Project Report - Histopathology OOD classification

Garance Gérard GARANCE.GERARD@ELEVES.ENPC.FR
Audrey Airaud AUDREY.AIRAUD@POLYTECHNIQUE.EDU
Group Number: 35 – Team Name: CPC

1. Introduction

This project tackles binary classification of histopathology patches from multiple hospitals, each with different staining protocols, equipment, and imaging conditions. These variations introduce distribution shifts that can affect the model's ability to generalize. Since the model must perform well on unseen hospitals, we must design models that are robust to such shifts and capable of performing reliably on data from unseen hospitals.

2. Architecture and methodological components

We began by exploring the baseline model provided for the challenge, which uses DINOv2 (Oquab et al., 2023) as a feature extractor followed by a single linear classification layer. No specific preprocessing was applied to the data beyond reshaping the images to match the model's input requirements. This baseline setup achieved an initial accuracy of 0.9060 on the Kaggle leaderboard. The use of DINOv2 was predefined in the project, as it uses self-supervised learning to extract strong features from images without requiring labels. This makes it useful in our task, where the high variability in tissue images require strong visual encodings to enable accurate classification. We explored two key directions to improve upon the baseline:

1. Replacing the Linear Head Instead of using a simple linear layer after the DINOv2, we introduced a multi-layer perceptron to model non-linear decision boundaries. This included: two linear layers, one layer with ReLU activations, one Dropout layer (0.2) that helps prevent overfitting. This new structure, with the MLP, improves the model's ability to process the features extracted by DINOv2 and adapt more effectively to the distribution shift across different centers.

2. Testing alternatives to DINOv2

• **Comparative Baseline with ResNet** As a benchmark, we also ran experiments with ResNet50 (He et al., 2016), using its penultimate layer as the feature extractor. This allowed us to evaluate whether DINOv2 truly offered superior representations in our context. To ensure consistency with the pre-training conditions of ResNet50, which was trained on the ImageNet dataset using a specific normalization, we applied the same normalization parameters to our dataset. Since the results were not as good as those obtained with DINOv2, it confirmed that DINOv2 was better suited for this task.

• **Replacing DINOv2 with Phikon (Chen et al., 2022)** Phikon is a self-supervised learning model for histopathology images, trained using iBOT by Owkin and made available through Hugging Face. We experimented with replacing DINOv2 by Phikon. Given that Phikon was already trained on histopathology images, we applied minimal preprocessing. With Phikon, we achieved a Kaggle score of 0.97493. However, we realized that we have no guarantee that Phikon was not trained on the same images as those in our dataset. If the model had indeed seen test images during pretraining, this would pose a serious data leakage issue and could explain the high performance.

• **Replacing DINOv2 with UNI2-h (Chen et al., 2024)** UNI2-h is a foundation model for digital pathology, trained on a vast collection of histopathology slides. With a simple linear classifier on top and minimal preprocessing, we achieved a Kaggle score of 0.98258. We then fine-tuned the hyperparameters, which led to our best overall performance on the Kaggle test set : 0.98872% of accuracy. However, as with Phikon, we cannot exclude the possibility that some of our test images were seen during the pretraining of UNI2-h, which raises concerns about potential data leakage and its influence on the final score.

3. Model tuning and comparison

Preliminary Dataset Analysis Our first objective was to verify that the distribution of positive and negative labels was approximately balanced, as the evaluation metric used in this task is accuracy, which can be misleading on imbalanced datasets. The training dataset includes data from three different hospitals, while both the validation and test sets contain data from a single hospital each. Therefore, it was crucial to assess the balance of the label distribution by hospital.

	Training Set			**Validation Set**
	Center 0	**Center 3**	**Center 4**	
Label 0	8 815	19 293	21 672	17 452
Label 1	8 941	19 463	21 816	17 452
Total	17 756	38 756	43 488	34 904

Table 1: Label distribution per center in the training set and in the validation set.

Once we confirmed the dataset was well balanced, as shown in Table 1, we were able to move on to other preprocessing challenges. We then investigated the variability in the dataset: differences in color tones, image intensity, zoom levels, and even slight rotations. These inconsistencies are likely caused by variations in imaging machines and protocols across hospitals. As said earlier, this variability can make it harder for the model to generalize, as it might learn to focus on visual differences that are not medically important.

Preprocessing Methods To address the heterogeneity in the dataset, particularly due to acquisition differences between hospitals, we applied two main types of preprocessing: data augmentation and normalization. These techniques aimed to improve the model's generalization by reducing the visual variability unrelated to the underlying pathology.
As shown in Figure 1, there are notable differences in both the distribution and intensity of colors across hospitals. For example, Center 4 exhibits more saturated and intense colors,

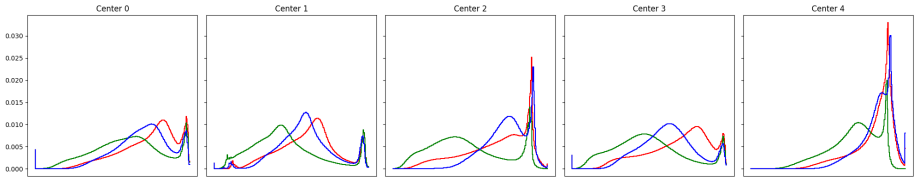

Figure 1: Comparison of RGB intensity histograms across centers.

which can be seen in the RGB intensity curves. These variations likely stem from differences in staining protocols and imaging devices.

1. Data Augmentation To help the model generalize across centers, we incorporated the following augmentations during training. These augmentations were used with the DINOv2 model to introduce variability and improve robustness. However, we stopped their use with Phikon and UNI2-h, as these models were already pre-trained on histopathology images and performed better with minimal preprocessing.

• **Horizontal and Vertical Flips:** These transformations simulate variations in tissue orientation. Since biological samples lack a fixed directionality, they help the model focus on morphological features rather than spatial positioning. In our experiments, horizontal flipping consistently improved performance, whereas vertical flipping had no impact. As a result, we retained only the horizontal flip in our augmentation pipeline.

• **Color Jittering:** We initially used Color Jittering to introduce variability in brightness, contrast, saturation, and hue. However, we observed that it introduced arbitrary changes in color that were not always biologically plausible. In histopathology, preserving the visual integrity of tissue structures is crucial. This led us to explore more domain-specific augmentations.

• **HED-Based Augmentations:** We therefore shifted to using HED-based transformations (`hedlightcolor`, `hedlightercolor`, `hedstrongcolor`), which simulate more realistic stain variations commonly observed in histological slides. These methods proved more effective at enhancing generalization and yielded better validation performance.

2. Normalization To further reduce color variability between patches—and thus minimize the need for aggressive color-based data augmentation—we applied two types of normalization:

• **Standard Normalization:** Since models like DINOv2, Phikon etc were pre-trained on images with specific mean and standard deviation values, we applied the same normalization to our datasets. This approach proved effective in stabilizing training and improving performance by aligning our data distribution with that of the pretraining dataset.

• **Stain Normalization (Reinhard Method (Reinhard et al., 2001)):** Given the significant differences in staining across centers, this method was particularly valuable. It standardizes the color appearance of histopathology slides by aligning each image's stain distribution to a common reference. This reduces variability introduced by different staining

protocols and helps the model focus on relevant morphological patterns rather than superficial color differences. Figure 2 shows the impact of stain normalization on the dataset.

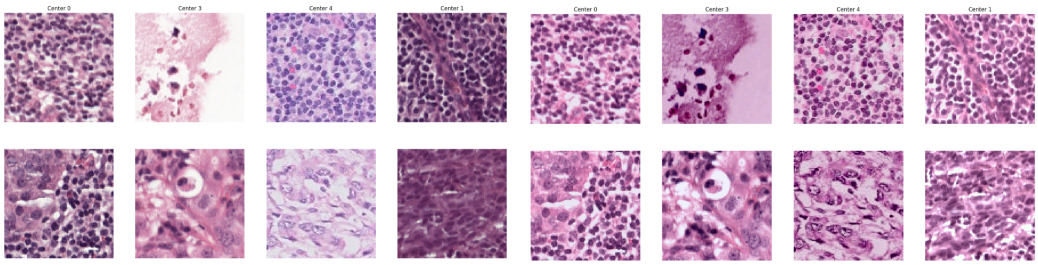

Figure 2: Example images from four different centers: original (left) and after stain normalization (right).

Fine-Tuning the Backbone : Initially, the backbone was frozen, and only the classifier was trained. We later experimented with unfreezing DINOv2, allowing gradient updates on the transformer weights to adapt the representations to our dataset. We tested two different approaches: first, training only the classifier on top of DINOv2 to allow for slight adaptation, followed by fine-tuning the entire model, including both DINOv2 and the linear layer. The second approach consisted in directly fine-tuning the whole model from the start. The main challenge was to prevent overfitting, which required limiting the number of training epochs. This method, combined with relevant preprocessing, yielded good results, achieving an accuracy of around 0.94. However, with Phikon and UNI2-h, we did not finetune the Backbone since it was already trained on histopatholy datasets.

Hyperparameter Optimization: We conducted a grid search to identify the best hyperparameters for our model. Several optimizers were evaluated, including *Adam*, *AdamW*, and *SGD*. The batch size was varied from 4 to 1024, and the learning rate was explored within the range of 10^{-2} to 10^{-4}. We noticed that a smaller batch size enabled a better generalization. For the best model with UNI2-h, the optimal hyperparameters were as follows: the *SGD* optimizer, a learning rate of 2×10^{-4}, a batch size of 8, and 10 epochs. We retained the default values for momentum and weight decay.

Validation Procedures Given that the training set consists of data from three different centers and the validation set comes from a fourth, our first step was to verify whether training the model on data from each center separately would lead to similar performance levels. This analysis was essential to ensure that the model does not become overly dependent on center-specific characteristics, which could harm its generalization ability. The results showed comparable performance across centers, suggesting no major bias or center-specific difficulty in the training data. Moreover, we performed a train-validation split using data from the same centers to ensure that the model was capable of learning and generalizing not only on data from different centers, but also on data with the same distribution (see Table 3.

Recap of Performances Table 2 presents a comparison of the scores obtained on the Kaggle test set for each model, along with a brief explanation of the preprocessing choices.

Model	Backbone	Classifier	Fine-tuning Backbone	Preprocessing	Accuracy
Baseline	DINOv2	Linear	✗		0.9060%
	DINOv2	MLP	✗	Random Flips	0.93130%
	DINOv2	MLP	✓	Random Flips + Stain Normalization	0.95086%
	DINOv2	Linear	✓	Standard Normalization	0.96870%
	Phikon	Linear	✗	Random Flips + Standard Normalization	0.97493%
Best Model	UNI2-h	Linear	✗	UNI2-h transform	**0.98872%**

Table 2: Comparison of model architectures using different backbones and training strategies.

Model	Validation id	Validation ood	Test ood
UNI2-h Best Model	0.9930%	0.9853%	0.98872%

Table 3: Accuracies over the different validation and test datasets for the UNI2-h model

Fine-tuning methods that we tried but that we discarded due to limited performance

- To reduce the domain shift between the train and validation/test set, we had the idea to add a **CORAL loss** (Sun and Saenko, 2016) term to the classification loss when fine-tuning the backbone. This domain adaptation technique encourages the model to produce domain-invariant features. We used the Skada (Redko et al., 2022) library which provides a ready-to-use implementation. However, due to unsatisfactory results on the validation set, we chose not to continue with this method.

- Fully retraining DINOv2 is computationally expensive and likely to overfit due to limited data. **LoRA (Low-Rank Adaptation)** (Hu et al., 2022) offers an alternative by allowing us to fine-tune only the most impactful parts of the model. However, existing LoRA implementations are not directly compatible with DINOv2. We also tested LoRA on a ResNet backbone, which is more compatible with existing LoRA tools. However, the performance was less satisfying than our other methods.

- We also had the idea of using a **Masked Autoencoder (MAE)** to fine-tune DINOv2. The idea is to randomly mask a large portion of each image patch (75%) and train the model to reconstruct the missing pixels. With this, DINOv2 has to focus on capturing the global structure of the tissue rather than low-level information, which may vary across hospitals. However, this approach increased a lot training time with little performance improvement.

4. Conclusion

This challenge highlights the superiority of large, pre-trained foundation models such as DINOv2, UNI, and Phikon. We also observed that foundation models specifically pre-trained for pathology images consistently outperformed others. Despite the sophistication of techniques like CORAL, these models, trained on massive and diverse datasets, exhibit strong generalization across unseen domains, such as new hospitals. While their baseline performance is already impressive, future work could investigate whether combining them with lightweight adaptation strategies can push performance even further.

References

Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30(3): 850–862, 2024.

Ruizhe Chen, Yutong Chen, Chao Lu, Ruibin Zhang, Zhen Zeng, and Shijian Wang. Phikon: Towards accurate histopathology image classification using patch-level vision-language pretraining. In *European Conference on Computer Vision*, pages 241–258. Springer, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Maxime Oquab, Théophile Darcet, Theo Moutakanni, Huy Vo, Marc Szafraniec, Yannis Kalantidis, Julien Mairal, Natalia Neverova, Herve Jegou, and Patrick Labatut. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Skada: A python toolbox for supervised and unsupervised domain adaptation on classification tasks. https://github.com/scikit-adaptation/skada, 2022.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.