Deep Learning for medical imaging, Kaggle Challenge Report

Elisabeth DAO ELISABETH.DAO@STUDENT-CS.FR

Timothée RIO TIMOTHEE.RIO@STUDENT.ECP.FR

Abstract

To help assessing prostate tumor severity based on Gleason Score and ISUP grades we implemented several multiple instance learning (MIL) models that classify bag of tiles from whole slide images (WSI) of prostate biopsy. We achieved 0.9166 AUC score on the public Kaggle learderboard using transfer learning (EfficienNetB2), an attention based model and a voting scheme.

Keywords: whole slide images, multiple instance learning, attention, transfert learning

1. Introduction

Each year prostate cancer is responsible for the death of more than 350,000 people worldwide. It is the second most common cancer among males worldwide with more than 1 million new diagnoses yearly. Developing an effective and fast diagnostic method is thus of the utmost importance.

Currently the diagnoses are based on the grading of prostate tissues biopsies that are analysed and scored by a pathologist. This score determines the severity of the cancer and the treatment the patient should be given.

To score a biopsy sample, the pathologist looks for so-called Gleason patterns based on the architectural growth patterns of the tumor and each pattern has a score (from 3 to 5). Depending on the most present and least present pattern, a grade from 1 to 5 is given to the biopsy. This grade is called ISUP grade and describes the aggressiveness of the cancer and its evolution perspectives.

This scoring solely relies on the pathologist expertise and is thus exposed to human mistakes. The pathologist could miss the cancer in the biopsy or could assign the wrong grade to the patterns which would ultimately lead in a mismatch between the patient cancer and the provided treatment.

In the context of this Kaggle challenge, our objective is to automatically assign an ISUP grade to prostates histopathology images. We are provided with a training set made of 340 large mutli-level tiff files containing whole slide images of the biopsy, alongside with their ISUP grade and Gleason scores. The test set on which we have to make the predictions is made of 86 images. The images come from two providers, *Radboud* and *Karolinska* which have also sometimes given segmentation mask with the images (only for the train set) to help localising Gleason Patterns.
In a first part we will describe the various models and pre-processing methods implemented. In a second part we will focus on the models tuning and their comparison.

This competition has two main challenges. At first we have to deal with the very large dimensions of the whole slides images which are sometimes larger than 25000×25000 pixels.

In practice we can not feed those images to a deep learning model as is without overloading our computers RAM. The second difficulty is that we are provided with a rather small dataset, which makes it very easy to overfit. We will thus have to be particularly careful during our validation procedure.

2. Architecture and methodological components

In this section we describe our pre-processing steps , the multiple instance learning approach (MIL) we adopted and the various models that we experimented.

2.1. Pre-Processing

As mentioned in the introduction **??**, one of the main challenge of this project is to deal with very large sized image. To address this issue, a common approach is to tile the images into a set of smaller patches and to work on subsets of those patches instead of working with the whole image.

2.1.1. Tiling

Before implementing any models we started by tiling each whole slide images on their first level (level 0) into patches of size 224×224 (which is the size of the images of the ImageNet data set on which many computer vision models have been trained).

To tile the images we had to be careful because many images have a gray background which is not made of tissue and which does not contain any region of interest. The first step was so to be able to identify regions of the images that contain tissue. To do so, we we applied Ostu thresholding technique which enabled us to discriminate between background regions and tissues. We started by sampling random patches from this area and we made sure that a minimum portion of tissue was present on the patch to save it. However this method was not satisfactory because we sampled only few samples from the borders of the images and some areas could be present several times in several patches. We thus decided to implement a grid tiling, were the whole tissue area of each image is fully tiles with non-overlapping tiles. Our initial method took around 10 minutes to fully process an image. After some research we found a Python library called *Histolab*, which specially optimizes the process of whole slide image thresholding and grid tiling. With this library we obtained much faster results (1.5 minutes for one image).For the train set (340 images) we obtained 67 845 tiles and for the test set we obtained 17 200 tiles. For each tile we also stored the exact location were the tile was extracted. On Figure 1 we plot a sample of 8 tiles from a train set image with their position labels.

2.2. Multiple Instance learning

Since we could not feed all patches at a time to a deep learning model, we decided to use a multiple instance learning (MIL) which is widely use in the context of medical images for instance here (Lu et al., 2021) in the context of brain tumor detection.

Multiple Instance Learning 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. In our case, we know the ISUP grade and the Gleason score for the whole image but some patches

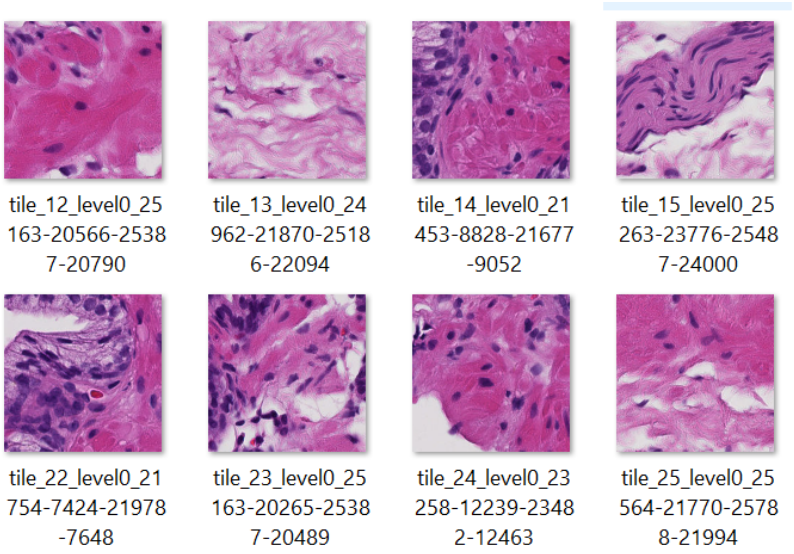

Figure 1: Tiles sample

can contain healthy or ill tissues which do not reflect the grade of the biopsy (since the grade is given only based on some Gleason patterns). We will provide our models with a bag made of tiles and predict the ISUP grade of the image.

2.3. Models

We experimented three main model architectures all leveraging transfer learning. All three models have the same overall architecture which is made of those three steps:

- **Embedding computation**: This step aims at computing one embedding vector for each tile to represent the information it contains. In order to have the most insightful embeddings as possible, we used pre-trained model. Those pre-trained models can be found in the Torch library and most of them were trained on the huge ImageNet data set. Here we chose to work with the EfficientNet-B2 model which achieved very good performances on ImageNet (aroung 80 % of accuracy) and which has a reasonable number of weights (around 10 millions) and could easily be loaded on our computers.

- **Permutation invariant aggregation**: After this step the model contains for each bag a representation made of as many embeddings vectors as the bag contains tiles. In order to obtain one representation for the entire bag, we aggregated those embeddings into one vector. This aggregation function has to be invariant to permutation because the tiles are un-ordered and the order in the bag is even random. We experimented several aggregation such as mean, max and min pooling and also attention based aggregation.

- **Classifier**: The last step is to classify the bag into one of the 6 possible ISUP grades. The classifier takes the bag representation vector as input and outputs a vector containing the probabilities that the bag belongs the different classes.

All our models have the same encoder (EfficienNetB2) and use a dense network with two layers for the classifier. Their main differences are in the aggregation step.

2.3.1. Base Line Model

The first model that we tried uses a simple average pooling aggregation as suggested here (Courtiol et al., 2018), which means that the vector representing the embeddings vectors of the bag is simply the mean of those vectors. We also tried to use a max pooling and min pooling architecture but we obtained the best results with the mean pooling layer.

2.3.2. Chowder Model

As mentioned in this article (Courtiol et al., 2018), the baseline approach works better for 'diffuse' disease, i.e. when "the number of disease-containing tiles, pertinent to the diagnosis label, are roughly proportional to the number of tiles containing healthy tissue". However when exploring the mask images of our data set, we noticed that most of the time the cancerous cells were only localised to some part of the biopsy. In (Courtiol et al., 2018), the authors introduce the Chowder Architecture which uses a completely different approach than the baseline model. They apply a 1 dimensional convolutional layer after the embedding step to obtain a 1d vector representing the bag but this time the vector contains as many elements as the number of tiles in the bag (in the baseline model the vector had the same size as the embeddings vector). This one-dimensional convolution is, in essence, a shortcut for enforcing a fully-connected layer with tied weights across tiles. They then apply a MinMax layer to only retain the instances with minimum and maximum score of the previous vector before feeding them to the classifier. The architecture of the model can be seen on Figure 2.

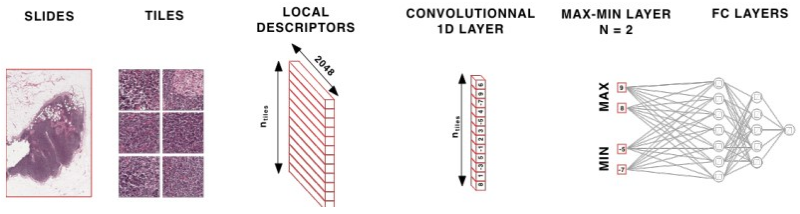

Figure 2: Chowder Model (Courtiol et al., 2018)

2.3.3. Attention based Model

We then implemented an architecture where the model is "free" to chose the contribution of each tile in the bag to the final decision. We thus created an attention based pooling as suggested in (Ilse et al., 2018).In this approach, we used a weighted average of instances where weights are determined by a neural network. Additionally, the weights must sum to

1 to be invariant to the size of a bag. Let $H = \{h_1, ..., h_k\}$ be a bag of K embeddings of dimension M, we used the following attention pooling:

$$z = \sum_{k=1}^{K} a_k h_k \tag{1}$$

where:

$$a_k = \frac{exp(w^T(tanh(Vh_k^T) \odot sigm(Uh_k^T)}{\sum_{j=1}^{K} exp(w^T(tanh(Vh_j^T) \odot sigm(Uh_j^T)} \tag{2}$$

where $U \in \mathbb{R}^{\mathbb{L}\times\mathbb{M}}$, $V \in \mathbb{R}^{\mathbb{L}\times\mathbb{M}}$, $w \in \mathbb{R}^{\mathbb{L}\times\mathbb{K}}$, $sigm$ is the sigmoid function and $\odot$ is a pointwise product. U, V and w are learnt by neural networks. Here we used a gated attention mechanism as suggested in (Ilse et al., 2018) and (Dauphin et al., 2016) to account for the fact that the $tanh$ function is almost linear and might lack of expressiveness for $x \in [-1, 1]$.

3. Model tuning and comparison

3.1. Cross Validation Strategy

As mentioned in the previous section, the amount of training data is rather small and the models can easily overfit. To prevent this issue, we implemented a 10 folds stratified cross validation strategy. Concretely we split the initial training set into 10 parts and we made sure that the distribution of ISUP grades in each fold reflects the original ISUP grade distribution. We obtained 10 training/validation sets that we used to fine tuned our models.

3.2. Metrics and performances

To assess the performance of the models we used the area under the ROC curve (as it is the Kaggle Metric), but we also used the multi-class F1 score. For each models we recorded the metrics on the train set and on the validation set. We quickly noticed that there was a strong overfitting (much higher score on the train sets than on validation sets), so we added dropout in all our models and we implemented an early stopping strategy which stops the training when the performances on the validation sets have not improve for a certain number of periods.

All the training parameters of the models can be found in A. We summarized the main performances of the results (averaged on the 10 folds) in Table 1.

Table 1: Models Performances

Metrics	**Baseline Model**	**Chowder Model**	**Attention Model**
F1 training	0.75	0.78	0.82
AuC training	0.89	0.92	0.93
F1 validation	0.43	0.48	0.52
AuC validation	0.81	0.86	0.89

We also checked the coherence of the models. For instance we plot the attention given to the tiles by the attention model on Figure 3. The image on top represents the mask with

the labels given by the pathologist, while the figure below represents the attention given to the tiles (the more colourful the tile is, the more attention it is given by the model). Of course on this image we only plot the content of 1 bag (180 tiles), but for the final prediction we used the content of several bags. The attention heat-map is rather coherent, since ,as we can observe, the attention weights given to the tiles located in the upper part of the biopsy are very low and the pathologist did not identify any cancerous patterns in this area.

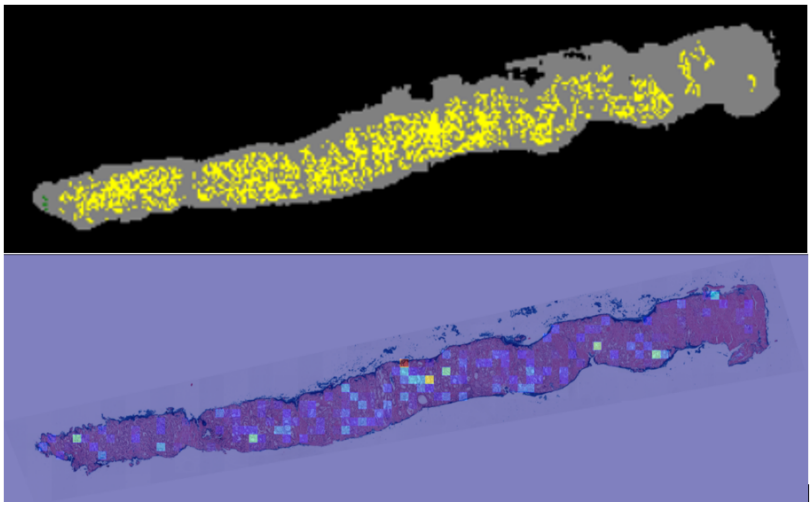

Figure 3: attention map

3.3. Final Prediction

After the training we obtained for each architecture 10 trained instances (one for each train sets of the 10-fold cross validation). For the test set predictions, we used the 10 instances of the attention model (which achieved the best results on the validations sets on average). We sum the 10 predictions (probability vectors of dimension 6) and we took the argmax as the final ISUP grade prediction. With this technique wea achieved an AUC score of 0.89 on Kaggle. However we realised that for each tile and each model we only used 180 tiles (which only covers a subset of the biopsy). To solve this problem, we repeated for each fold the prediction process 100 times (so as to draw randomly tiles 100 times and obtain 100 predictions). As before, we took the argmax of the total resulting prediction vector (sum of all predictions). With this method we achieved 0.91666 on Kaggle. Out of curiosity, we tried to sum for each folds the predictions of the three models but it only deteriorated our performance.

4. Conclusion

For this challenge, we implemented efficient Multiple Instance Learning methods that achieved 0.92 AUC score on the public baseline on Kaggle.
With more time we could have implemented approaches that leverage the location of tiles,

for instance using sparse CNN ((Lerousseau et al., 2021)). We could also try to use the provided mask in order to help the model choosing the most helpful tiles for the predictions ((gro)).

References

Pierre Courtiol, Eric W. Tramel, Marc Sanselme, and Gilles Wainrib. Classification and disease localization in histopathology using only global labels: A weakly-supervised approach, 2018. URL https://arxiv.org/abs/1802.02212.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks, 2016. URL https://arxiv.org/abs/1612.08083.

Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning, 2018. URL https://arxiv.org/abs/1802.04712.

Marvin Lerousseau, Maria Vakalopoulou, Nikos Paragios, and Eric Deutsch. Sparse convolutional context-aware multiple instance learning for whole slide image classification. *CoRR*, abs/2105.02726, 2021. URL https://arxiv.org/abs/2105.02726.

Diyuan Lu, Gerhard Kurz, Nenad Polomac, Iskra Gacheva, Elke Hattingen, and Jochen Triesch. Multiple instance learning for brain tumor detection from magnetic resonance spectroscopy data, 2021. URL https://arxiv.org/abs/2112.08845.

Appendix A. Models and training parameters

In this appendix we summarized the best parameters we found and used:
For all models we used the following training parameters:

- Number of tiles per bag: 180
- Batch size: 16
- Number of epoch for early stopping: 10
- Maximum number of epochs: 100
- Optimizer: Adam (learning rate: 0.0001)
- Dropout rates: 0.25

For the models we used the following parameters:

- **Embeddings Computation**: EfficientNetB2
- **Aggregation**:
  - Baseline Model: 1D average pooling
  - Chowder Model: 1D convlutional layer + MinMax layer
  - Attention Model: 2 small networks with 2 layers to compute the gated attention mechanism
- **Classifier**: Small dense networks with 2 layers