Kaggle Challenge: DL for Medical Imaging

Paul Caucheteux* PAUL.CAUCHETEUX@ENS-PARIS-SACLAY.FR

Antoine Mirri† ANTOINE.MIRRI@DAUPHINE.EU

1. Introduction

Histopathology involves the microscopic examination of tissue to detect disease, such as cancer. In this challenge, we are tasked with classifying whether a histopathological image patch contains tumor tissue (label 1) or not (label 0). Each image corresponds to a small patch of tissue extracted from a whole slide image. While the task is conceptually a binary classification problem, one of the main challenges lies in the domain variability introduced by the origin of the images.

The dataset is collected from five different medical centers. The training set includes 100,000 images from three centers, the validation set consists of 34,904 images from a fourth unseen center, and the final test set contains 84,724 images from a fifth center, for which the labels are hidden. This setup introduces a significant domain shift due to differences in staining protocols, slide preparation, and scanner types across centers. Therefore, building a robust classifier requires more than just a performant architecture, it must also generalize well across domains. Some samples for each center are exposed in the appendix 1.

To address this, we explored both non-learning-based stain normalization techniques, such as Macenko normalization (Macenko et al., 2009), which rely on color statistics of the tissue, and learning-based methods, including CycleGANs (Zhu et al., 2020), to align color distributions between centers.

Once stain normalization was applied, we focused on selecting suitable architectures for feature extraction. We experimented with several vision foundation models, including DINOv2 (Oquab et al., 2024) and MedImageInsight (Codella et al., 2024). These features were subsequently fed into a classifier for final prediction.

Our pipeline achieved a final leaderboard accuracy score of **0.98056**, ranking 11st out of 48 teams.

In the following, we first present the methodological components and architectural choices we explored (2), then describe our strategy to address domain shift adaptation (3), before detailing our experiments, tuning strategies, and comparative evaluations (4).

2. Section 1: Architechture and methodological components

Our framework is built upon a simple yet effective pipeline: we extract high-level representations from images using a pre-trained foundation model, and then train a lightweight classifier on these embeddings using our labeled dataset.

* ENS Paris Saclay
† Université Paris Dauphine

2.1. Foundation Models

2.1.1. DinoV2

As a baseline, we used DINOv2 (Oquab et al., 2024), a widely adopted self-supervised Vision Transformer (ViT) model pre-trained on billions of images.

In our initial setup, DINOv2 was used in a frozen manner, and we extracted embeddings from its final layer. Despite being a general-purpose model not specifically trained on medical data, it performed reasonably well on our task (see 1).

2.1.2. MedImageInsight

We explored the use of MedImageInsight (Codella et al., 2024), a recently proposed foundation model specifically designed for the medical domain. MedImageInsight follows a "two-tower architecture" composed of a vision encoder and a text encoder, jointly trained using a contrastive loss called *UniCL* (Yang et al., 2022), inspired by CLIP. The image encoder is based on the DaViT (Dual Attention Vision Transformer) architecture.

In our use case, we only leverage the image encoder of MedImageInsight, without using the text encoder or report generation capabilities. The choice of this model was particularly motivated by its training setup: unlike generalist models such as DINOv2, MedImageInsight was trained on a large corpus of more than 3 million medical images, covering 14 different modalities (X-rays, histopathology, MRI, CT, etc.), making it well suited for our task.

More importantly, MedImageInsight was explicitly trained on the PatchCamelyon dataset, a histopathology dataset similar in nature to our challenge data. Using a model pre-trained on histopathology data increases the chances of obtaining domain-adapted representations that can generalize well to our own tumor classification task.

2.2. Classifier

Once embeddings were extracted from the foundation models, we trained a simple fully connected neural network as the final classifier. The architecture consists of four linear layers with decreasing dimensions (512, 256, 128, 1), each followed by batch normalization and ReLU activation, and interleaved with dropout layers to prevent overfitting. The final output layer uses a sigmoid activation function to predict the probability of the image containing tumor tissue.

2.3. Finetuning

In order to better adapt the DINOv2 model to our specific histopathology classification task, we opted to fine-tune the last layers of its ViT backbone.

To that end, we unfroze the last two transformer blocks of DINOv2 as well as the final normalization layer, and jointly trained them alongside our classifier in an end-to-end manner. We used a small learning rate and included weight decay, dropout and early stopping to improve generalization. The fine-tuning process was computationally expensive and significantly slower than training a standalone classifier on frozen embeddings.

By contrast, we chose not to fine-tune the MedImageInsight model. This decision was motivated by the fact that MedImageInsight was already pre-trained on histopathological

images — including the PatchCamelyon dataset, which is closely related to the challenge data.

3. Section 3/2: Domain shift Adaptation

The domain shift problem, already introduced in 1, refers to the distributional gap between training and testing data due to variations in staining protocols, scanners, or acquisition centers. To assess domain adaptation methods, we relied on a fixed baseline model composed of frozen DINOv2 embeddings and a trained classifier. All proposed domain shift methods are evaluated using this fixed model to isolate the effect of preprocessing alone (see 1 for results).

3.1. Non learning Method

Non-learning approaches offer a fast way to reduce domain shift, since they do not require additional training. Our first attempt was to convert all images to grayscale, under the assumption that removing color information would reduce inter-center variability. Surprisingly, this naive strategy slightly improved the baseline score from 0.90560 to 0.90911 on the platform. However, we suspect that it discards important diagnostic features embedded in color differences, even if it helps align distributions across centers.

We then explored more advanced color normalization techniques developed specifically for histopathology, notably the Macenko method (Macenko et al., 2009). This method operates in the optical density (OD) space, which models how stains absorb light. The idea and an exemple of such a transformation is described in A.2.

In our case, the objective was to transfer the color profile of all source images (training and validation) toward the test domain. The standard Macenko pipeline requires a single target image. Since test labels are unavailable, we considered two strategies: (1) pick a single representative test image, or (2) average stain vectors over multiple test samples. Due to implementation complexity, we opted for the first strategy and experimented with different target images.

Unfortunately, results were disappointing, with a drop in accuracy to 0.8978. We attribute this to the high sensitivity of the method to the chosen target image, especially in our patch-level setting. While stain normalization aligns color distributions, it may also introduce distortions or remove subtle features useful for classification.

Other non-learning stain normalization methods exist (Hoque et al., 2024), such as Reinhard or Vahadane, but were not tested due to time constraints.

3.2. Learning Method

While non-learning stain normalization techniques offer simplicity, they often lack robustness and adaptability when applied across multiple domains. To address these limitations, we turned to learning-based approaches using generative adversarial networks (GANs).

3.2.1. CycleGAN

CycleGAN is a widely used model for unpaired image-to-image translation (Zhu et al., 2020). It learns two mappings between a source and a target domain, enforcing a cycle-consistency loss to preserve structural content while altering image style.

However, this approach is limited to one-to-one domain mappings and therefore requires a separate model for each source-target pair. This increases computational cost and reduces training data per model. In our experiments, we trained one CycleGAN per source center (centers 0, 1, 3, and 4) and used them to translate images toward the target domain (center 2).

3.2.2. MultiStain-CycleGAN

To overcome the scalability limitations of standard CycleGAN, we adapted the **MultiStain-CycleGAN** framework introduced by Hetz et al. (Hetz et al., 2023). This method reformulates stain normalization as a many-to-one domain translation task. It main contribution is the introduction of an *intermediate grayscale domain*, which separates structure from color information.

Instead of learning a direct mapping from each source center to the target, the model learns to reconstruct colorized images from grayscale inputs augmented with diverse color shifts. This training setup enables the model to handle previously unseen domains without retraining. An overview of the framework can be found in Figure 12.

The primary advantage of MultiStain-CycleGAN is its ability to support **multi-domain adaptation** using a single unified model, significantly reducing the number of models to train and obtaining much more source images for training (all source centers combined).

We compared two domain adaptation strategies: (1) a *multi-domain approach*, using a single MultiStain-CycleGAN trained on images from all source centers, and (2) a *domain-specific approach*, using one standard CycleGAN per source center.

These models were used to normalize the training and validation images with respect to the style of the test domain. After normalization, the images were passed through the pipeline described in 2.

Examples of transformations for each center produced by both the MultiStain-CycleGAN and per-domain CycleGANs are shown in Appendix A.4. On these specific examples (with identical training settings), the MultiStain-CycleGAN appears to produce smoother and more consistent transformations. We discuss in the results section of 4 the short training time of these methods leading to cautious interpretations.

4. Section 2: Model Tuning and comparison

4.1. Ablation Study

To assess the contribution of each component in our pipeline, we conducted an ablation study using a fixed classifier architecture across all experiments. Our baseline consists of raw images (without any filtering or stain normalization) encoded using a frozen DINOv2 backbone, followed by the classifier.

We then progressively added or modified components mentioned earlier to evaluate their individual impact.

4.2. Preprocessing

4.2.1. Abberant Datas

After inspecting the training data, we observed that some patches exhibited clear artifacts—entire regions filled with black or white pixels, likely due to acquisition or patch extraction errors (see Figure 7). Notably, such issues were absent in the test set.

These anomalous images may lead the model to learn spurious features. To mitigate this, we removed any image with over 200 pure black pixels (RGB = 0), since clean patches rarely exceed 10. This filtering discarded 386 training and 87 validation samples.

4.2.2. Data Augmentation

To improve generalization and reduce overfitting, we applied geometric data augmentations during training: horizontal/vertical flips and small rotations (up to 15°). These preserve the color distribution and thus maintain the effect of stain normalization.

Color-based augmentations (e.g., `ColorJitter`) were avoided, as they could disrupt the normalized stain distribution achieved through methods like Macenko or CycleGAN.

4.3. Performance and Results

Accuracy scores reported in Table 1 correspond to models trained on the training set and evaluated on the validation set. Although we later retrained some models on the combined train/val data, this brought minimal improvement on the final test set. Thus, all results shown here reflect models trained exclusively on the training data for consistency.

Feature Extractor	**None**	**Grayscale**	**Macenko**	**CycleGAN**	**Multi-CycleGAN**
DINOv2 (frozen)	0.90560	**0.90911**	0.89878	X	0.90429
DINOv2 (fine-tuned)	X	X	X	X	**0.92043**
MedImageInsight	**0.98056**	X	X	0.97399	0.97620

Table 1: Results for combinations of stain normalization techniques and feature extractors.

While we aimed for a comprehensive ablation study, computational limits prevented full exploration of all combinations. For instance, training multiple CycleGANs—one per source center—was time-consuming (up to 10 hours per model for 10 epochs). Although MultiStain-CycleGANs reduced this cost with a shared generator, training still required 12 hours for 10 epochs.

Thus, we focused on the most promising configurations. We believe the lack of improvement from GAN-based adaptation is mainly due to insufficient training time, likely resulting in suboptimal source-to-target transformations. The stain normalization effect was probably incomplete.

In contrast, fine-tuning DINOv2 improved performance as expected, enabling the model to better adapt to histopathological data. Finally, MedImageInsight outperformed other feature extractors, consistent with its pretraining on similar histopathology datasets.

References

Noel C. F. Codella, Ying Jin, Shrey Jain, Yu Gu, Ho Hin Lee, Asma Ben Abacha, Alberto Santamaria-Pang, Will Guyman, Naiteek Sangani, Sheng Zhang, Hoifung Poon, Stephanie Hyland, Shruthi Bannur, Javier Alvarez-Valle, Xue Li, John Garrett, Alan McMillan, Gaurav Rajguru, Madhu Maddi, Nilesh Vijayrania, Rehaan Bhimai, Nick Mecklenburg, Rupal Jain, Daniel Holstein, Naveen Gaur, Vijay Aski, Jenq-Neng Hwang, Thomas Lin, Ivan Tarapov, Matthew Lungren, and Mu Wei. Medimageinsight: An open-source embedding model for general domain medical imaging, 2024. URL https://arxiv.org/abs/2410.06542.

Petra Hetz, Lukas Eder, Jens Kleesiek, and Julia Eggert. Multistain-cyclegan: Stain normalization for histopathological images using a single model trained on multiple domains. In *Medical Imaging with Deep Learning*, 2023. URL https://openreview.net/forum?id=pxLb3vt2uU.

Md. Ziaul Hoque, Anja Keskinarkaus, Pia Nyberg, and Tapio Seppänen. Stain normalization methods for histopathology image analysis: A comprehensive review and experimental comparison. *Information Fusion*, 102:101997, 2024. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2023.101997. URL https://www.sciencedirect.com/science/article/pii/S1566253523003135.

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. doi: 10.1109/ISBI.2009.5193250.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space, 2022. URL https://arxiv.org/abs/2204.03610.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020. URL https://arxiv.org/abs/1703.10593.

Appendix A. Illustrations

A.1. Samples from the initial dataset

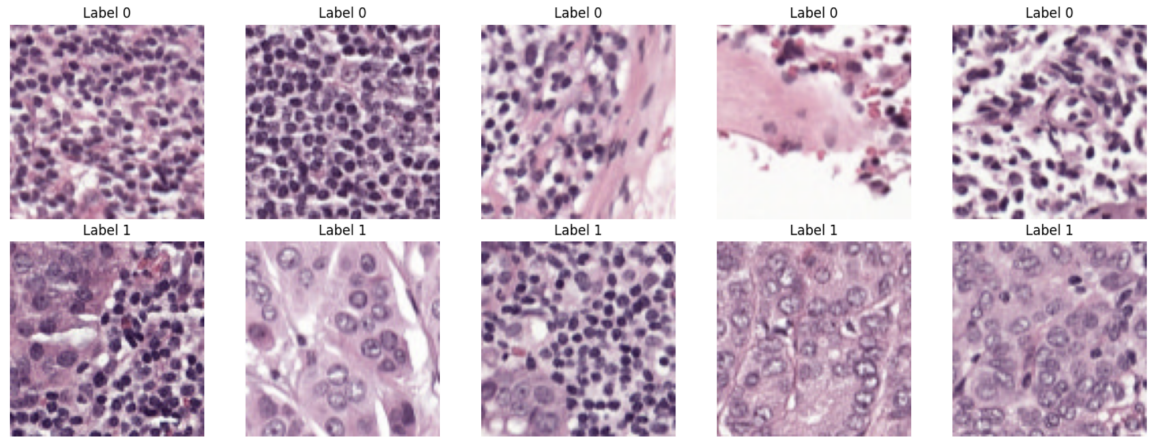

Figure 1: Samples from center 0 (Training set).

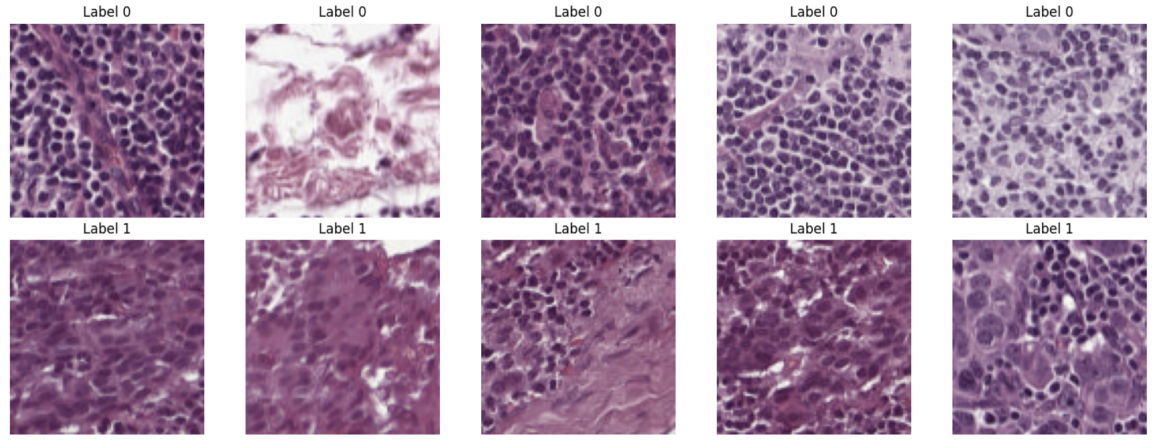

Figure 2: Samples from center 1 (Validation set).

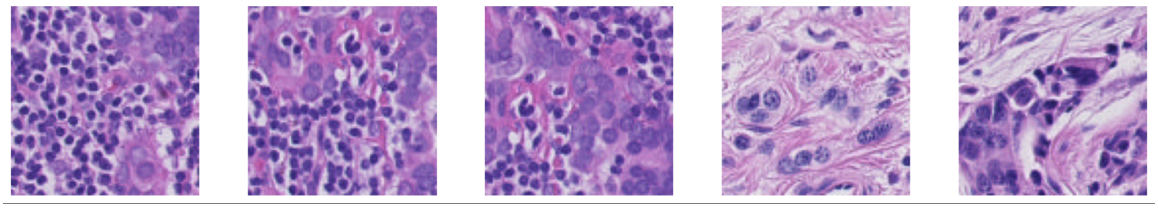

Figure 3: Samples from center 2 (Test set).

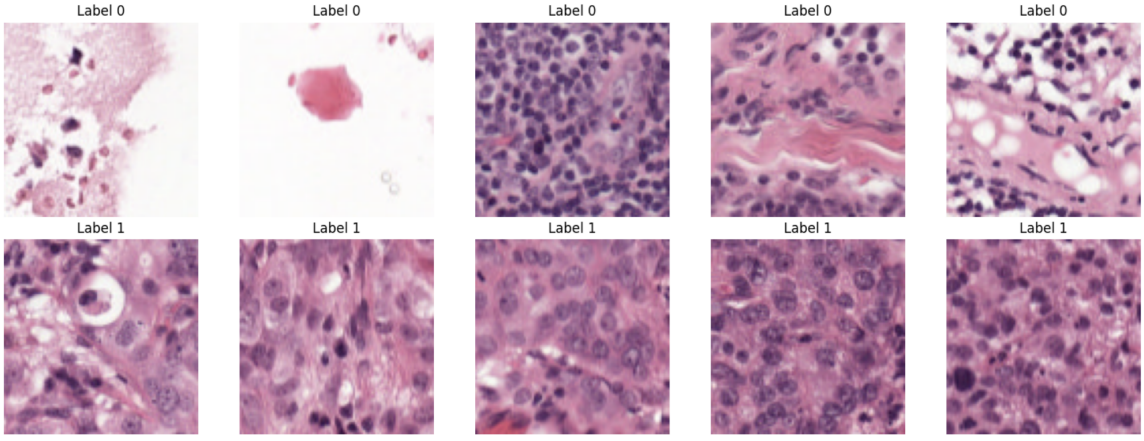

Figure 4: Samples from center 3 (Training set).

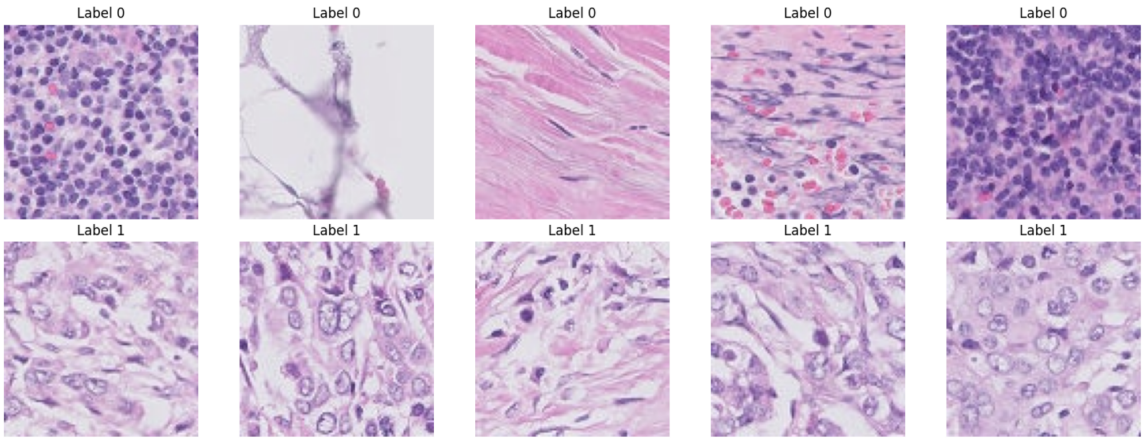

Figure 5: Samples from center 4 (Training set).

A.2. Macenko Transformation

The idea is to:

- First, both the source image I_s and the target image I_t are converted into the optical density (OD) space using the transformation: $OD = -\log\left(\frac{I+1}{255}\right)$.
- Low-density background pixels are removed, and a singular value decomposition (SVD) is performed on I_t in the OD space to extract two stain vectors, denoted as H_t.
- Each I_s is then projected onto its own stain vectors to estimate stain concentrations, denoted as C_s.
- The stain vectors of I_s are then replaced by those from I_t, producing the normalized OD: $OD_{\text{norm}} = H_t \cdot C_s$.
- Finally, the normalized image is reconstructed in RGB space via: $I_{\text{norm}} = 255 \cdot \exp(-OD_{\text{norm}})$.

An example of such a transformation is described in Figure 6.

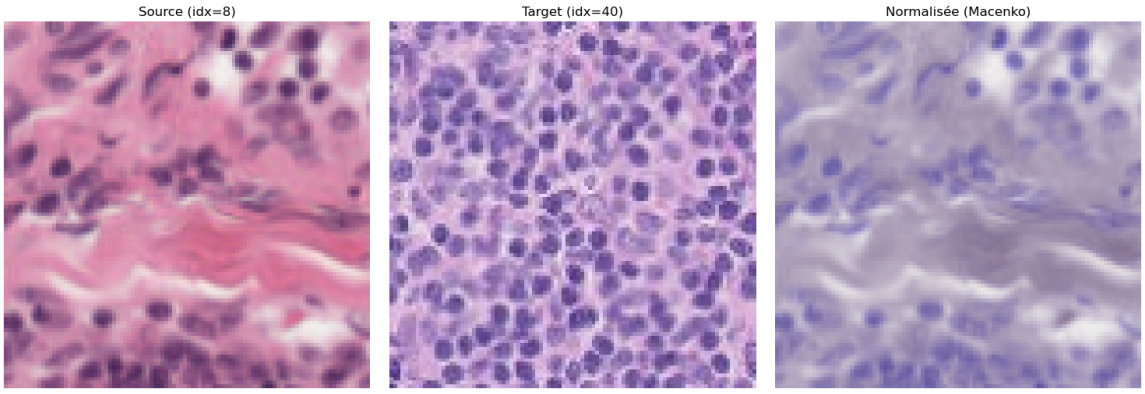

Figure 6: Example of a transformation using Macenko.

A.3. Aberrant data

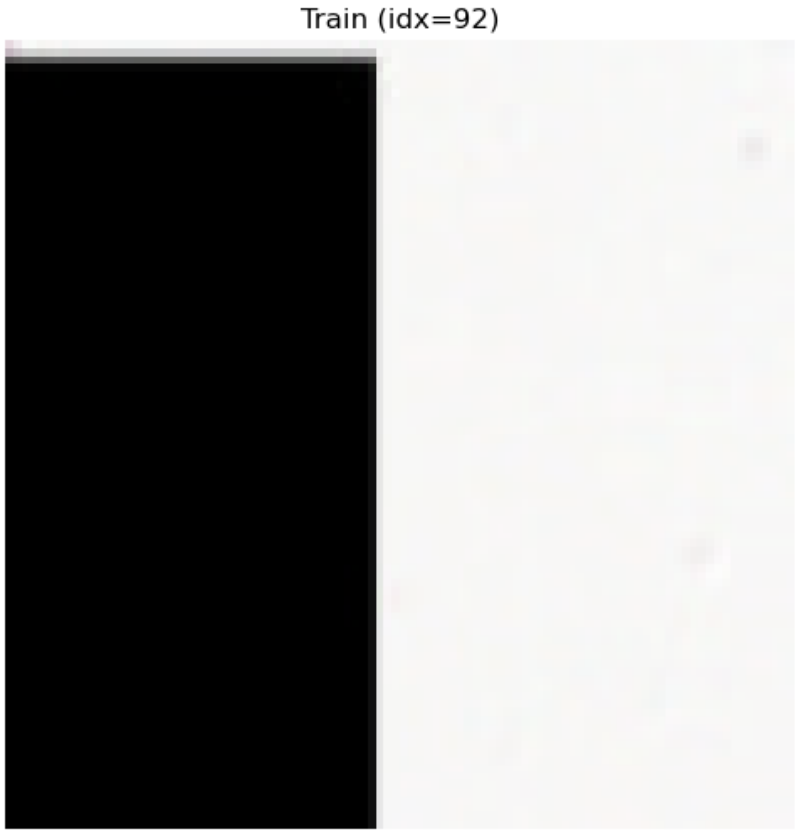

Figure 7: Example of an abberant image in the train dataset.

A.4. Cycle GAN

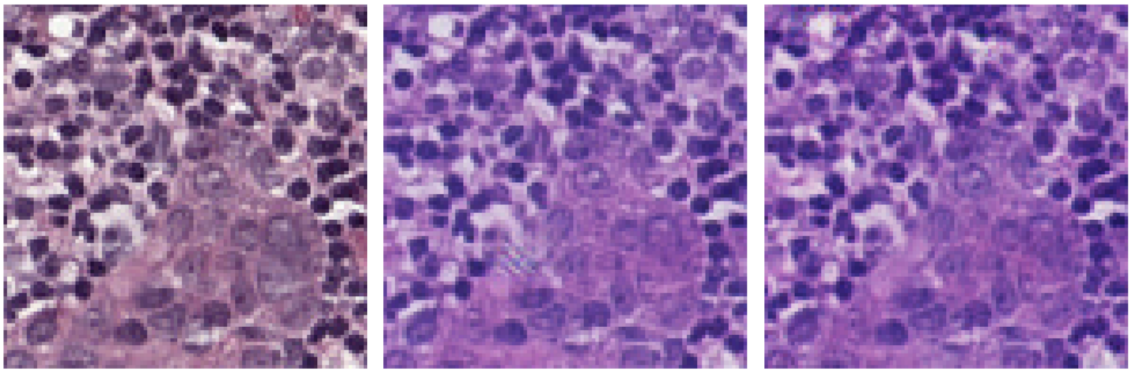

Figure 8: From left to right : Original image from the training set (center 0), Multi-Cycle GAN transfo in the test domain, Center 0 specific Cycle GAN transfo in the test domain.

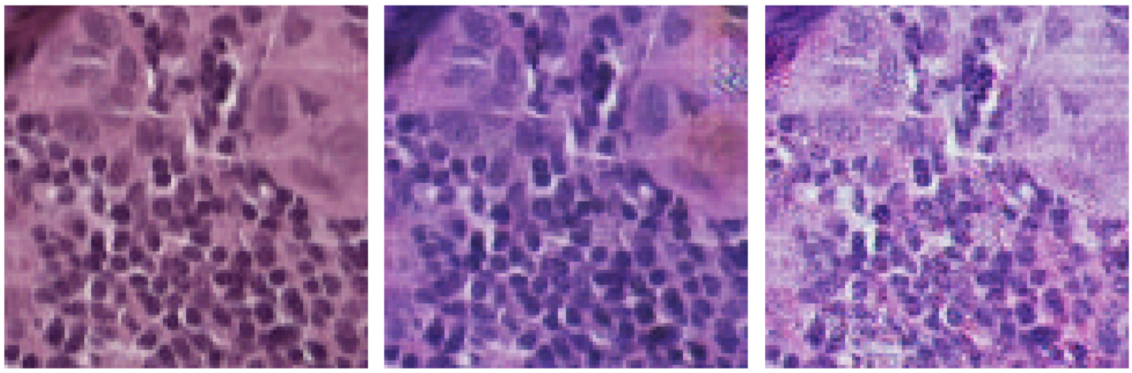

Figure 9: From left to right : Original image from the training set (center 1), Multi-Cycle GAN transfo in the test domain, Center 1 specific Cycle GAN transfo in the test domain.

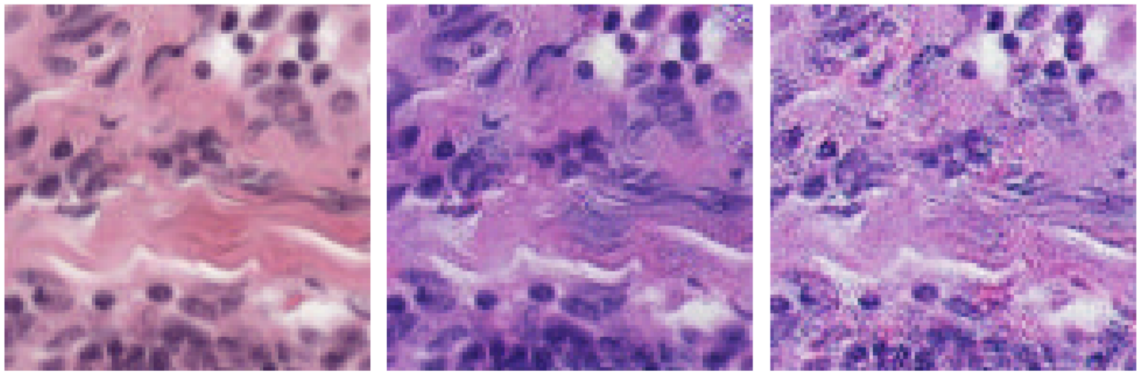

Figure 10: From left to right : Original image from the training set (center 3), Multi-Cycle GAN transfo in the test domain, Center 3 specific Cycle GAN transfo in the test domain.

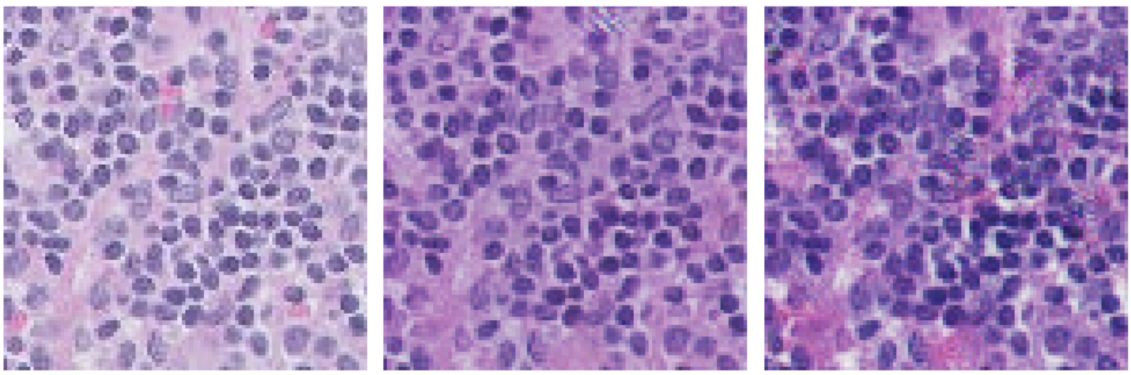

Figure 11: From left to right : Original image from the training set (center 4), Multi-Cycle GAN transfo in the test domain, Center 4 specific Cycle GAN transfo in the test domain.

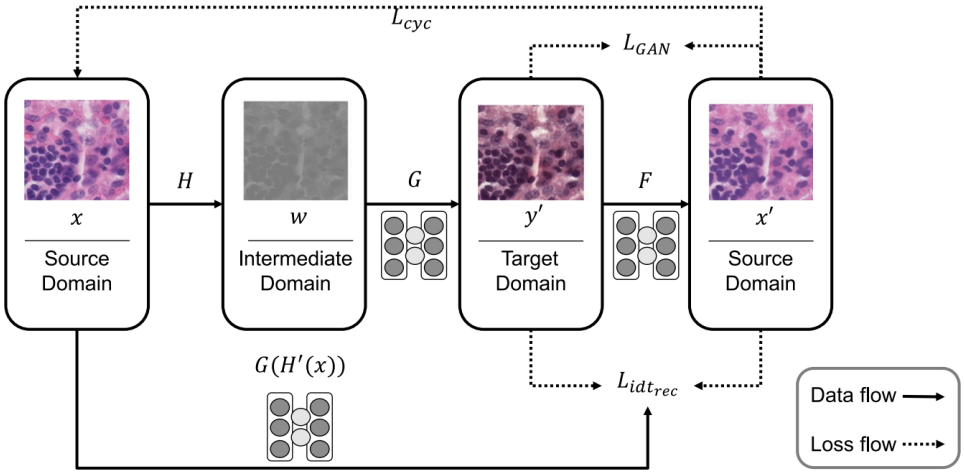

Figure 12: Architecture of MultiStain-CycleGAN: Input RGB image is converted to grayscale, color-augmented, then reconstructed to match target style. The cycle loss is applied in RGB and grayscale space.