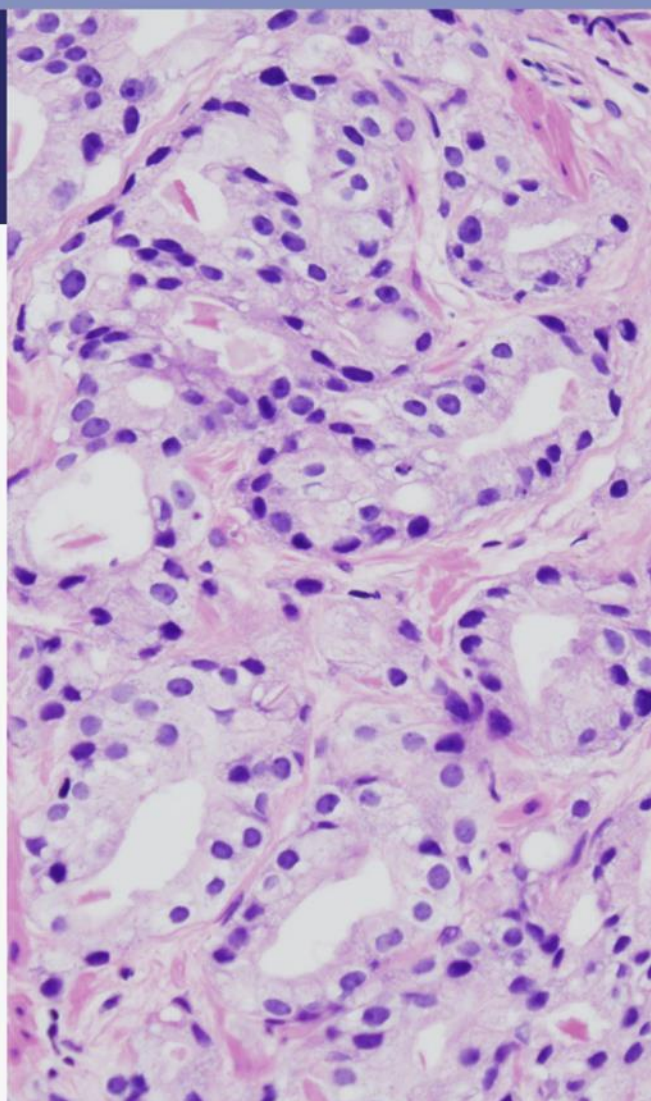
JUNIA ISEN

First Semester Project Report :

Classification of microscopic images of cancerous tissue using Artificial Intelligence

GROUP 18 - MASTER 1



STUDENTS :

Marine GEORGES
Harith JADID
Carlos JIMÉNEZ GARCÍA
Michel SAUVAGE
Adrien Junior TCHUEM TCHUENTE

PROJECT REFERENTS :

Bilel GUETARNI
Feryal WINDAL

2022-2023

Introduction - Context

As seen in the first semester, non-Hodgkin's lymphoma (NHL) is an aggressive cancer that affects the lymphatic and immune systems. Diffuse large B-cell lymphoma (DLBCL), which accounts for 30-40% of NHL cases worldwide, is the most common subtype and can be fatal if left untreated. Two main versions can be found for this subtype: ABC (Activated B-Cell) and GCB (Germinal Center B-cell), which have a higher and lower mortality rate, respectively.

In the framework of our project, we aimed at implementing and training a powerful Machine Learning system able to classify, from the biopsies of a given patient, the subtype of diffuse large B-cell lymphoma (ABC or GCB) he has. For this purpose, we were provided with biopsies in the form of WSIs (Whole Slide Images) from about 160 patients, which were annotated by anatomopathologists.

This document develops the different steps of the technical implementation of the project. The creation of the dataset is discussed in a first part, followed by the implementation of the classification models of the tissue (cancerous or normal) and the subtype (ABC-GCB). A detailed explanation of the final prototype is also proposed.

I. First part : Dataset creation

First, we had to create the dataset for the project. This step was essential in order to guarantee the smooth running of the rest of the project. Indeed, the data manipulation is relatively complex because for each patient, several WSIs can be associated. Moreover, annotations for each of these WSIs were provided and represent either the cancerous areas on the image or the non-cancerous areas. Therefore, rigor was required to organize all of this data in an optimal manner and in appropriate folders.

The dataset was created using the following steps. First of all, knowing that the WSIs are very heavy images, we needed to use a library able to open and use them. For this, we decided to use Openslide. Another option we tested was to use the library « cv2 » . However, this library gave us access to unwanted parts of the image (labels, edge of the blade...), which were not present with Openslide. In addition, Openslide allows you to select the level of detail/zoom that you want, which is advantageous to reduce the processing time of the images by the computer.

For each image, we separate the tissue from the background with a mask created with the cv2 library. The mask is created thanks to a threshold of coloring of the image framed between two values (too dark or light colors) determined manually thanks to tests. This method allows to select only the tissue areas (80% tissue) and without artifacts or too dark areas.

Then, we use another mask to identify, this time, the annotation areas. We retrieve from a .geojson file the coordinates of the anatomopathologists annotations in order to identify the areas in our image. To do this, we draw the shapes according to the available coordinates and we fill them with white. Everything outside is black. For this step, the cv2 library was also used.

Finally, we extract the patches according to the following method. We look at the image by moving patch by patch. For each patch, we proceed by steps. We use the fabric mask. If the patch is located in a white area (tissue area), we go to the next step of the mask of annotation areas. It is important to note that there are two types of annotations : those surrounding a cancer area (outside = non-cancerous) and those surrounding a non-cancerous area (outside = cancerous). This is well taken into account in the algorithm. The patch is then saved in the cancerous or normal folder according to its characteristics. The patches are saved at a zoom level of 1 according to Openslide, which allows to have a good image resolution.

Path = ProjectM1/data_saved/n° patient/name slide WSI/cancerous or normal/patch.png

II. Second part : classification

1. Group data into train, validation and test sets

Before starting the classification, it is necessary to separate the data into different sets. For this, we based ourselves on the patients and their number of patches. Initially, we had chosen to separate the patients into three sets: train (70%), validation (15%) and test (15%). To make the division, we based ourselves on the number of patches by creating thresholds. However, a patient can have a part of his patches below and another one above the threshold. In this case, the patient is assigned to the next category to avoid duplication. Three folders are created in the working directory in order to arrange the patients according to their category. Thus, in each train, validation and set folder, there are symbolic links to the patient folders to avoid moving previously saved patches.

2. Model for classification of cancerous/normal patches

Plan A - Development of a CNN :

After the state of the art established in the first part of the project, we choose to develop a CNN that was initially based on the SEF model used in the paper "A Deep Learning Approach for Breast Invasive Ductal Carcinoma Detection and Lymphoma Multi-Classification in Histological Images" (Brancati et al. 2019) , because it provided the highest "accuracy" (0.97). However, we could not find the structure of this model and decided to try the second option, which was to use a ResNet model with an "accuracy" metric only 2% lower than the previous model.

ResNet (Residual Networks) is a CNN architecture that is characterized by its ability to train very deep neural networks with a small number of parameters. It is widely used in image processing. These neural networks are made up of residual blocks (which could be called the basic unit of the ResNet), which are grouped together and give rise to 'stacks'. A `stack' can contain different amounts of residual blocks depending on the desired depth of the network. In this way, we can find different types of ResNet, with the largest number having greater depth and the greatest ability to learn more complex feature representations and adapt to the learning parameters.

Since we had a very large dataset for training, we initially decided to test ResNet50. However, as the number of parameters to fit was too high for the time available, we decided to reduce its size and develop a ResNet14, prioritizing time over accuracy. The results obtained for the best model were an accuracy of 86.7% and a precision of 71%.

It should be noted that we would have liked to have had the time to continue testing the ResNet and adjusting the parameters, or even to use PyTorch, a framework that would have allowed us to debug the code and experiment with different models more easily, as well as to customize the different strata of the ResNet, thus creating a much more specific model for the task at hand.

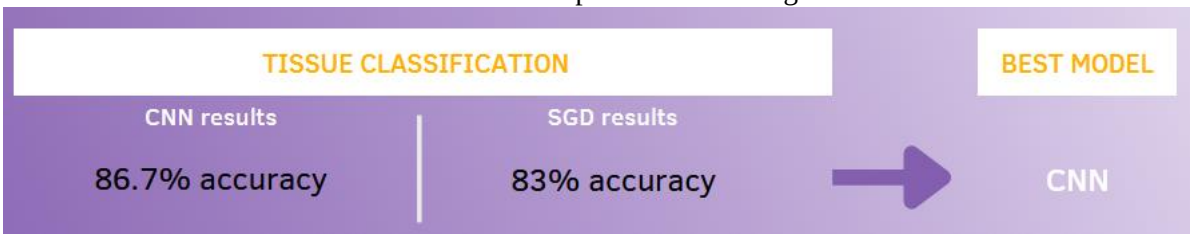
In parallel, we also decided to train a Stochastic Gradient Descent (SGD), used for cancer type classification, but adapted, in this case, to distinguish cancerous from non-cancerous tissue, in order to have an alternative in case ResNet did not produce the expected results.

Plan B - Development of SGD (Stochastic Gradient Descent):

The second agreed-upon option for classifying cancerous and normal patches was therefore to use an SGD, which is a linear classification model approach. The SGD applied here is the same as the one originally planned for the next part; only the input differs between the two versions. Here, we will therefore have as input for the training patches labeled cancerous and normal, depending on the file from

which they come. At the output of the model, only the patches predicted as cancerous are kept for the next step. Its complete operation is detailed in the next section.

The best results for this classification step are the following:



1 - Meilleurs résultats

3. Model for the classification between the subtypes ABC and GCB

For this step (as well as the previous one), we used an SGD, which is similar to an SVM. The particularity of this SGD model is that it is able to run in mini-batches (partial_fit). This is essential for this project given the amount of data to process. In addition, we had highlighted in the first part of this project in December the interesting use of the SVM model for image classification, according to the paper "Multi-radial LBP Features as a Tool for Rapid Glomerular Detection and Assessment in Whole Slide Histopathology Images" (Simon et al. 2018) .

First, the data needed for the model is retrieved as input. In the previous step, patches labeled cancerous or normal were used. Here, we use cancerous patches only. The latter are labeled with the ABC or GCB subtype according to the information previously provided for the project in a ".csv" file. It is important to note that the data were finally divided into two sets train (85% = patient records train + test collected) and validation (15%). It was not significant to keep a test set because it was not possible to compare the results of this project with another project already carried out before with a similar method and objective. The information related to each patch is stored in a tuple, which makes it easy to manipulate.

Then, in order to train the model, a batch system was created. The number of batches and their size were deduced in each case from the number of patches we have in total in order to optimize their use. To perform the partial fit, it is necessary to use a loop to go from batch to batch. Within this loop, we first perform a Local Binary Pattern (LBP) on each patch and then we make a histogram. We recover the features of the histogram (concatenated) as well as the labels. For information, if a patch is badly recorded or does not have the right dimensions, it is not used for the rest of the program. The path of the patch and the corresponding label are accessed through the tuple made previously. We then carry out the training thanks to the partial fit with the features according to the classification classes (cancer or sub-type as the case may be). We then delete the contents of the batch to free up RAM space.

Thanks to the joblib library, we save the trained model. Although it was not possible to create all of them due to lack of time, we had ideally chosen to use the following metrics to evaluate the model:

- Accuracy: the percentage of correct answers.

- F1-Score: calculates the average of the precision and recall.

- Confusion matrix: displays in a matrix the number of True Positives, True Negatives, False Positives and False Negatives.

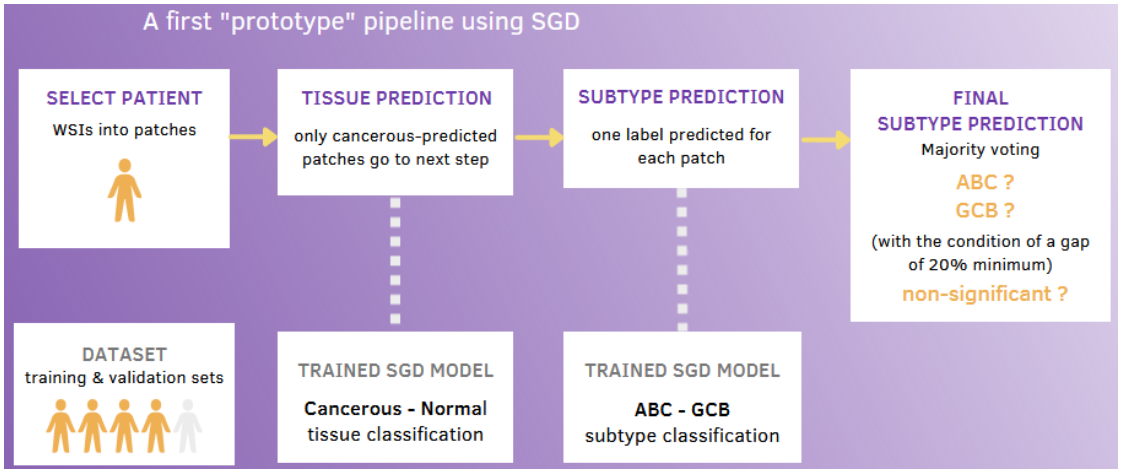
- ROC AUC : allows to know if there is a marked difference between the two classes.With the accuracy, we can determine if the model is very wrong or not. The F1-score allows to determine for

each class what are the performances of the model. The confusion matrix allows to visualize, by class, how many times the model is wrong or has determined the right result. Finally, the ROC AUC allows to specify if the model is able to differentiate the two classes. It can be noted that it would have been interesting to have access to the metrics by patient to improve the quality of the information.

Moreover, two losses were tested for the model: hinge (linear SVM) and modified_huber (brings tolerance to outliers and probability estimates). These gave quite similar results. In a second step, we also tried to balance the ABC-GCB dataset in order to have better performances.

After different tests, the final results of the training and validation of the program showed that we obtain an accuracy of 70% for the best model, with a balanced dataset but which did not give a constant result. To improve this, we tried two different approaches. The first was to use more batches but training a smaller amount of patches at a time because the assumption here was that the partial fit was so large that it made the old training negligible. This did not yield improved results. Another method tested was to use a Support Vector Classification (SVC), training 100,000 patches at once. Indeed, the assumption here was that one cannot train the whole dataset because of its very large size (1700000 patches). So, maybe with 100 000 randomly taken from the dataset, we can get better results. As the results are not yet available, it is not possible to conclude on this test.

III. Final program and technical explanation



2 – Overview of the proposed prototype

Finally, a pooling of each of the steps was carried out to produce a first prototype of the whole program. Thus, as an input to the program, a patient whose cancer subtype we want to know is selected. In order for the program to work as expected, the WSIs related to the patient must be recorded in a file named by the patient's identification number. For each of its slides, the program will perform all the steps discussed beforehand in a loop and the final decision of the model will be made at the very end.

As a reminder, the first step is to make the 512x512 pixel patches of the WSI. These are saved to ensure that the code works properly in the event that a patient has many WSIs. The patches are all saved together in one folder. The patient number and path of each patch is kept in a tuple.

Then, the first SGD classification model starts. For each batch, the LBP is performed to create a feature histogram and then the prediction is performed. As an output of this first model, a tuple is created keeping only the information of the patches predicted as cancerous.

For the second classification model, we proceed in the same way, but this time using the tuple created in the output of the first model. For each patch, a LBP is redone, then the model predicts the subtype for each patch by batch. These predictions are recorded in a list, which is completed as the patient's WSIs are added.

Finally, once all the patient's WSIs are passed to the program, the list of ABC or GCB predictions is used. Based on the number of ABC and GCD patches predicted, we will be able to determine what type of cancer the patient has using the majority voting method. In the case of a difference of less than 20% between the total of ABC and the total of GCB predicted, no result is given as it is considered not significant.

On the other hand, due to lack of time, it was not possible to test the program further with the CNN or other parameters and metrics. Nevertheless, this brings a first confirmation of the possible sharing of the created models and could be reused to improve the whole realization.