AI for Medicine - Project Report

Prof.	Stefano Diciotti – University of Bologna
Academic Year :	2024-2025
Student	Clément Klein
Degree Program:	Erasmus, Biomedical Engineer student
Submission Date:	June 2025

Project title

Classification of histological slide images from colorectal cancer patients using a CNN algorithm

Problem statement

Colorectal cancer (CRC) is the second most deadly cancer in the USA [1], assessing its early prediction is then a milestone for public health, as it is easier to treat it in advance and with more soft methods for the body.
One of the main spread techniques for agreeing on a prognostic is the use of histological slides. Hence, after collecting a tissue sample from patients, some chemical treatments are performed to keep the tissue properties intact while working on it. Then, each sample is divided into thin layers, to allow microscopical observation. The goal-standard assessment to emphasize the contrast in different tissue types is the use of hematoxylin–eosin ($H\&E$) dyed tissue. Indeed it colors the tissues in purple or pink depending on the pH [2].
Since stroma cells (cancerous cells) are not everywhere in the colon, we need to spotlight their presence and location, this is done while professionals categorize tissue slices obtained from patients.
Using artificial intelligence (AI) to find the cancerous pattern slices, and then be able to recognize the cancerous stroma in tissues, can save time for the oncologist.

Objective of the study

In this study, a convolutional neural network (CNN) was used to classify a set of histological images of CRC patients into nine classes, including cancer-associated stroma. The goal was to achieve great global performance on the classification, but also, more accurately, on the categorization of cancerous tissues.

Data description

0.1 Division of the data set

The data set used is the PathMNIST one, composed of histological dyed slices with $H\&E$ method and digitalized normally with the Macenko method [3]. The set used is divided into two different sets : the development set (100 000 patches from NCT-CRC-HE-100K) and the test set (7 180 patches from CRC-VAL-HE-7K). Patches come from the tissue slices collected from CRC patients, which have been divided manually. It is said that the images (patches) in and within the sets are non-overlapping. [4].
With these two sets of images, a first preprocessing has been done, namely resizing the images from a 224*224 size to a 28*28. Then the development set has been split into two subsets, namely 90% for the training set and 10% for the validation set [5]. To resume, the PathMNIST set was used, it is composed of a training set (89996 images), a validation set (10004 images) and a test set (7180 images).

0.2 Constitution of each set

Thus, three separated sets are in the PathMNIST set, each of them is made of list of sub-lists. Each element in each set is a list of two elements, namely the image matrix and the label attributed to the image, which was realized by an expert.

On the one hand, the images size is 28*28*3, hence 3 channels include a 28*28 image matrix representation. On the other hand, the labels are ranged from 0 to 8, for a number of 9 labels in total. The labels have been attributed as follow and some instances are shown in Fig1:

'0': 'adipose' '1': 'background' '2': 'debris' '3': 'lymphocytes' '4': 'mucus' '5': 'smooth muscle' '6': 'normal colon mucosa' '7': 'cancer-associated stroma' '8': 'colorectal adenocarcinoma epithelium'

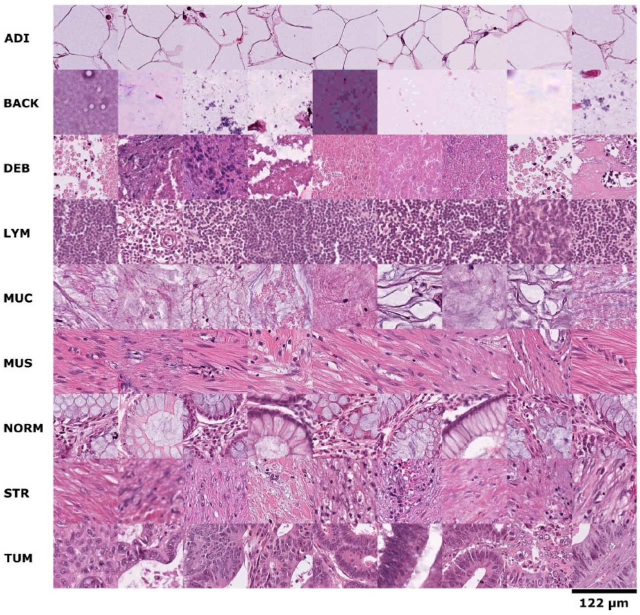
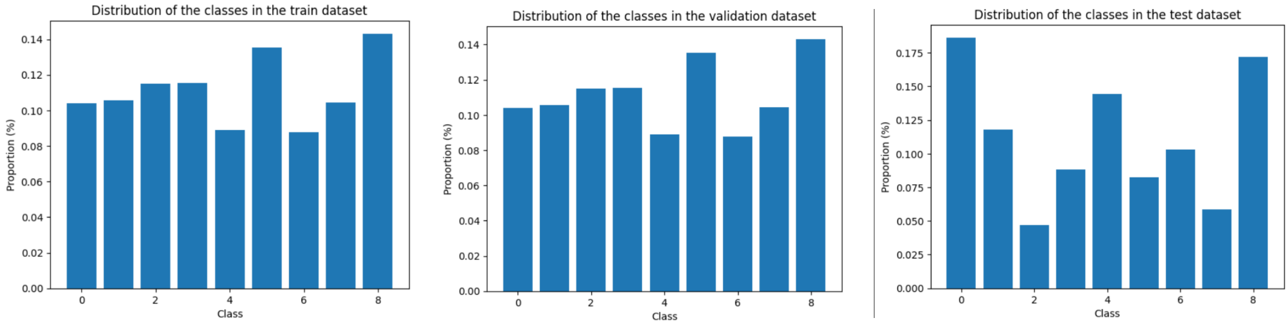


Figure 1: Example images for each of the nine tissue classes represented in the NCT-CRC-HE-100K data set. ADI, adipose tissue; BACK, background; CRC, colorectal cancer; DEB, debris; HE, hematoxylin–eosin; LYM, lymphocytes; MUC, mucus; MUS, smooth muscle; NCT, National Center for Tumor Diseases; NORM, normal colon mucosa; STR, cancer-associated stroma; TUM, colorectal adenocarcinoma epithelium [5].

Since three different sets are used, one may wonder whether there is a class imbalance. With a histogram over the different sets, one can conclude that the class imbalance is present in the way that each label doesn't have the same number of images compared with the other labels. Besides, since the training and validation sets have been split from the same original set (NCT-CRC-HE-100K), it can be seen that between these two sets, the distribution of images into the labels has been kept (Fig 2). Hence, the data set has been stratified over the so-called development set. Nevertheless it is emphasized that there is a class imbalance between the development set and the test set. This can lead to a larger gap between the accuracy of the model in the validation set and the accuracy on the test set.



(a)

(b)

(c)

Figure 2: Distribution of the images over the 9 different labels in the : train set (a), validation set (b), test set (c)

Data preprocessing

As images included in the sets come from MNIST, the preprocessing has already been done, the images have the same dimensions, with normalized colors. Besides, when the data from PathMNIST are downloaded, as Pytorch and its dependencies are used in the algorithm, it is required to transform the representation of the data, from lists to tensors. It enables to run the program faster because Pytorch is designed to work with tensors. Moreover, a common process in classification algorithms, especially when the data are images, is to normalize the data with a mean of 0.5 and a standard deviation also of 0.5. Most of all, each datasets has been split in batches of length 256 images, it is also a common technique to run the algorithm faster.

Avoiding data leakage

Working on Google Colab enables to run the different parts of the code in order. Each cell is executed after the previous ones have been finished to be computed. Then in a first part, using the training set and the validation set was necessary to find the hyperparameters in the code, namely the number of epochs and the learning rate value. In a second part, after the selection of the parameters with the best AUC obtained, the model was run on the test set to quantify the real performance of the model elaborated.

Since it is said that the PathMNIST dataset is composed of non-overlapping patches, since the dataset is divided in three subsets with non-overlapping patches, and that one set (test set) haven't been used to conceive the CNN model, but only to evaluate it at the very end of the program, then data leakage is avoided.

Machine learning pipeline

0.3 Model used

We chose to implement a CNN model to classify images into nine labels. To do so, the model is composed of five layers, the first fourth sequentially compute a 2D convolution, then normalize the values and finish by adding non-linearity. The fifth layer uses a 2D convolution and linearizes the vector into 9 values, what is equal to the number of labels. Thus, the vector represents the probabilities for a given image, to belong to each class. The attribution of a class is done further with the softmax function, which erases all the statistic values in the vector, and it replaces the maximum value with the unit value.
Then the use of the cross-entropy between the predicted class and the real label of images, for the computation of the loss, was obvious to feedback the model and optimize the parameters.

0.4 Validation technique

Since the datasets have been defined and are available directly from PathMNIST, the use of a nested hold-out technique appears to be not only the simplest option, but also the better. In fact, the amount of images provided by the PathMNIST is huge, and implementing techniques like cross-validation or nested cross-validation would have taken lot of computing time.

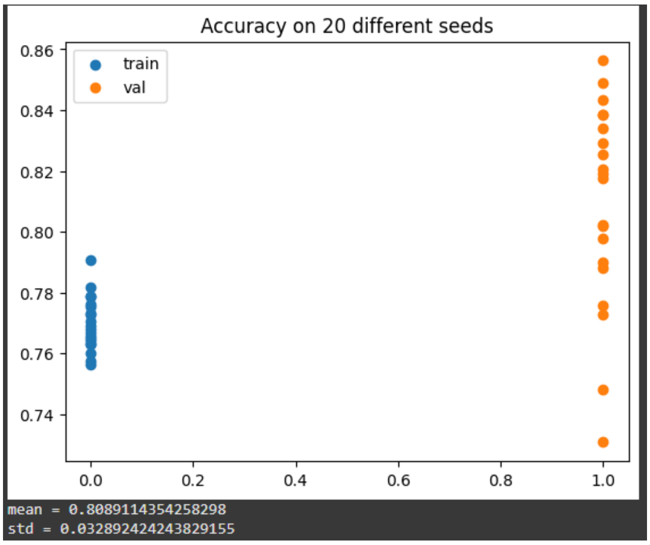
The area under the ROC curve (AUC) was used to evaluate the performance of the classifier. It was chosen because in the same time it avoids any dependency of the imbalanced distribution of the data on the result, which is not usually the case for the accuracy for instance.

It must be said that attention was also focused on the separation of the training values and validation values during the training process since the validation set has to be used as an evaluation of what the model already learned on the train set. That is why the calculation of the loss can be computed also on the validation set, but the gradient descend technique to reduce further its value, can't be used anymore.

0.5 Hyperparameters

To choose the hyperparameters, the development set solely was used. Two hyperparameters were highlighted, namely the number of epochs and the learning rate (lr).
First of all, as we have several batches, to be sure to have a robust and reproducible computation, we train the model over the development set, but with different shuffle seeds. In fact, the seed used seems to modify effectively

the value of the accuracy. An experiment was done over a 1-epoch test (Fig 3). Since the seed impacts the results, the same given seed is fixed to any random source in the code.



```
mean = 0.8089114354258298
std = 0.032892424243829155
```

Figure 3: Impact of the randomization of the batches during the evaluation of the efficiency of the CNN model

0.6 Number of epochs

An epoch is a one-shot run of the model over a set of data (here both the training and the validation sets). Running the algorithm over several epochs enables to increase the evaluation value because the model doesn't have to start from the scratch at each iteration, and can refine the previous parameters already found. Indeed, the model is composed of layers, each layer is divided into several functions, which are characterized by weights. These weights come from computation of the previous epochs, they are then updated in the actual epoch.
Focusing on the number of epochs, the decision was made to limit it to five, because the computational time increases very fast as there are also different seeds taken into consideration. Hence, the algorithm computes a new model for each seed (start from scratch to avoid influence of the previous model on the new one), where it runs a CNN model over all batches (100 000 images in total), for each epoch.
Fixing lr=0.001 and three different seeds, the result in figure 4 shows that the AUC raises over the five epochs with the training set. In addition, the AUC average raises over epochs for the validation set. It also means that the model is not overfitting on the train set values.

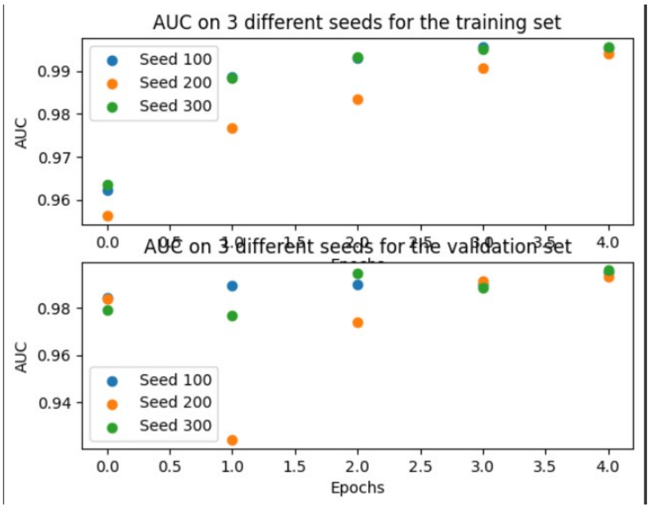
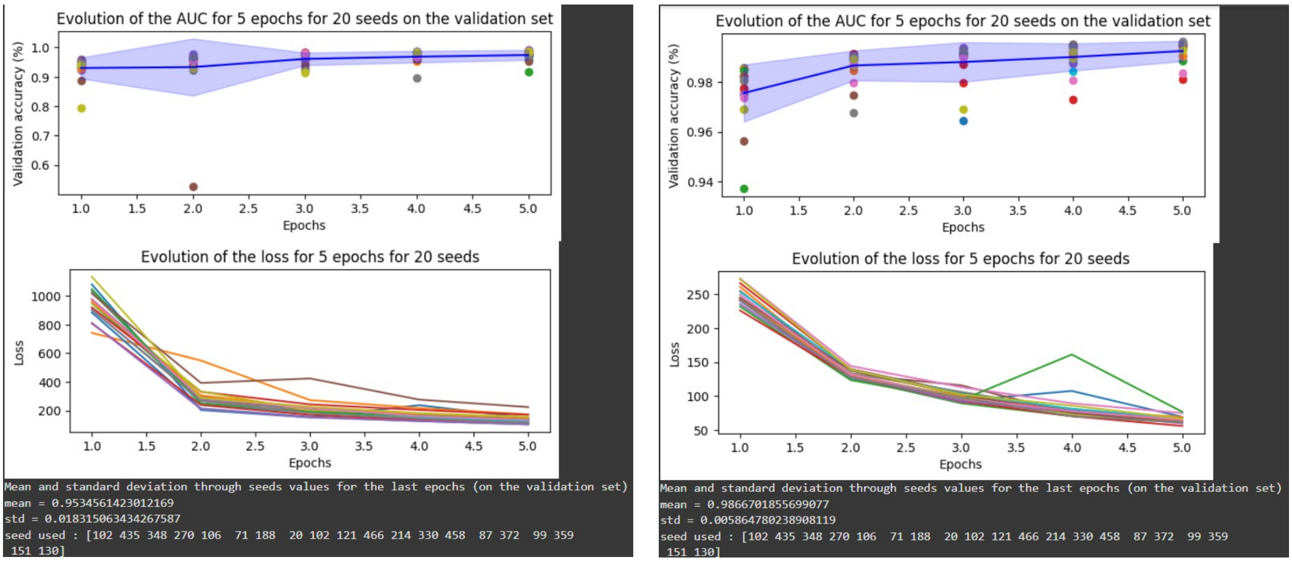


Figure 4: Impact of the number of epochs on the accuracy over the training (top) and validation (bottom) sets

0.7 Learning rate

The learning rate is a parameter which characterizes the velocity in the modification of the weights in the CNN model. Then a high value enables each actual batch to impact more heavily the parameters of the model. Usually, a trade off need to be found as a high value of the lr doesn't suffice to train the model efficiently, and a small value of the lr can lead to overfit the values of the train set.

Three learning rate (0.1 ; 0.01 ; 0.001) were used, the algorithm was computed with a number of five epochs and twenty seeds chosen randomly. Thus, a first try was done with lr=0.1, which leads to a randomized classifier (AUC around 0.5).
On second try, fixing lr=0.01, the classifier was not random anymore and reached a mean accuracy of 0.95 with a standard deviation of 0.02. These values were further refined with a lower learning rate fixed at 0.001 (Fig 5). In these graphs, the full line represents the mean AUC value from 20 AUC values, one for each different seed.



```
Mean and standard deviation through seeds values for the last epochs (on the validation set)
mean = 0.9534561423012169
std = 0.018315063434267587
seed used : [102 435 348 270 106  71 188  20 102 121 466 214 330 458  87 372  99 359
 151 130]
```

(a)

```
Mean and standard deviation through seeds values for the last epochs (on the validation set)
mean = 0.9866701855699077
std = 0.005864780238908119
seed used : [102 435 348 270 106  71 188  20 102 121 466 214 330 458  87 372  99 359
 151 130]
```

(b)

Figure 5: efficiency of the model over 5 epochs and 20 seeds for a learning rate equal to 0.01 (a) and 0.001 (b)

What is also shown in the Fig(5) is the convergence of the loss value through the number of epochs. It then completes the previous presentation of the impact of the learning rate since the variation of the loss is more smooth with a smaller learning rate. What can also be added is that the final value of the loss, at the end of the computation, is lower. Indeed for lr=0.01, the maximum value of the loss is near 200 whereas for lr=0.001, the loss value is less than 100. It is then coherent that the AUC of the model is higher with a smaller learning rate.

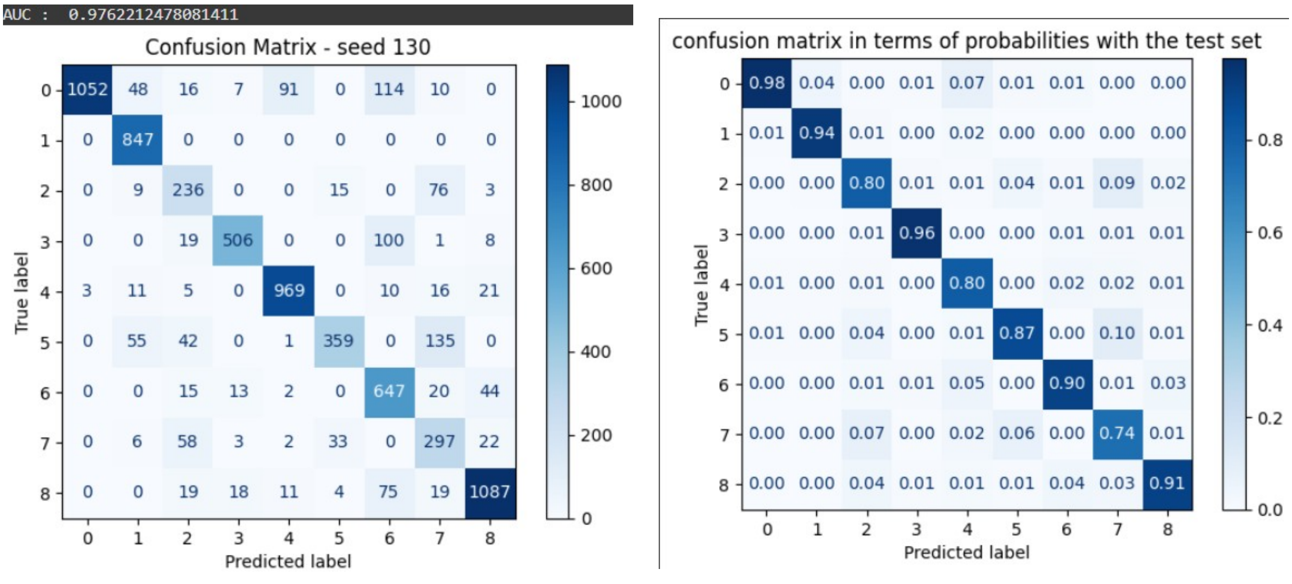
To conclude on the choice of the hyperparameters, we decided to fix the number of epochs to five, and to fix the learning rate of the model to lr=0.001.

Results

As long as the parameters of the model are fixed to optimize the performances of the model on the validation set, we obtain in the end an AUC=0.98 on average, over a repetition of 20 different random seeds, with a standard deviation around 6.10^{-3}. The value of the standard deviation leads to conclude that the algorithm is reproducible, and the high AUC value lets think that the classifier performs very well.

Then it is possible to use the CNN model on the test set, since all the parameters have already been fixed. The AUC obtained is a little bit less than the one returned with the development set, indeed it reaches 0.97. This was expected because the model was fitted to run on the development set. Besides, this value still be high, and the fact that the evaluation value on the development set and the evaluation value on the test set are close, means that overfitting the development set has been avoided.

Notwithstanding the area under the ROC curve is a good indicator of the performance, it doesn't suffice to point out the disparities of miss-classification within the labels. Hence it is useful to look at the confusion matrix, which shows the classification made by the model regarding the real values.
The classical confusion matrix publishes the number of images taken from a label, and classified by the model. Since there are disparities over the number of images available between the labels, it is also useful to show classification in terms of percentage (Fig 6)



```
AUC :  0.9762212478081411
```

Confusion Matrix - seed 130

True label \ Predicted label	0	1	2	3	4	5	6	7	8
0	1052	48	16	7	91	0	114	10	0
1	0	847	0	0	0	0	0	0	0
2	0	9	236	0	0	15	0	76	3
3	0	0	19	506	0	0	100	1	8
4	3	11	5	0	969	0	10	16	21
5	0	55	42	0	1	359	0	135	0
6	0	0	15	13	2	0	647	20	44
7	0	6	58	3	2	33	0	297	22
8	0	0	19	18	11	4	75	19	1087

(a)

confusion matrix in terms of probabilities with the test set

True label \ Predicted label	0	1	2	3	4	5	6	7	8
0	0.98	0.04	0.00	0.01	0.07	0.01	0.01	0.00	0.00
1	0.01	0.94	0.01	0.00	0.02	0.00	0.00	0.00	0.00
2	0.00	0.00	0.80	0.01	0.01	0.04	0.01	0.09	0.02
3	0.00	0.00	0.01	0.96	0.00	0.00	0.01	0.01	0.01
4	0.01	0.00	0.01	0.00	0.80	0.00	0.02	0.02	0.01
5	0.01	0.00	0.04	0.00	0.01	0.87	0.00	0.10	0.01
6	0.00	0.00	0.01	0.01	0.05	0.00	0.90	0.01	0.03
7	0.00	0.00	0.07	0.00	0.02	0.06	0.00	0.74	0.01
8	0.00	0.00	0.04	0.01	0.01	0.01	0.04	0.03	0.91

(b)

Figure 6: Confusion matrix in term of number of images(a) and in term of percentage (b)

Following the confusion matrix, one can conclude that there are disparities looking at the percentage of good classification in labels made by the model. What is particularly interesting in this analysis is to succeed in recognizing the images labeled with cancer-associated stroma (label n°7), which is unfortunately the lower performance classification over the dataset with only 74 % of good images classified among the total amount of images labeled 7.

As a comparison, training a RandomForest classifier with the test set, and evaluating its performance with the AUC, leads to a result of 0.84, which is under the 0.97 obtained with the developed CNN model. However, it is important to highlight that the validation set is not taken into account in this classification.

Discussion

With the previous work, an effort was developed to realize a CNN model to classify histological images into nine labels. Decisions were taken on the hyperparameters, taking also into consideration the computing time. One can assume that with an increase of the number of epochs and with the reduction of the learning rate, the performances of the model would be better. All the more, the model himself is limited with five layers, an augmentation of the number of layers is linked with an increase of the number of parameters, then with a raise in the computational time but also in the complexity of the model.
Most of all, increasing the complexity or trying to go further in the increase of the performances with the development set can lead to an overfitting. In this program, it is clear that overfitting have been avoided, since the results with the test set is close to the results with the development set, then the model succeeds in generalization.

Another point is that we tried to improve the computing time, using Pytorch and tensors with a GPU rather than a CPU. Indeed, it was needed to reduce this running time since a free version of Google Colab enables to use a GPU only on a couple of hours a day. Nevertheless, even if everything under our knowledge was used to perform the program, the algorithm still be optimizable.

Conclusion and future work

To conclude, we realized a CNN model which permits to assess an AUC of 0.97 over the classification of histological images from the PathMNIST set into the nine labels under consideration. However, this general AUC hides disparities in misclassification of images into the different labels. Focusing on the cancer-associated stroma, it is shown that the AUC is less than the general one. In a future work, one can take into account another dataset as big as the one used in this study, collecting only images labeled cancerous and non-cancerous with an approximately equal distribution to make the model focus on the cancer stroma-associated features.

Ethic and data privacy

0.8 Ethic

All experiments were conducted in accordance with the Declaration of Helsinki, the International Ethical Guidelines for Biomedical Research Involving Human Subjects by the Council for International Organizations of Medical Sciences (CIOMS), the Belmont Report, and the US Common Rule. Anonymized archival tissue samples were retrieved from the tissue bank of the National Center for Tumor diseases (NCT; Heidelberg, Germany) in accordance with the regulations of the tissue bank and the approval of the ethics committee of Heidelberg University (tissue bank decision numbers 2152 and 2154, granted to NH and JNK; informed consent was obtained from all patients as part of the NCT tissue bank protocol; ethics board approval S-207/2005, renewed on 20 December 2017). Parts of these samples originated from the DACHS study [31,32]. Another set of tissue samples was provided by the pathology archive at University Medical Center Mannheim (UMM; Heidelberg University, Mannheim, Germany) after approval by the institutional ethics board (Ethics Board II at UMM; decision number 2017-806R-MA, granted to AM and waiving the need for informed consent for this retrospective and fully anonymized analysis of archival samples). HE images from the The Cancer Genome Atlas (TCGA) [33] were downloaded from public repositories at the National Institutes of Health (NIH; USA). These images were randomly drawn from colorectal adenocarcinoma (COAD) and rectal adenocarcinoma (READ) patients.

0.9 Privacy

The MedMNIST dataset is licensed under Creative Commons Attribution 4.0 International (CC BY 4.0), except DermaMNIST under Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).

Code and reproducibility

Source code findable : https://github.com/clementklein/Project-AI-Medicine

Versions of the libraries : medmnist 3.0.2 matplotlib 3.10.0 numpy 2.0.2 sklearn 1.6.1 torch 2.6.0+cu124

References

[1] Cercek A Smith RA Jemal A. Siegel RL, Wagle NS. Colorectal cancer statistics, 2023. *CA Cancer J Clin.*, 73(3), 2023, May-Jun.

[2] Thaína A. Azevedo Tosta, Paulo Rogério de Faria, Leandro Alves Neves, and Marcelo Zanchetta do Nascimento. Computational normalization of he-stained histological images: Progress, challenges and future potential. *Artificial Intelligence in Medicine*, 95:118–132, 2019.

[3] Marc Macenko, Marc Niethammer, J. S. Marron, David Borland, John T. Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E. Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, 2009.

[4] Marks A Kather J.N, Halama N. 100,000 histological images of human colorectal cancer and healthy tissue. *Zenodo*, April 7, 2018.

[5] Pornpimol Charoentong Tom Luedde Esther Herpel Cleo-Aron Weis Timo Gaiser Alexander Marx Nektarios A. Valous Dyke Ferber Lina Jansen Constantino Carlos Reyes-Aldasoro Inka Zörnig Dirk Jäger Hermann Brenner Jenny Chang-Claude Michael Hoffmeister Niels Halama Jakob Nikolas Kather, Johannes Krisam. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *Plos medicine*, 16(1): e1002730, January 24, 2019.