. In fact, It is not possible to measure the AUC since we submit class predictions and not probabilities. This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters).

ISUP grades are not just classes but they represent ordered labels. The advantage of a regression loss over the cross-entropy loss is that it penalizes more misclassifications between very low and very high ISUP grades while cross-entropy handles all misclassifications in the same way. And since we are evaluating models with the quadratic kappa score it is better to use a regression loss.

We observed that training a model with a regression loss (eg. MSE loss) gave better results. In fact, we use a rounder that is optimized during training that will round the output of the model to get integer values.

We tried also to train a model with two heads, a regression head and a classification head, that were optimized using respectively MSE and cross-entropy loss.

Another interesting approach is ordinal regression Niu et al. [2016] which is useful in grading problems with a binning loss. E.g.:

- label = [0,0,0,0,0] means ISUP grade = 0
- label = [1,1,1,0,0] means ISUP grade = 3
- label = [1,1,1,1,1] means ISUP grade = 5

3.4 Backbone architecture

We tried different CNN architectures namely Efficientnet-b0 Tan and Le [2019], Resnet34 and Resnet50 He et al. [2015]. Deeper architectures will require more memory which would limit the size of the batch and therefore degrade the performance.

3.5 Data augmentation and overfitting

One of the main issues that we encountered is the overfitting. to get the best test results we use early stopping to prevent the model from overfitting the training set. We also add a dropout layer with 0.5 probability. We also add several data augmentation transformations:

- HorizontalFlip (p=0.5)
- VerticalFlip(p=0.5)
- RandomRotate90(p=0.5)

- ShiftScaleRotate(p=0.15)
- one of (HueSaturationValue, CLAHE, RandomBrightnessContrast) p=0.15

Each tile will be transformed independently. On top of that at each iteration, the batch is composed of randomly sampled tiles (e.g. for 256x256 tiles we randomly select 24 tiles out of 36)

Finally, we add weight decay to our Adam optimizer.

3.6 Ensemble learning and TTA

To obtain better predictive performance, we use ensemble learning. The Ensemble learning is based on a majority voting ensemble of 4 models with 8 TTA (test time augmentation). TTA consists in using 7 additional augmented inputs (using flip and transpose transformations) for each each sample of the test set and then averaging the predictions.

4 Experiments and ablation studies

In the table below we sum up the experiments that we conducted using techniques presented in the previous sections.

	Parameters						Kappa score	
	Architecture	Pool head	Loss	# tiles	tile size	Aug/drop	val set	LB.
Baseline	resnet50	AvgPool	CE	64	128	✗	0.63	0.60
model 1	resnet50	AvgPool	CE	64	128	✓	0.68	-
model 2	resnet50	Avg/MaxPool	CE	16	256	✓	0.75	-
model 3	resnet34	Avg/MaxPool	CE	16	256	✓	0.77	-
model 4	effnet-b0	Avg/MaxPool	CE	16	256	✓	0.79	-
model 5	effnet-b0	Avg/MaxPool	MSE	16	256	✓	0.80	0.80
model 6	effnet-b0	Avg/MaxPool	MSE	24	256	✓	0.84	0.83
model 7	effnet-b0	Attention	MSE	24	256	✓	0.86	0.85
model 8	effnet-b0	Attention	binning	24	256	✓	0.83	-
model 9	effnet-b0	Attention	MSE/binning	24	256	✓	**0.88**	**0.88**

5 Conclusion

In this project, we had the chance to implement and compare different deep learning architectures combined with the appropriate preprocessing techniques to gradually improve our model's performance. Among others, we learned that patch sampling is an efficient way to deal with high-resolution images, and EfficientNet achieves good performance on this type of images as well as the attention mechanism. We also learned that ensembling, TTA and using a proper loss can give a performance boost to our models. Overall these techniques allowed us to obtain good results on the public leaderboard.

References

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928, 2016. doi: 10.1109/CVPR.2016.532.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. URL http://arxiv.org/abs/1905.11946.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

https://phortail.org/club-informatique/definition-informatique 136.html. Date de dernière consultation : 26/08/2019.