Machine Learning Engineer Nanodegree

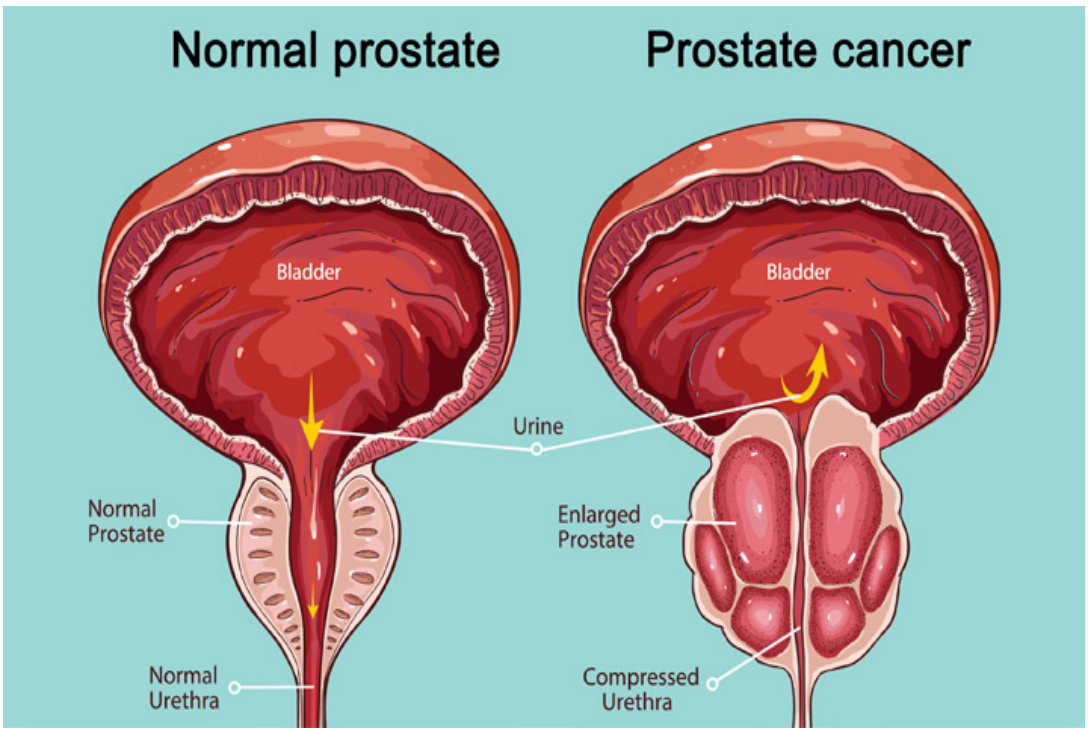
Capstone Project: automating prostate cancer diagnosis on whole-slide image biopsy using patches extraction and TensorFlow

Do Huynh
Github | Linkedin
June 18th, 2020

I. Definition

Project Overview

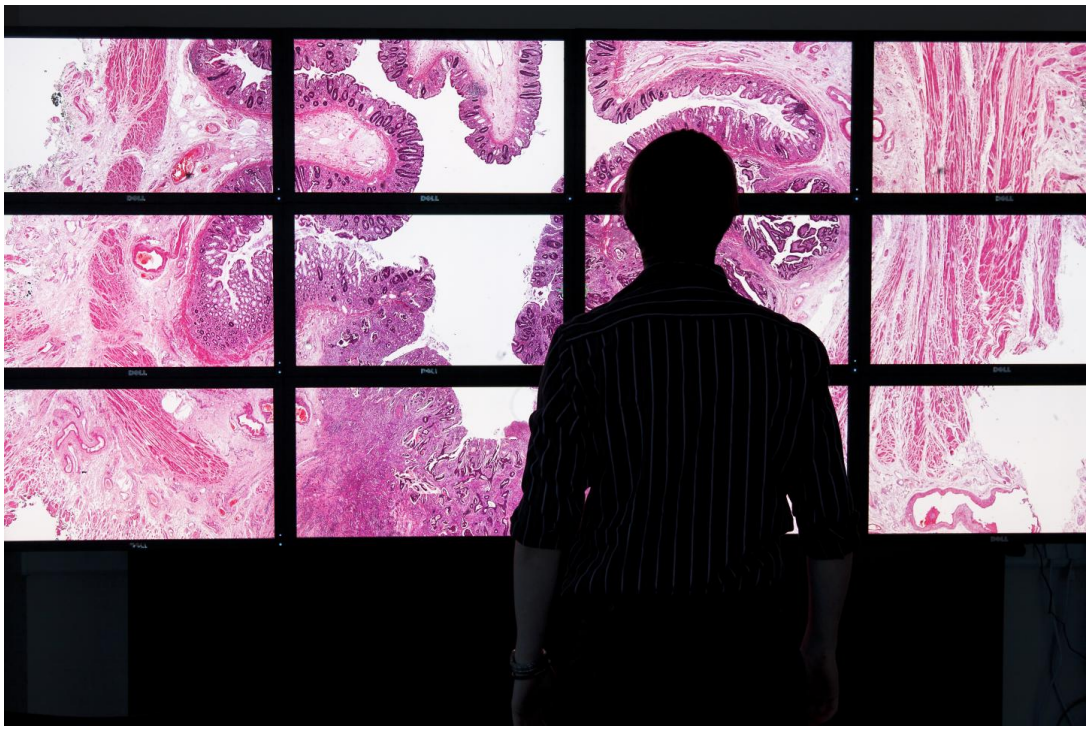
According to the world health organization, cancer is the second leading cause of death in the world with 9,6 millions death in 2018[1] (after cardiovascular disease). Moreover according to the world cancer research fund, prostate cancer is the second most common cancer in men with 1.3 millions new case in 2018[2].



source: Mount Elizabeth Hospital

In the healthcare system, the role of a histopathologist (from greek “histos” = tissue) is to search and detect cancerous cells from a tissue sample (biopsy). It is a sensible responsability depending on the experience of the specialist and the technicals conditions. With the high definition digitalization of

microscopy images (call Whole-Slide Image WSI) and the progress of AI image analysis during the last decade, digital pathology has made great improvement in pathology detection assisting the medicals professionnals in their diagnosis and prognosis assumptions [3].

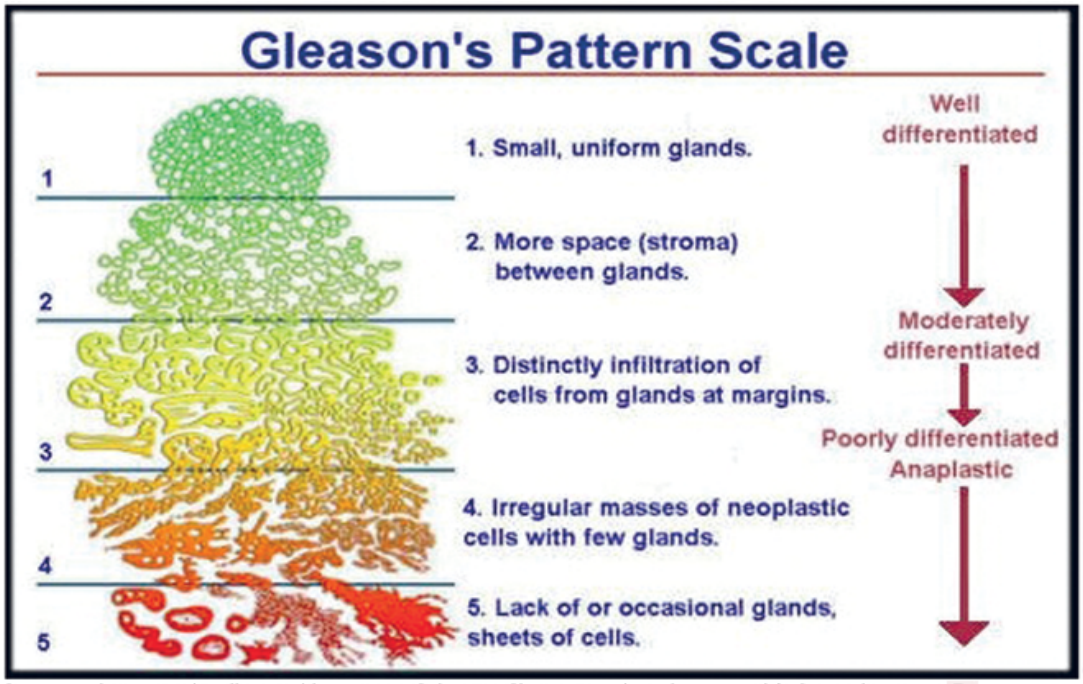


source: section of Pathology and Tumour Biology, University of Leeds

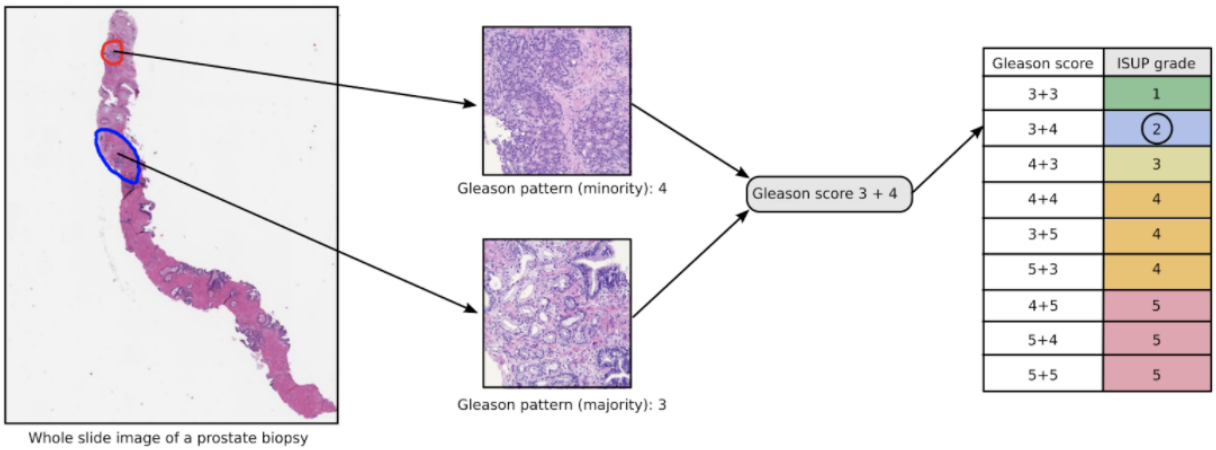
In April 2020, two european medical research institutions (Karolinska Institute and Radboud University Medical) have publish together the largest known dataset of prostate biopsy whole-slide image under the form of a Kaggle competition: Prostate cANcer graDe Assessment Challenge. This dataset is composed of 10,616 prostate biopsies whole-slide images scored by pathologists from the 2 medical institutions.

Problem Statement

Today, the 'gold standard' in prostate cancer detection is the human visual diagnosis[4] : from a whole slide image, a pathologist detect some known cellular architecture, call Gleason's pattern, that inform on the development state of the tumor.



The different pattern are scored according a international grade system call ISUP Grade (from the International Society of Urological Pathology) which globally indicate the state of the cancerous cells (from 1=localized/benign to 5=spreaded/agressive).



Source: [Prostate cANcer graDe Assessment (PANDA) Challenge]

Detecting and determining the good score depends directly on the pathologist's experience and working conditions which can lead to important variation in the cancer diagnosis. Our project will try to propose a solution that use deep learning techniques to automatically evaluate the ISUP grade of a given biopsy image and in fine help pathologist make accurate and faster diagnosis of prostate cancer tissue.

Metrics

The goal of our machine learning solution is to predict the ISUP grade from a whole slide prostate biopsy which is a multi-class categorical problem. Following the Prostate cANcer graDe Assessment Challenge organizers, the evaluation of the submission will be calculate with **the quadratic weighted kappa**[5]:

$$\kappa = 1 - \frac{\sum i,\ jwijOij}{\sum i,\ jwijEij}$$

with:

- O: an N x N histogram matrix such that Oij corresponds to the number of isup_grades i (actual) that received a predicted value j
- W: an N x N matrix of weights Wij calculated on the difference between actual i and predicted values j

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

- E: an NxN histogram matrix of expected outcomes, E, 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.

This specific metric measures the agreement between two categorical ratings varying from negative (less agreement than expected by chance) to 1 (complete agreement). This continuous score variable tend to penalize too far answer from the ground truth but also reward close answer. In fact, it is a more tolerant metrics than an absolute categorical accuracy metric and also tend to avoid unbalanced bias distribution.

II. Analysis

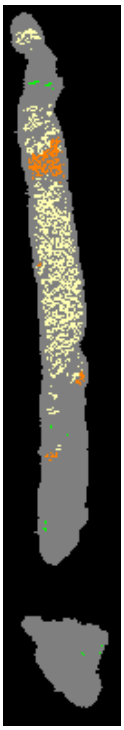
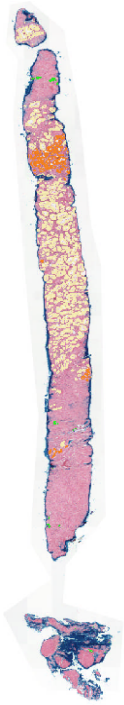
Data Exploration

The provided PanDa dataset is composed of 10616 biopsy whole-slide image (.tiff format), 10516 labelled masks and a table (.csv format) of corresponding gleason score / ISUP grade from the two medical institutes.

The score table contains 4 columns with 10616 unique image ID, 2 providers (Radboud and Karolinska institutes), 11 gleason's score combinaison (from 0+0 to 5+5), 6 ISUP grades (from 0 to 5). Both institute provide equivalent number of slides.

![train_df_head()](https://github.com/huynhdoo/ML/blob/master/prostate-cancer-diagnosis/images/train_head.png

Each WSI contains 3 levels definition (original, x4, x16) than can be open independently. The original definition target for specific medical screen device can not be read and stand fully on local memory. It must be read by region. The labelled mask files are also under format .tiff. They contains in the first color channel a pixel value indicating the gleason's grade scored by a pathologist. However, each institute use different scoring scale (from 1 to 3 for Karolinska institute and 0 to 5 for Radboud institute). For this main reason, we will not used the mask for this version of our project.

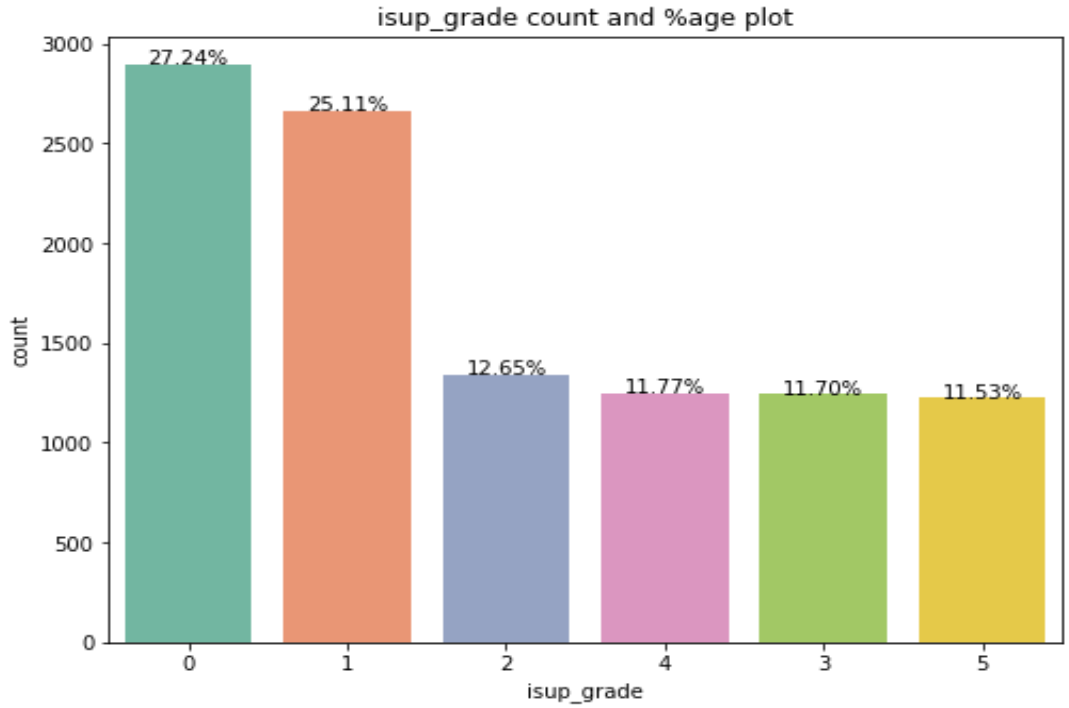
Slide sample	Mask sample	Mask on slide
		

After checking the table, we found only one item with an incoherent value between gleason score 3+4 and ISUP grade 2 (should be 3). We have corrected this line directly during data preprocessing.

	image_id	data_provider	isup_grade	gleason_score
7273	b0a92a74cb53899311acc30b7405e101	karolinska	2	4+3

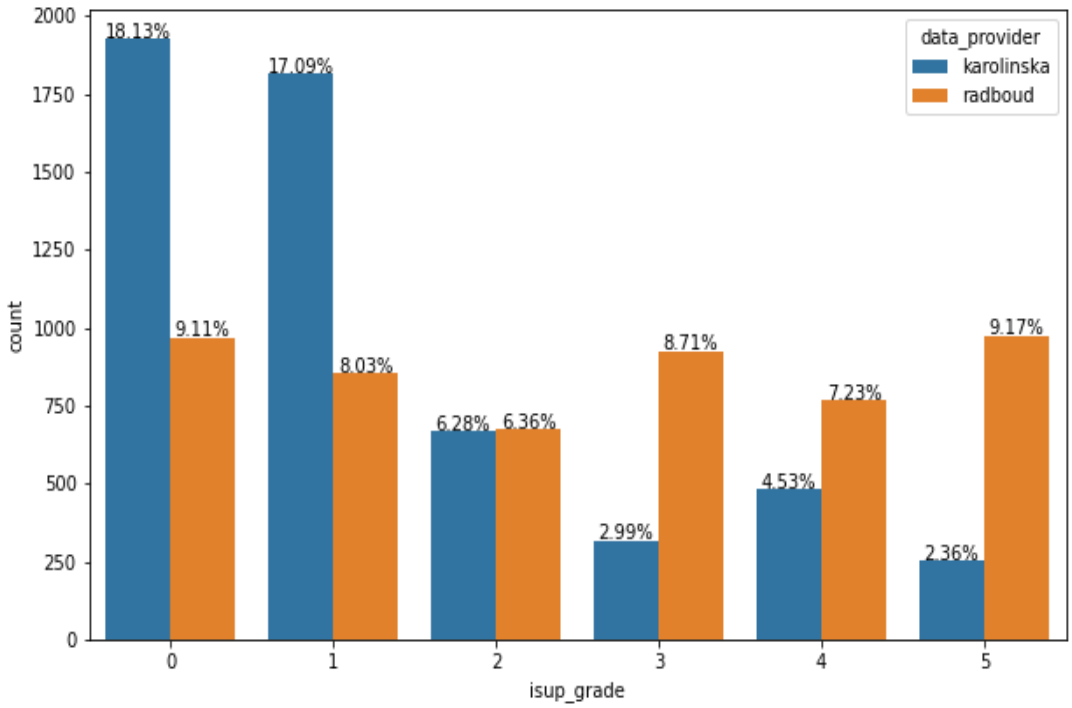
Exploratory Visualization

The distribution of the slide by ISUP grade show that the dataset is unbalanced between the grades [0, 1] and [2, 5]. In fact, their more cases where the biopsy reveal no tumor (+50% of the cases).



Number of biopsy slides by ISUP grade from PanDa dataset 2020

Moreover, if we split the distribution between the two data provider, we can see that Karolinska institute propose a majority of cases grade 0 to 1 since Radboud institute present a majority of cases grade 3 to 5.



Number of biopsy slides by ISUP grade and provider from PanDa dataset 2020

Each slide has 3 levels corresponding to a downsampling factor of 1, 4 and 16. The original image dimensions are quite large (typically between 5.000 and 40.000 pixels in both height and weight). The Tagged Image File Format (TIFF) format can contains some additionnals informations like dimensions, dowsample factor, etc.

```
File id: OpenSlide('/kaggle/input/prostate-cancer-grade-assessment/t
rain_images/07a7ef0ba3bb0d6564a73f4f3e1c2293.tiff')
Dimensions: (24900, 29228)
Microns per pixel / pixel spacing: 0.503
Number of levels in the image: 3
Downsample factor per level: (1.0, 4.0, 16.00457121779945)
Dimensions of levels: ((24900, 29228), (6225, 7307), (1556, 1826))
```

Depending on the laboratory procedures, slides can appear in different colors, rotations, brightness. Because the biopsy tissue represents a small part of the whole slide (less than 20%), we should normalize the size before training by extracting different tile/region of tissue from the original image.

Algorithms and Techniques

Since the advent of deep learning, many AI algorithm using neural network have been test and deploy on medical image analysis[6]. Specifically, two recent studies from the Kaggle competition organizers have demonstrated the relevance of Convolutional Neural Network (CNN) on this type of computer vision problem[7] [8]. In the case of our project, we will also use CNN for our training and predicting model.

One of the limit of actual CNN is the size of the input which need to fit into a maximum height, weight, channel. In the specific case of biopsy whole-slide image, the image are too large to be treated directly by the neural network. The first step of our solution consist in reducing the size of the slide by selecting N relevant patches of the original image (1). Then, for optimal computation, the extracted patches are compress and group together into batch of files that can be read as fast as possible during training process (2). Since the files are ready, we can next apply any image preprocessing needed, split our labelled images into training, validation and testing dataset, train our CNN model and evaluate his performance (3). The main steps are detailed below:

(1) Extract patches from WSI

- Correct any error in the dataset table
- Match images and ISUP grade from dataset
- For each image in the dataset:
  - Extract all patches of size S x S
  - Score and order each patches by pixel color intensities
  - Select N patches with most pixel intensity
  - Concat N selected patches into a unique image
  - Save generated image with corresponding ISUP grade label

(2) Export patches to TFRecords

- Load patches image and label

- For each patches:
  - Compress the image
  - Regroup image into batch of files
  - Save images with label into TFRecords files
  - Put generated TFRecords files on Google Cloud Storage

(3) CNN training

- Split files into training, validation and testing dataset
- For each image in training dataset:
  - Normalize image (set pixel to range 0 to 1)
  - Process random augmentation (flip horizontally, vertically)
  - Shuffle image order
- Define a CNN model:
    - Load pre-trained CNN model
    - Add a 6 class output layer
    - Define kappa loss function
    - Define optimizer and learning curve function
- Split into K folds training/validation dataset for cross validation
- Train CNN model on each training/validation fold dataset
- Save the weights of the trained CNN
- Test predictions of the model on the testing dataset

Benchmark

The Kaggle challenge leaderboard is a good starting point to benchmark our solution with other competitors. The submitted predictions are made on around 1000 unknown biopsy. But this measure is only on the absolute accuracy of the machine learning model, not on his capacity to be easily deploy. However that help us figure out what is the possible best result on the given dataset without any resources concern.

For the purpose of our project, we can take as base performance on a more realistic goal: according to the institutes organizers, the actual kappa agreement between expert pathologist in inter-study is between 0.60 and 0.73[7] [8]. Building an AI model that reach or outperform this range will mean that we have a possible helpful model.

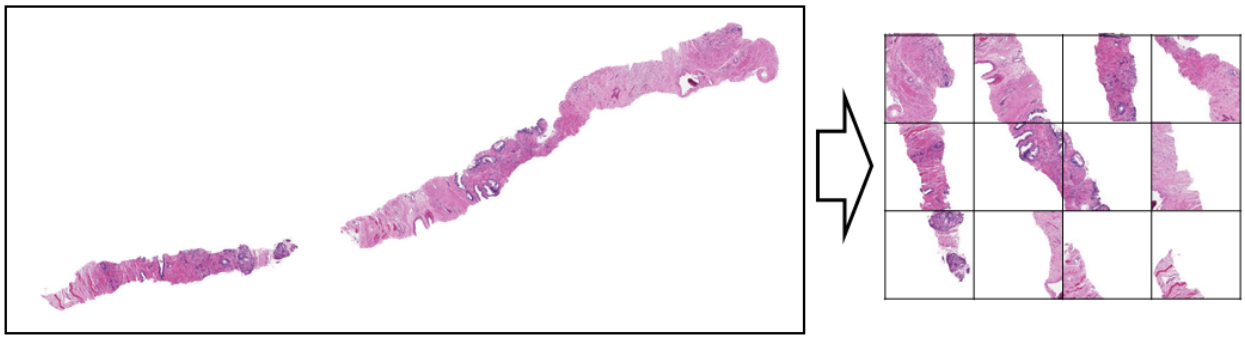
III. Methodology

Data Preprocessing

As we have seen previously, the image slides are too large to be process directly by a CNN which can only work on a maximum size between 128x128 pixels (on GPU) and 512x512 pixels (on TPU). To read the slide, we must reduce the information to the most relevant part by extracting some

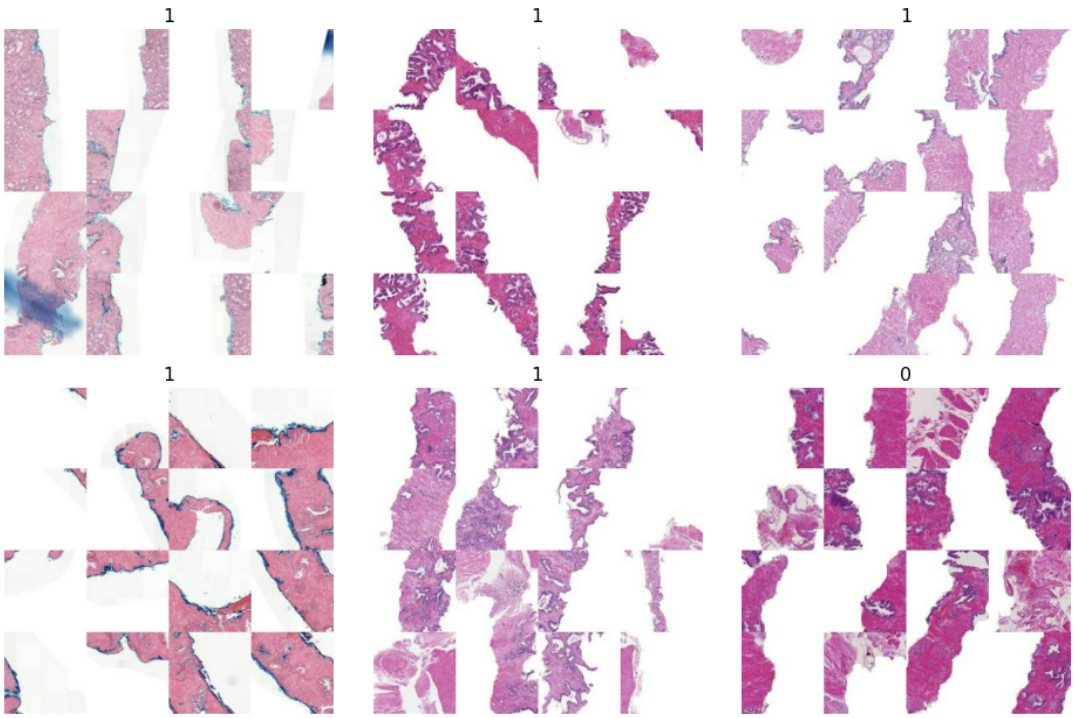
patches from the original image and concat them into a single image that fit into any CNN. Moreover, the relevant tissue occupy less than 20% of all the slide size.

For this purpose, we have first extracted all the patches of 128x128 pixels from the lowest level dimensions. To read specific region of a WSI without loading all the image in memory, we have used the python interface of the Open slide library. Then, each patches are scored and ordered by the sum of their pixel color intensities (from 0=black to 255=white). Why using color intensities ? The pathology process use a coloring technic that stain in pink and purple (call hematoxylin and eosin staining[9]) the tissue and nuclear cells. Knowing that in one hand the concentration of nuclear cells reveals the development of cancerous cells and in other hand, nuclear cells stain in purple are darker than other cellular tissue, we can reasonably assume that a darker patch are more relevant than a lighter patch.
After scoring the different patches by pixel color intensities, we keep the N firsts patches and concat them into a square image of size 128x128xN (ex: 128x128x16).



Source: https://www.kaggle.com/iafoss/panda-16x128x128-tiles

As the dataset is quite small for a deep learning purpose, we have double its size by generating 2 different patches split. The second split is half shifted from the first grid so it generates different patches from the same image (like 2 eyes or cameras looking at the same object).



128x128x16 patches on lowest level slide definition

All generated images (10616 x 2 = 21232 images) are then saved and split into different ISUP grade folder on Google Cloud Storage for the next processing. All this previous steps are implemented in the notebook **PANDA 1 - Extract patches from WSI.ipynb**

Implementation

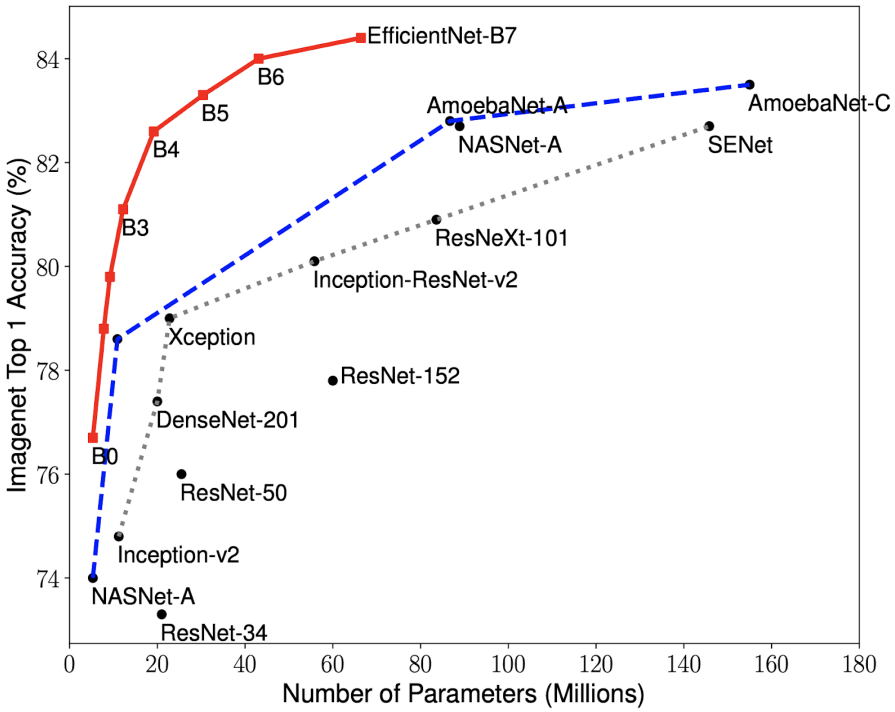
Training a CNN even on a GPU need a lot of times and resources. To make this step faster, Google Colab Notebook provide some free TPU (Tensor Process Unit) that are specially optimized for tensor computation. But with high computation flow, the file reading process can slow down the whole calculation. In consequence, the image files must be group into large files (format TFRecords of size 100 to 200 Mb) that can be read directly by CNN running on a TPU board. The ISUP grade of each image are also encoded as parameters. The aim of our second notebook is to prepare this optimized files and saved them again to Google Cloud Storage. All this part is implemented and documented in the notebook **PANDA 2 - export to TFRecords.ipynb**.

We have choose to build the convolutionnal neural network with Tensorflow/Keras architecture. Indeed, TPU board host by Google colab are optimized for Tensorflow architecture (even if we could also read TFRecord file with pytorch using appropriate library). Keras is a high level API build on Tensorflow that help iterating fastly during model definition.

Assuming that the datas are previously encoded in large TFrecords file, we use Tensorflow dataset to load and read on the fly the images and labels (ISUP grade). The whole dataset is split and batch into training, validation and test dataset (respectively 85%, 12% and 3% size ratio). On the training dataset, we make some data normalization (set pixel value in range 0 to 1), data augmentation (image flip) and data balanced (apply a weight on the different target class to avoid distribution training effect). The validation dataset is used during the training process while the test dataset is keep aside for a final model evaluation. For accuracy robustness, we also apply a cross validation process by splitting our training step in K-folds (10 folds).

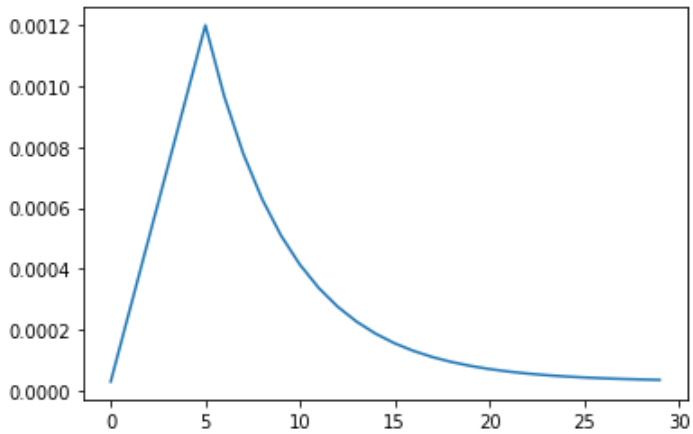
```
TOTAL IMAGES: 21230 - FOLDS: 10
TESTING IMAGES: 512
REMAINING IMAGES: 20718
----------------------------
TRAINING IMAGES: 18158 - STEPS PER EPOCH: 141
VALIDATION IMAGES: 2560
----------------------------
```

Using transfer learning, we build our CNN model on a EfficientNet architecture pre-trained on ImageNet dataset. Publish in 2019, EfficientNet[10] is an ensemble of deep learning models for computer vision that optimize the trade-off between the number of parameters to train and the final accuracy. For now, there are 7 models (from B0 to B7) that can be used progressively to increase the performance of the final model. As output layer, we just add to EfficientNet a 6 dimensions one-hot encode vector corresponding to the predict ISUP grade (from 0 to 5).



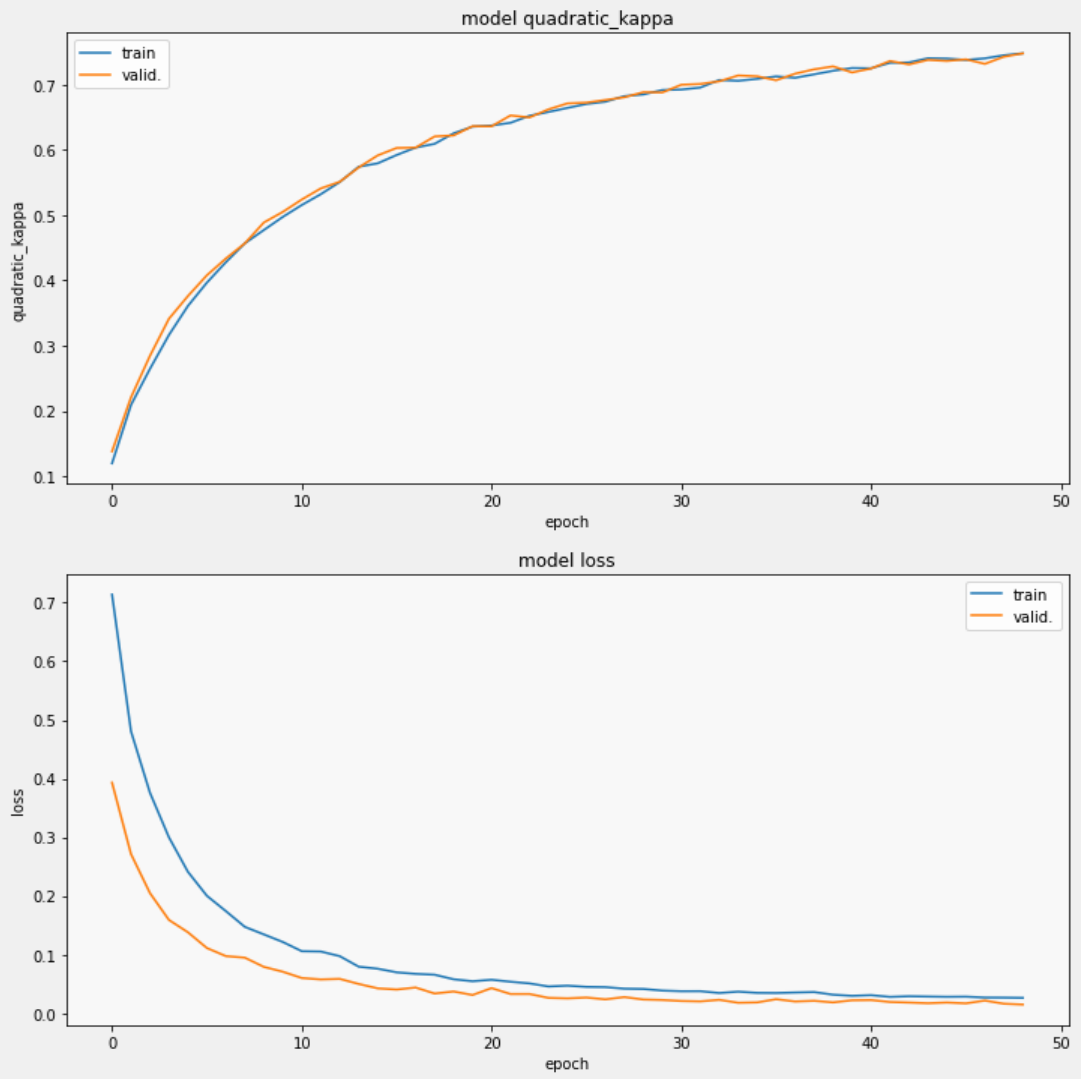
Source: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

To avoid breaking pre-trained model, we apply a “mountain” learning curve that starts with a low learning rate during the first epochs (3e-5), going up and finishing down back to starting rate during 30 epochs.



Learning rate schedule: from 3e-05 to 0.0012 back to 3-05 during 30 epochs

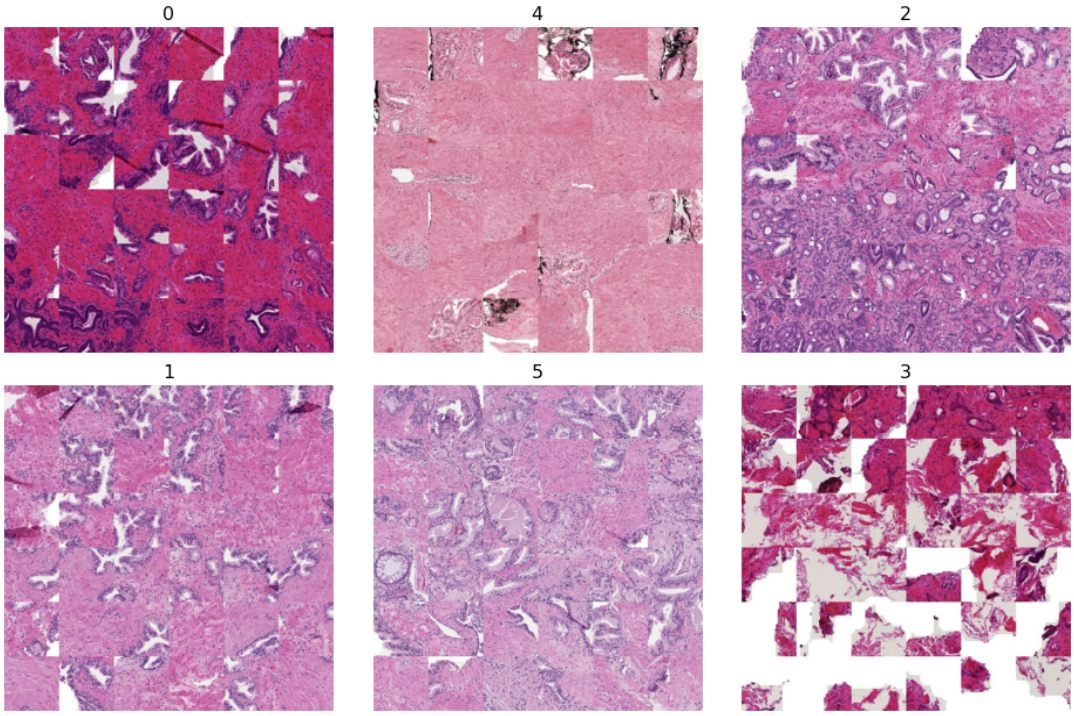
Our initial solution train on 16 low level definition patches of 128x128px with transfer learning from EfficientNet achieve a score on Kaggle test of **0.68 (B0), 0.70 (B4) and 0.73 (B6)**. These initials results are all in the range of the inter-rating between expert pathologist.



EfficientNetB6 model train kappa and loss during 50 epochs - 0.73 kappa score

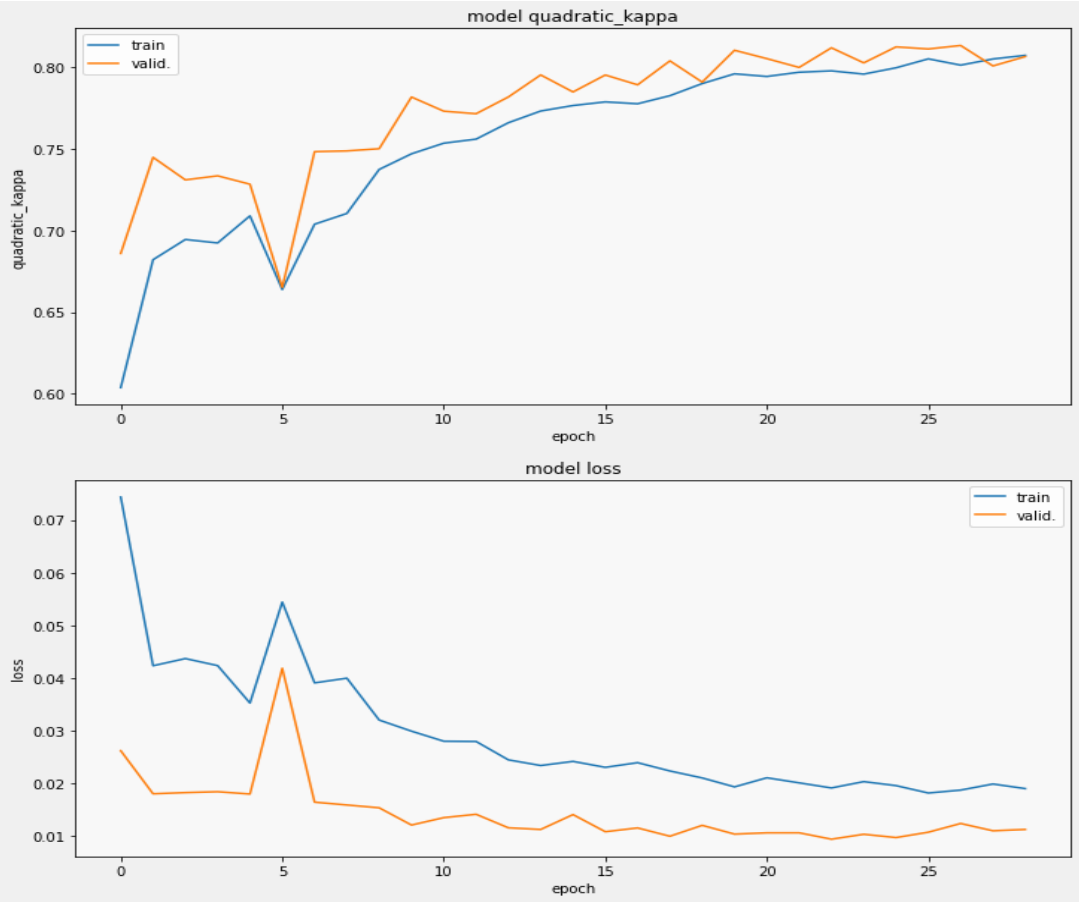
After the training process, we make some predictions on the testing dataset to evaluate how our final model generalize on unknown datas. All this last part is implemented and documented in the notebook **PANDA 3 - CNN training on TPU.ipynb**.

Refinement



128x128x36 patches on medium level slide definition

Some other solutions propose by Kaggle competitors achieve a better score by using higher definition image and larger input size. Indeed, with the same parameters, when we fit our model on 36 medium definition patches of 128x128px and same training parameters, we improve significantly our kappa score **up to 0.80** which out-performed the upper score of the gold standard.



EfficientNetB6 model train kappa and loss during 30 epochs - 0.80 kappa score

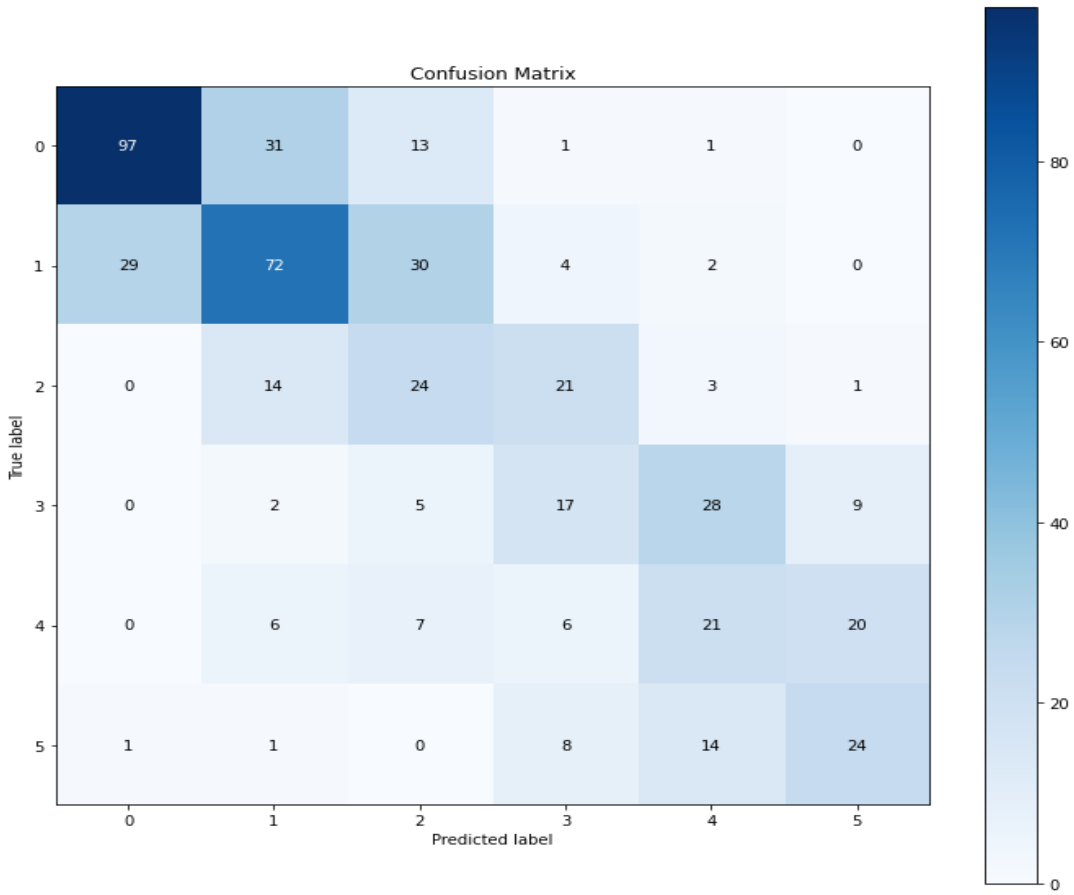
IV. Results

Model Evaluation and Validation

In the aim to help our model generalize inputs datas and prevent overfitting, we have introduce some random variations on the inputs. We have also doubled the size of the original dataset by applying two different window stride during the patches generation step. And for robustness of our model, we apply a cross validation method to train and validate our model on different split of our initial dataset.

For final test accuracy, we have kept aside 512 images for an internal test before submitting our model to Kaggle evaluation.

If we look at the correlation matrix, we can see that the main mislabelled slides are around ISUP grade 1 and 4. Our model tend to be more pessimist than the ground truth. In the case of medical diagnosis, that is better than underestimate the problem.



Justification

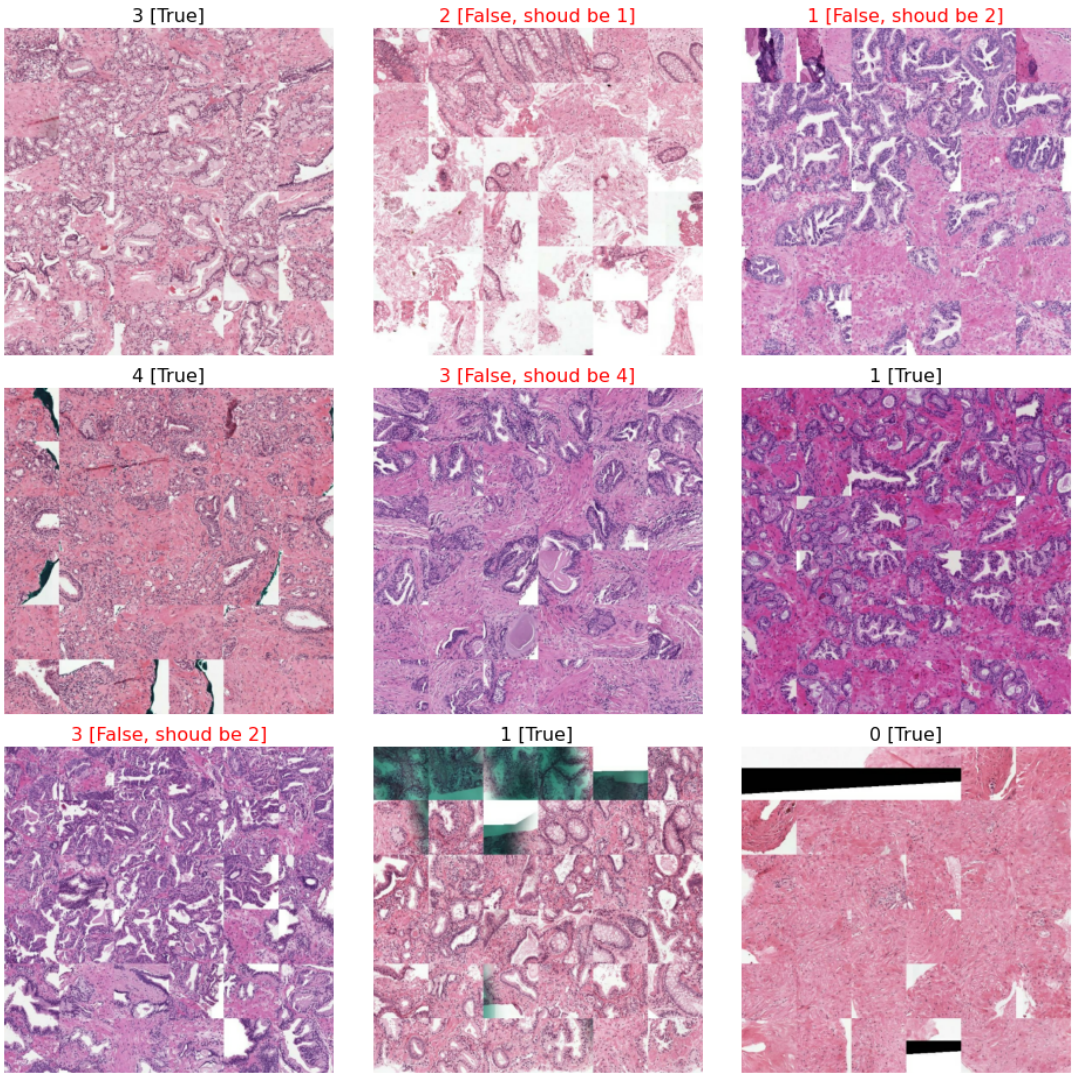
Our final CNN model achieve a score of 0.80 kappa agreement on unseen test dataset (both internal and external). Although a very simple patches selection, this is quite a promising model that exceed the inter-ratings standard evaluation by expert pathologist. The final notebook for Kaggle submission is details in [PANDA 4 - Kaggle model inference.ipynb]

At the moment, the best solution on [Kaggle prostate cancer grade assessment leaderboard] achieve 0.91 kappa score! Our implementation as a generic pipeline can be clearly improve and also reuse any pre-trained model including the best one. Moreover, with the optimization on TPU computing, **the training process is quite fast and cheap with less than 1 hour to fit a model on 30 epochs**.

V. Conclusion

Predictions visualization

Our model can now diagnose prostate cancer ISUP grade from whole-slide image biopsy with an estimated kappa confidence of 0.80. Here, we can see some predicted label corresponding with ground truth:



Reflection

In the limitation of the given dataset, our solution out-performed the gold standard of human prostate cancer grading with a kappa agreement of 0.80. This baseline solution show that with a minimal, fast and low-cost deep learning CNN based on pre-trained EfficientNet architecture,

prostate cancer detection on inter-studies biopsy can be achieved. Of course, further developments should be done to improve the overall performance and build a strong model that generate consistent ratings over any biopsies from other institutions.

Beyond accuracy, our solution shows 3 main points:
(1) The image patching operation is an essential and efficient step prior to deep learning training.
(2) Batching images into optimized files for processor drastically reduce the training time and by consequence the iterative process of modeling.
(3) Transfer learning from pre-trained EfficientNet architecture is a good starting point for Prostate cancer diagnosis from biopsy whole-slide image.

Improvement

During image patches extraction, we have to made a trade-off between image definition and memory capacities. Indeed, with a higher definition, we can surely have better information for the training process but it takes more memory to train. Even with a TPU process, we can only manage a maximum of 768x768 pixel image. Having a too much deep zoom level will limit the vision zone although retaining a too low level zoom will produce too much noises.

In our opinion, one the biggest potential improvement is to develop a better patches selection. A possible solution could be to train a CNN segmentation model on the given labelled mask to score each patches according to the recognize cellular pattern and filter non tissue or benign zone. This segmentation step could help removing any noise on the dataset before training and in fine improve the model performance.

Acknowledgement

Because *"I am just standing on the shoulders of giants - Isaac Newton"*, I am fully grateful for these inspiring works from machine learning magicians :

- Images patching
- Loss and metrics kappa function
- Keras training on TPU
- Keras kaggle baseline

---

1. World health organization - https://www.who.int/news-room/fact-sheets/detail/cancer ↩
2. World cancer research fund - https://www.wcrf.org/dietandcancer/cancer-trends/prostate-cancer-statistics ↩
3. Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6880861/ ↩

4. Clinical histopathology of the biopsy cores : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1769959/ ↩

5. PANDA challenge evaluation metrics - https://www.kaggle.com/c/prostate-cancer-grade-assessment/overview/evaluation ↩

6. The medical image analysis journal publish monthly a selection of the last research paper on deep learning techniques and algorithms apply to medical images (MRI, CT, echography, etc.) - https://www.sciencedirect.com/journal/medical-image-analysis/vol/62/suppl/C ↩

7. Pathologist-Level Grading of Prostate Biopsies with Artificial Intelligence - https://arxiv.org/abs/1907.01368 ↩ ↩

8. Automated Gleason Grading of Prostate Biopsies using Deep Learning - https://arxiv.org/abs/1907.07980 ↩ ↩

9. Hematoxylin and eosin stain https://en.wikipedia.org/wiki/H%26E_stain ↩

10. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks - https://arxiv.org/abs/1905.11946 ↩