Mixing weakly supervised and self-supervised learning techniques for melanoma relapse detection

IMA206

Roxane Goffinet Elouan Gardès Tahitoa Leygue Thibault Mazette

***Abstract*—Melanoma, a highly malignant form of skin cancer, poses significant challenges in early and accurate detection due to its high metastatic potential. Traditional supervised approaches for melanoma detection rely on comprehensive and precise annotations, which are time-consuming and costly to obtain. In this paper, we explore the combination of self-supervised and weakly-supervised learning techniques as promising alternatives for melanoma relapse detection. These approaches leverage the intrinsic information present in the data without the need for extensive manual labeling. We investigate two self-supervised learning frameworks, simCLR and BYOL, to learn rich representations of melanoma images. We also introduce an attention-based deep multiple instance learning (abMIL) framework to leverage weak image-level labels for detecting malignant regions. Our experiments utilize a dataset obtained from the visiomel challenge, consisting of 2.6 million patches from 1,209 patients. We demonstrate the effectiveness of our proposed approach through comprehensive evaluations and comparisons with existing methods. The results highlight the potential of combining weakly-supervised and self-supervised learning techniques for accurate melanoma relapse detection, offering a more cost-effective and scalable solution for clinical applications.**

***Index Terms*—self supervised learning, weakly-supervised**

I. Introduction

Melanoma is a skin cancer that develops from the cells responsible for skin pigmentation. This type of cancer accounts for around 10 % of all skin cancers, and is the most dangerous due to its high probability of metastasis, i.e. spread [10]. For this reason, early and accurate detection of melanoma is of paramount importance in improving the chances of cure and reducing the morbidity associated with this disease.

Traditional detection methods often rely on supervised approaches requiring accurate and comprehensive labels of malignant regions and instances. However, acquiring these annotations is costly and time-consuming, limiting the applicability of these methods. To overcome this, new approaches based on self-supervised and weakly-supervised learning have emerged as promising alternatives. These methods exploit the intrinsic information in the data without the need for manual, comprehensive labeling of malignant regions. Instead, they only rely on label at the patient level.

To learn rich representation of data, unsupervised learning usually consists in creating a supervised task for a model to solve, such a task designed so that its realization should rely on understanding the data at a deep level. It usually relies on modifying the data to provide the original as a target for some downstream task that shall need from the model that it sees beyond these modifications. On the other hand, weakly supervised learning aims to exploit limited image-level labels to detect malignant regions. Rather than providing precise annotations for each skin region or even for pixels, these methods are only shown general labels at a high level and must learn from that feedback only. This enables more cost-effective annotation and facilitates large-scale deployment.

In this report, we will explore combinations of self-supervised and weakly-supervised methods for melanoma detection. We will examine the main concepts, the neural network architectures commonly used and the techniques specific to each approach. We will also evaluate the per-

formance of these methods, compare them with results published in other papers and, finally, discuss their respective advantages and limitations.

II. Presentation of the data

To gain a comprehensive understanding of both the problem and the network architecture we will be implementing, we begin by providing an overview of our data. The data used for our study was sourced from the visiomel challenge [10], and a portion of it had already undergone preprocessing. The data was organized in folders, with each folder representing a specific patient. For each patient, we had between a few hundreds and several thousands patches which, combined, represent a whole slice of the patient's skin. In total, we have 2,600,000 patches, each consists of an image with dimensions 256 x 256. While we possessed labels indicating which individuals experienced a recurrence of skin cancer, we did not have annotations at the instance level nor at the pixel level.

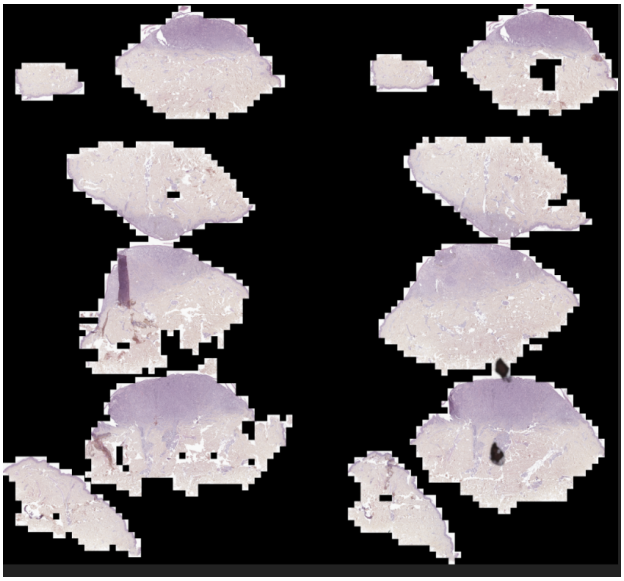

Figure 1: Slice of a patient

Out of all the patients in our dataset, 84 % (1018) were identified as healthy, while 16 % (191) had experienced a cancer relapse. Furthermore, it should be pointed out that most of the patient instances with cancer recurrence are similar to those of healthy patients, and therefore do not aid classification. This leads to a strong imbalance in our dataset when handling instances, which we will have to take into account in the future.

Two split of the whole dataset were followed in order to ensure good evaluation capabilities at every step. The first step, self-supervised learning, aims at gathering good embeddings of all instances. At this point, the full dataset is divided into a training and test dataset, keeping the proportion of positive and negative patients. Then, the training set is further divided into another training set and a validation set, still keeping proportion of patients. Using a validation set which is a portion of the initial training set allows us to test properly the weakly-supervised learning model, with less errors coming from the self-supervised step. Also, keeping data unseen from both the self-supervised and weakly-supervised steps allows us to assess precisely the performance of the whole pipeline.

This split can be seen in figure 2

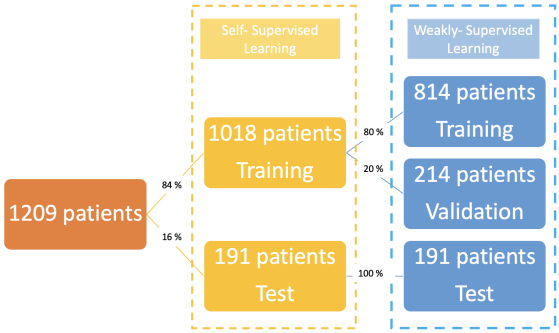

Figure 2: Data architecture diagram

III. Framework

A. *Self-supervised learning*

Self-supervised learning is a machine learning approach where a model learns from the data itself, without requiring explicit human-labeled annotations. Instead, it leverages the inherent structure or information present in the input data to create surrogate supervised tasks [9]. The key idea behind self-supervised learning is to use the vast amount of unlabeled data available, which is easier and cheaper to collect compared to

labeled data. It comes from the fact the self-supervised learning is built to generate a powerful and compact representation of data.

There are various models of self-supervised learning that perform well on images but we will focus on two of them, simCLR [4] and BYOL [5], which we have trained and tested. Of course, other models do exist, such as DiNo [7], but training them is often very time-consuming and requires considerable computational resources.

Both studied approaches belong to the family of contrastive self-supervised learning models. These models create a supervised task for the algorithm to solve, hoping that good representation of the data must be acquired in order to perform these artificial tasks. In both SimCLR and BYOL, two different versions of an instance are obtained through augmentations, then the artificial task is to compare the two embeddings of these augmented instances in some way.

1) Data Augmentation: Data augmentation plays a pivotal role in facilitating the effective functioning of self-supervised learning frameworks. The selection of appropriate augmentations is essential to ensure that the network learns relevant features. For instance, in our specific case, the color of the images does not provide informative cues, and thus, we aim to prevent the network from relying on color distribution for learning. To address this, we employ color jitter, a technique that introduces random alterations to the color channels of an image, resulting in variations in brightness, contrast, saturation, and hue. By employing color jitter, we enhance the diversity of the training data and improve the models' robustness against changes in color and lighting conditions.

In addition to color jitter, we utilize other augmentations to mitigate geometric biases, such as random horizontal flipping and Gaussian blur. However, it is crucial to be mindful of the augmentations used, as we aim to avoid disturbing the learning process. While cropping and resizing are commonly employed in self-supervised learning research papers, particularly with the ImageNet dataset, to allow the network to focus on specific object characteristics, these approaches may not be directly applicable in the context of biomedical imaging. The distribution of information in biomedical images differs from that in general object images, requiring a careful tradeoff. Consequently, we have explored various configurations for data augmentation and have selected the most effective ones. The parameters employed in our chosen configuration are detailed in the subsequent sections.

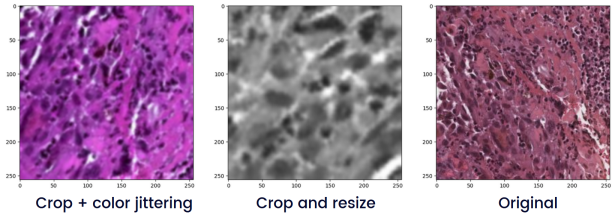

Figure 3: Example of data augementation

2) SimCLR: SimCLR is a self-supervised learning framework that aims to learn useful representations from unlabeled data. The key principle of SimCLR is to encourage the model to learn representations that are invariant to various data augmentations, hoping to then grasp the underlying informative embeddings.

The SimCLR framework consists of two main steps: data augmentation and contrastive learning. In the data augmentation step, two augmented views of each input sample are created. These views are generated by applying different augmentations to the original input. We can see in the Figure 4 that we start by sampling two separate data augmentation operators from the same family of augmentations ($t \sim T$ and $t' \sim T$) and we apply them to the data example. The main augmentations are cropping, rotation, or color distortion but Gaussian, mean or sobel filters have also been studied [4].

In the contrastive learning step, the model is trained to maximize agreement between different augmented views of the same sample while minimizing agreement between views of different samples. This is done by using a contrastive loss function in the latent space. It thus encourages the model to map similar views close together in the learned representation space and push dissimilar views apart. In order to do that, we train a base encoder network $f(.)$, followed by a projection

head $g(.)$.

By training the model on this contrastive learning objective, simCLR learns representations that capture meaningful and useful features from the data. These learned representations can then be used for downstream tasks, such as classification or regression, by adding a task-specific head on top of the learned representation.

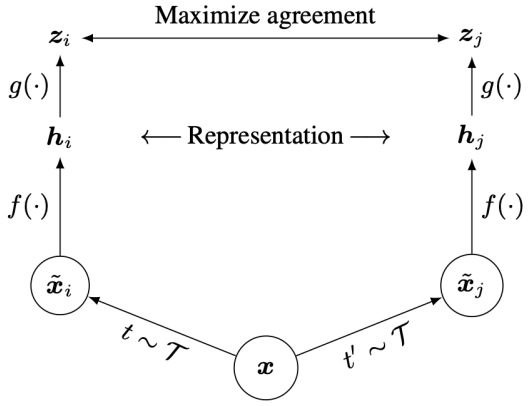

Figure 4: Illustration of SimCLR's principle [4]

3) BYOL: BYOL (Boostrap Your Own Latent) is a self-supervised image representation learning method that relies on two neural networks that interact and learn from each other [5]. Those two networks are referred to as online and target networks, see figure 5. The online network is trained with an augmented view t of an image to predict the target network representation of the same image under a different augmented view t'. During the same time, the target network is updated with a slow-moving average of the online network. In the online network a projection head (q_θ) is added to the projection (z_θ) output in order to obtain a prediction that will be used to calculate the loss and do the back-propagation. The authors' motivation to add the non-linearity was to prevent the model from learning the augmentations.

In contrast to simCLR, BYOL focuses exclusively on positive pairs, which proves compelling within the visiomel context due to the abundance of closely related patch representations. While we aim to ensure that different augmented views of a single patch remain highly proximate in the latent space, we do not necessarily aim to compromise the representation of other patches.

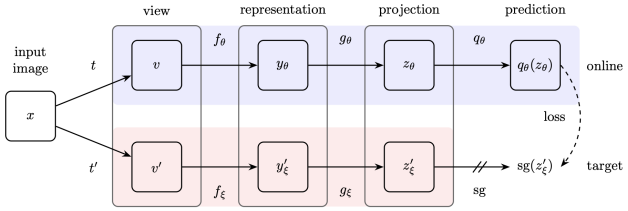

Figure 5: Illustration of BYOL's principle [5]

For traning the network, we use an MSE loss which compute the euclidian distance in the latent space between the prediction of the online network and that of the target network.

$$\mathcal{L}_{\theta,\xi} = \|\bar{q_\theta}(z_\theta) - \bar{z}'_\xi\|_2^2$$

BYOL also come with a slow-moving average (SMA) mechanism that plays a crucial role in BYOL's robustness. SMA enables the network to attend to informative regions in the input data, enhancing the learning of meaningful representations. The SMA mechanism leverages a separate network, known as the momentum encoder, which computes a moving average of the online network's parameters. By using this moving average as a target for the online network, the SMA mechanism encourages the network to focus on capturing important information that is essential for generating consistent representations across different views of the data. This attention mechanism promotes the extraction of relevant features while suppressing the influence of noisy or irrelevant components, leading to more robust representations.

By training the online network to match the target network's representations, BYOL enables the model to learn meaningful and transferable features from the unlabeled data. These learned representations can then be fine-tuned or used for downstream tasks, such as classification or regression, by adding task-specific layers on top of the network.

B. Weakly-supervised learning

Multiple Instance Learning (MIL) is a machine learning problem that differs slightly from classi-

cal learning problems in that each image is not a single instance with a label indicating the class to which it belongs. In MIL, training examples are organized in sets called ”bags”. Each bag is made up of several instances, but unlike conventional problems, here the class label is assigned to the bag as a whole. MIL's models belong to the large class of weakly supervised learning, i.e. learning with very weakly labelled data.

In our case, we want to create a system capable of detecting malignant tumors in medical images. We are provided with a label at the patient level only. We will thus consider that each patient represents a bag. For each patient, we have patches which are parts of a whole slice too big to be processed directly. These patches will be the bag's instances. The instances have hidden-labels, i.e they are either containing cancerous cells or not, but this information is not accessed during training.

The goal of the MIL approach is to find some good aggregation of all instances for one patient in order to perform a prediction at that high level. We can then compare the prediction with the bag's, or patient's, label. We then backpropagate a loss for that task.

C. Attention-based Deep Multiple Instance Learning)

AbMIL or *Attention-based Deep Multiple Instance Learning* is a framework that combines the concepts of Multi Instance Learning and attention mechanisms to address the problem of learning from bags of instances [3]. For medical applications, having an explicable model is crucial in order to provide guarantees on its performance but also to understand its decision, which can then be better completed by an expert opinion. The attention mechanism provides some insight on that level.

AbMIL takes a bag's embedding as input, i.e the matrix comprised of the embeddings of all instances for a given patient. Such a matrix is thus of shape NxD, where N is the number of instances for the current bag (N is thus not fixed across the dataset), and D is the dimension of embeddings coming out of the backbone trained with self-supervised learning. In this review, that backbone

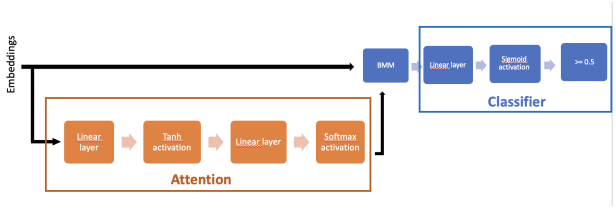

Figure 6: Schéma de l'architecture AbMIL

is a resnet18 without the prediction head, it thus produces embeddings of size 512 from 256x256 instances.

AbMIL incorporates an attention mechanism to assign attention weights to the instances within each bag. This mechanism allows the model to focus on the most discriminative or relevant instances for bag-level prediction. The attention weights are then used to aggregate the encoded instances within each bag, producing a compact representation that captures the most informative aspects of the bag. Various aggregation techniques, such as weighted sum or weighted average, can be used, here AbMIL uses a learnable classifier neural-network with a sigmoid activation, as wee can see in the diagram on Figure 6. This aggregation thus produces a single number representing the probability that a bag is positive.

We can then use the provided label for the bag to process a binary-cross-entropy loss and backpropagate to learn good attention and classification weights.

IV. Implementation & Results

A. Self-Supervised Learning implementations

For our implementation, we used Python as our programming language and executed our code on the GPUs available at Telecom Paris. To facilitate code sharing and collaboration, we utilized GitHub as our platform.

Initially, we began by implementing simCLR and BYOL from scratch. However, we soon realized that training these models was time-consuming, even when using popular backbones like resnet18 (although the articles achieved good results with resnet200 [4, 5]). To overcome this challenge, we decided to leverage the solo learn library. This library provided optimized code specifically designed for training simCLR and

BYOL, along with other techniques. It allowed us to select the backbone of our choice, which was pre-trained on the large-scale ImageNet dataset. By using the solo learn library, we were able to streamline the training process and improve efficiency in our implementation.

1) SimCLR: We employed the simCLR framework using various pre-trained backbones available in the Solo Learn library. To fine-tune our models, we conducted multiple simulations, aiming to identify the optimal set of parameters that would minimize the loss function.

The advantage of SimCLR is that it takes into account in its loss both the positive pair and all the negative images of a given batch, thus exploiting all the images of a batch. It seeks to move the positive pairs closer together and away from the negative images. However, for various reasons, some negative images may be very close to the positive point in the representation space. This is particularly the case in our problem given that we are dealing with medical images of skin cells, so there are strong similarities between the diseased patches and the healthy patches given that the tumours represent only a tiny proportion of the pixels in the images. Another argument in favour of this is that the two classes are unequally represented; there are far more healthy patients than diseased patients in the dataset, and even within the diseased patients, some patches have no tumours. We therefore end up with the majority of negative samples that are 'hard negative samples', i.e. points that are difficult to distinguish from our target point. However, contrastive learning methods benefit from hard negative samples [6]. According to a recent article by Kumar Pranjal and Chauhan Siddhartha [8] who studied the influence of temperature τ on contrastive learning methods, it is preferable to use a low temperature (of the order of 0.2 or 0.3) to penalise the hard negative samples more so that they can be better taken into account and distinguished.

We had to adapt the model's hyperparameters to our problem and our dataset. To find the parameters, we ran several tests on a small sample of patients in order to evaluate the influence of the different parameters on loss, these results can be seen in Figure 7. The backbone used for feature extraction is a **resnet18**. For reasons of time and hardware constraints, we opted for a simpler model than the resnet50. A resnet50 is perhaps too complex a model for medical images, with a fairly high risk of overfitting. The learning rate was set at **0.3**, and taking a higher value runs the risk of not being able to converge properly as the epochs progress, and we didn't have the time to take a lower learning rate. The weight decay is **1e-6**, tests with a lower value do not show a significant drop in the loss, the difference observed can be explained by the increase in overfitting. The batch size was set at 128 patches. From the various curves displayed, it can be seen that a batch size that is too large leads to a significant increase in loss. One intuition is that too many negative patches are represented and so the model needs more time to understand the image representations; for logistical reasons, we could not afford to opt for such a strategy, especially as several GPUs would have been needed in parallel. Finally, as explained earlier, the temperature was set at **0.2**. We would have liked to have had more time to try out other combinations of hyperparameters.

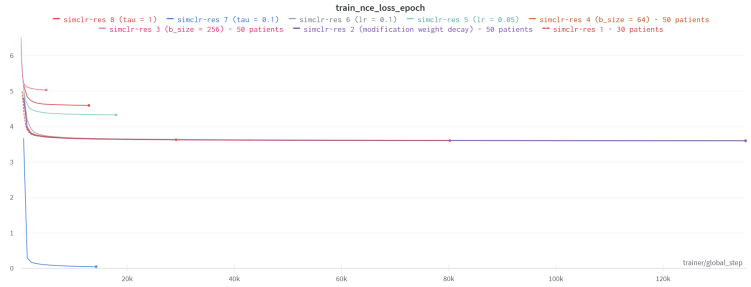

Figure 7: SimCLR loss function value as a function of epochs

Although we achieved some promising results with the selected parameters, the overall performance remained unsatisfactory. Despite our efforts to fine-tune the models, we encountered challenges in achieving the desired level of accuracy and effectiveness in our predictions. It is worth noting that simCLR, which was originally evaluated on the ImageNet dataset, may not be the most appropriate choice for medical images because they have distinct characteristics and

variations that differ significantly from the images in ImageNet. These differences can impact the performance of the algorithm when applied to medical imaging tasks.

2) BYOL: Although the configuration we chose for the ResNet18 (for instance, papers show that ResNet50 and even wider networks like x4 ResNet50 gives best results) did not align with the optimal settings outlined in the BYOL paper, it was a pragmatic decision based on our limited computational resources and project timeline.

During the training process, we encountered several challenges related to adapting the simCLR framework to our medical histopathology dataset. One significant hurdle was determining the most appropriate model configuration, as the settings described in the BYOL paper were originally designed for ImageNet and Vision datasets. Unlike ImageNet, where relevant information is often concentrated around a central object, histopathology images contain critical information distributed throughout the entire image. Consequently, we had to carefully tailor data augmentation techniques to preserve the integrity of the histopathology images. The parameters and augmentations we used are shown in table I.

Moreover, training the ResNet18 model with SSL proved to be a computationally intensive task, requiring substantial time and resources. The SSL training alone, conducted over 50 epochs on our dataset, took approximately four days. Given the intermediate nature of the SSL training step in our overall research pipeline, it was challenging to draw conclusive insights solely from the SSL training process. However, we closely monitored the loss function's behaviour, examining its shape, slope, and stability as indicators of model convergence.

To further optimize the training process and the performance of our model, we explored various configurations by adjusting parameters such as batch size, learning rate, and network weight initialization. These parameters played a crucial role in determining the network's convergence and overall performance. Throughout our experimentation, we generated and analyzed learning curves to gain valuable insights into the effects of different parameter settings.

Transformation	Parameter	Value
color_jitter	prob	0.8
	brightness	0.8
	contrast	0.7
	saturation	0.6
	hue	0.2
grayscale	prob	0.2
gaussian_blur	prob	0.5
horizontal_flip	prob	0.5
crop_size	-	256

Table I: Data Augmentation Parameters

Our research highlights the complexities and unique considerations associated with adapting SSL methods, including simCLR and BYOL, to the specific domain of medical histopathology. By overcoming challenges related to data augmentation and parameter selection, we established a solid foundation for the subsequent MIL-based data processing in our research. We did the different tests on 50 epochs and the results of the training loss curves are in the figure 8.

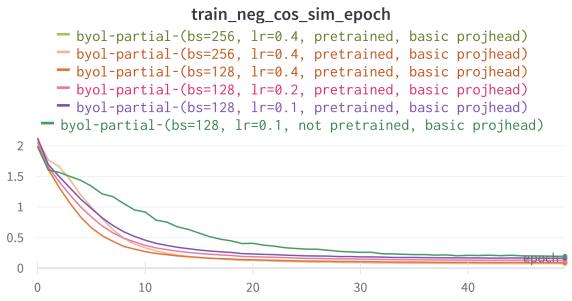

Figure 8: BYOL Train Loss on 50 epochs on a 4 patients from the dataset

Even if we see that there is not a big difference between the different curves, we can note that as long as the learning rate is not too low, all the learning curves are very close to each other.

Given those training parameters, we then train the model with the whole data set using a batch size of size 256 and a learning rate of 0.2 using random weights initialization and a pretrained one, so we will be able to analyse them later. The loss for those model are presented on Figure 9.

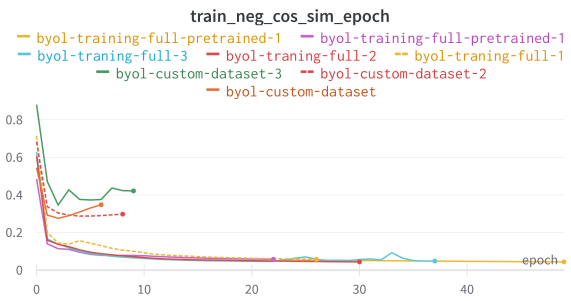

Figure 9: BYOL Full training loss curves

B. AbMIL

1) Adapting AbMIL to our problem: The simple binary-cross-entropy loss has had to be adapted to our very imbalanced dataset. Having 84% of negative patients, a classic binary loss would find its local minimum with a model always predicting "negative" for every patient. We thus use a balanced version of the binary-cross-entropy:

$$\mathcal{L}_{balanced} = -pf \cdot y \cdot \log(proba) - nf \cdot (1-y) \cdot \log(proba)$$

where y is the ground truth (0 negative, 1 positive), $proba \in [0, 1]$ is the output of the model and pf, nf respectively the positive and negative frequencies. The frequencies are the proportion of positive and negative samples in the dataset.

Still tackling the dataset's imbalance, we could not rely only on accuracy to assess the training performances of the model, since a 84% accuracy could be reached simply by classifying every patient as negative. Instead, we thus used the Area Under the Curve (AUC) (see fig 10) of the curve giving the true-positive rate as a function of the false-positive rate. This metric gives us a good glance of the model true performance in terms of balance between classification of positive and negative patients.

2) Training metrics: Balanced loss, accuracy and AUC are computed for each training to compare between different backbones (embedding extractors) and between different hyperparameters for the MIL classifier.

Bellow (see Figures 11 and 12) are typical training curves giving metrics on the training and on the MIL validation set.

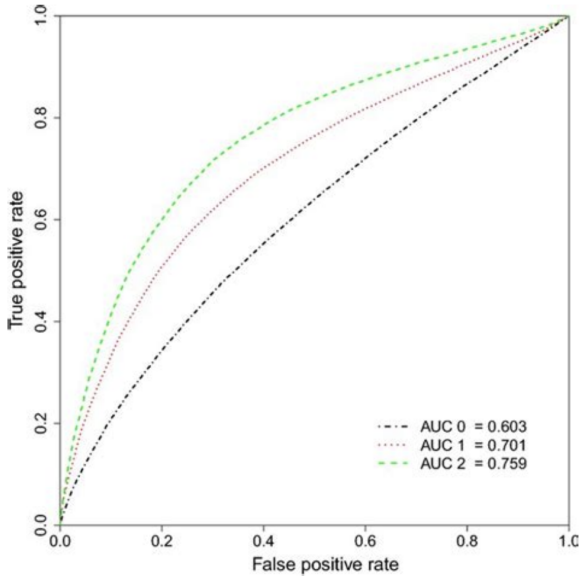

Figure 10: AUC for AbMIL

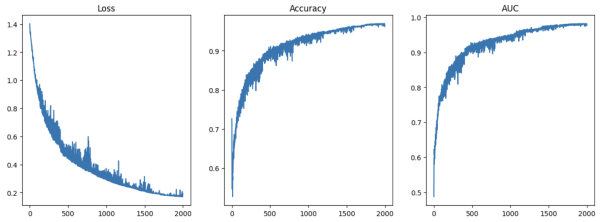

Figure 11: Typical metrics for AbMIL on training set

Notice on the training accuracy graph: over the first 10 epochs or so, accuracy drops drastically. This comes from the fact that the model initially classifies most samples as negatives, yielding a high accuracy and a high loss. Accuracy then drops when the model adapts through the balanced loss, before increasing again.

Our three metrics improve to near-perfect predictions during training. There is however a high overfit, with the validation metrics finding their sweet spot at around 100 epochs. We can see moreover that accuracy finds a plateau at around 80%, and at the same time the loss increases a lot and the AUC decreases. This means that the model becomes increasingly confident in its mistakes (high loss) and starts predicting many patients as negatives (high accuracy and low AUC).

To tackle overfitting at the MIL level, we tried:

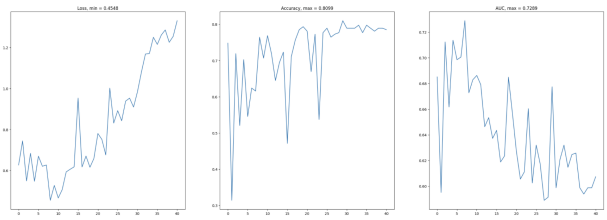

Figure 12: Typical metrics for evaluation of Ab-MIL on validation set (1 value every 10 epochs)

label=•

- Dropout (different values, different layers),
- L1 and L2 regularization (constraint on the model weights' complexity),
- Take instances inside a bag in a random order when passing the bag through the model,
- Drop some instances for bags in which their number far exceeded the mean number of instances per bag (typically over 3000 instances).

None of these approaches improved our validation metrics. If it reduces overfitting, it is always because of worse training metrics - not better validation metrics.

We have some ideas left to try and improve overfitting at the MIL level: label=•

- Augmenting positive bags: adding some noise to every instance's embedding of a positive bag to add new positive samples and balance a bit the dataset,
- Every K epochs, take M epochs to train only on positive bags (with K and M hyperparameters, for instance K = 20, M = 3),
- Simplify the model, eventhough it is already rather simple.

3) Results: Here bellow are reconstructed whole slices from their patches, with some patches highlighted by the attention module as important for the decision. Note that besides its explicability advantages, the attention module is also very useful to predict as most patches are shared between positive and negative bags, they are thus not informative for prediction.

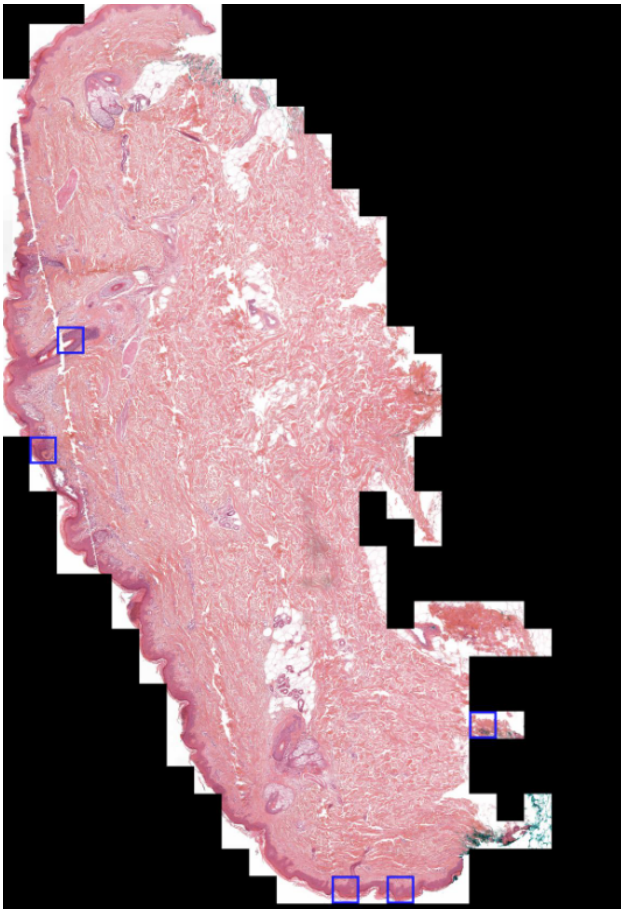

Figure 13: Positive patient 1

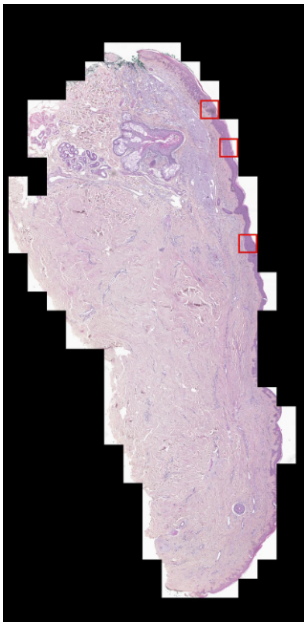

Figure 14: Positive patient 2

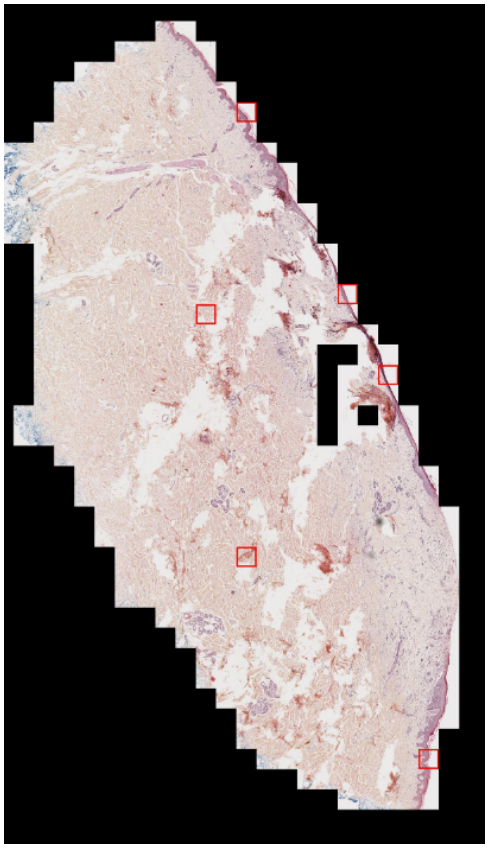

Figure 15: Negative patient

We can notice on all three slices that the attention module focuses on border. They seem to be very informative to make prediction of melanoma relapse.

By comparing between positive and negative slices, we can also notice that indeed the border's features change a lot, especially its thickness. We find that positive patients show thicker borders than negative ones. This is coherent with the admitted findings in the medical field regarding melanoma (see e.g [1]).

Our results confirm that our pipeline works to some extent at least: we manage to gather informative features through unsupervised learning, and our AbMIL is able to then find the best instances to take a decision at the patient level.

We can also see (figure 16) that, when a misclassification happens, it can probably mostly be attributed to a poorer attention map.

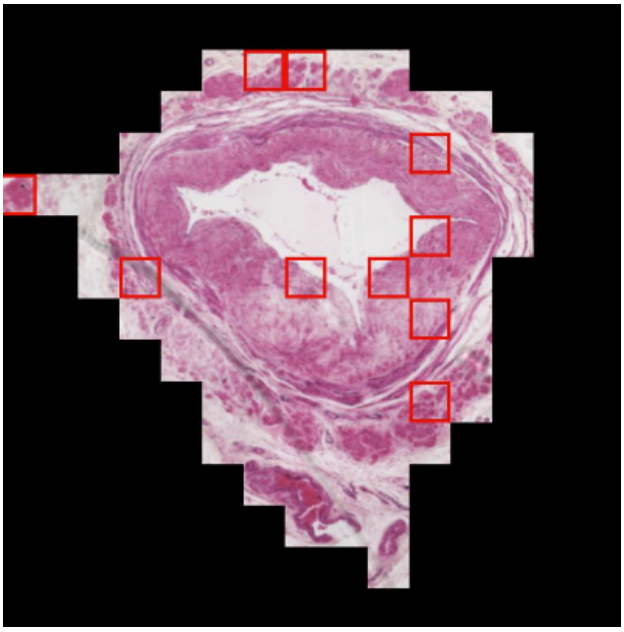

Figure 16: Slice of a misclassified cancer patient

Let us now compare backbones, i.e embeddings extractors. We compare them by computing loss, accuracy and AUC after the MIL model. A different backbone means that the bags taken as input by MIL change, the features of each instances are different. A summary of the results can be seen in the Table II.

All backbones are resnet18. The difference is simply their weights, obtained through different training methods. The choice of a resnet18 is a pragmatic one: training self-supervised method with the dataset at hand requires several days on Nvidia A100 graphic cards with that backbone. It would take weeks with a resnet50 or wideresnet. Also, Resnet18 gives embeddings of size 512 from 256x256 patches, which amounts to around 5 gigabytes of data in embedding form (when storing them as torch .pt matrices, one matrix for one patient).

We can first notice the performance of the baseline resnet18 pretrained on ImageNET. *A priori*, ImageNet images are vastly different from our medical ones, but it seems enough to have a latent space in which positive instances are different enough so that MIL picks something up. In particular, it should be pointed out that this backbone yields better metrics (especially

Backbone	min loss	max accuracy	max AUC
Resnet18 pre-trained on ImageNET	0.82	0.70	0.61
SimCLR - Resnet18 with random base weights	1.02	0.72	0.57
BYOL 20 epochs - Resnet18 with random base weights	0.99	0.73	0.59
BYOL 20 epochs - Resnet18 pre-trained on ImageNet	0.45	0.81	0.73
BYOL 30 epochs - Resnet18 pre-trained on ImageNet	0.55	0.80	0.68

Table II: Results obtained for AbMIL with different backbones

loss and AUC) than SimCLR and BYOL resnet18 when not pretrained. Also, BYOL yielded better metrics than SimCLR on a randomly-initialized resnet18. This tends to confirm that BYOL is more suited than SimCLR for medical images where most images should be close and negative pairs are probably often counter-productive. BYOL with a pretrained backbone led to the best result on all metrics. This indicates that the self-supervised learning does improve representations of our dataset for the prediction task. Finally, we point out that training for longer periods of times leads to poorer metrics for the prediction task, eventhough the self-supervised learning task's loss still decreases.

Let us now compare our best metrics with the Visiomel winners. We take the same metrics, namely a classic binary-cross-entropy (Log-loss score), as well as accuracies and false negative, false positive rates. We show (see Table III) that our pipeline outperforms by far the winner's approaches when it comes to detection of relapse events. This comes at the cost of poorer prediction of negative patients, and an overall poorer accuracy because of the dataset imbalance.

Rank at VisioMel Challenge	1st	2nd	3rd	Our best model
Log-loss score	0.39	0.40	0.40	0.47
Percentage of correct prediction	82.26	81.52	81.70	80.00
Percentage of no-relapse event correctly predicted	96.30	96.30	95.15	80.00
Percentage of no-relapse event correctly predicted	96.30	96.30	95.15	83.00
Percentage of false negative	16.10	16.77	15.92	40.00
Percentage of relapse event correctly predicted	25.93	22.22	22.78	60.00
Percentage of false positive	36.36	40.00	41.18	18.00

Table III: Comparison to the top results of the VisioMel Challenge

We have to note however that our good performance has to be put back in context, that is the validation set used to get these metrics is the one of the MIL step, which is a part of the training dataset for the BYOL step. We shall study how our pipeline performs on the test set, data unseen from both the MIL and the BYOL steps.

V. Dissucsion

In the end, we were able to achieve promising results, although they should be interpreted with caution due to the lack of rigorous evaluation on the test set and thus, the generalizability and reliability of our results remain uncertain. Our

progress was hindered by time and computational resource limitations, which prevented us from exploring all possible avenues for improvement, as for example, a single epoch on BYOL took around 4 hours. Nevertheless, even experienced medical professionals face challenges when it comes to accurately predicting cancer relapse. Despite their expertise, accurately foreseeing whether an individual will experience a cancer relapse remains a formidable task as the complex nature of cancer makes it difficult to determine with certainty the likelihood of recurrence. This, highlights the need for advanced computational techniques, to assist and improve the accuracy and reliability of prognostic predictions and ultimately enhance patient outcomes.

Then, the data used for training our machine learning algorithm may not be ideal due to its origin from real-world sources, specifically Parisian hospitals and collected over the past 15 years. These data present challenges for machine learning as they lack standardization and exhibit variations resulting from evolving techniques over time. The dataset includes diverse factors such as variations in cell colorization, different zoom levels, scales, and instances of blurry slices. These characteristics make the training process more complex and introduce additional difficulties in achieving reliable and consistent results. In the future, we may need to look at avenues such as image recolorisation [2] to try and standardise the dataset. Perhaps also other processes for segmentation and patching. Additionally, it is challenging to pinpoint exactly which aspects need improvement, such as preprocessing techniques, data augmentation, self-supervised learning, or multi-instance learning as the nature of our experimentation with the AbMIL framework makes it difficult to evaluate the effectiveness of each individual component until the completion of the entire process. Without concrete results from the AbMIL framework, we are limited to monitoring loss values as an indicator of progress. However, this provides limited insight into the specific areas requiring refinement.

In conclusion, despite the time-consuming nature of working on a concrete machine learning project tailored to image analysis, we found the experience highly rewarding. The obtained results have been quite satisfactory, reflecting the potential of the applied techniques. Moving forward, it would be beneficial to allocate additional time and resources to examine and improve each aspect of the processing in order to enable us to achieve better outcomes and thus enhance the efficacy and reliability of our model.

REFERENCES

[1] N. R. Abbasi et al. "Early Diagnosis of Cutaneous Melanoma: Revisiting the ABCD Criteria". In: *JAMA* 292.22 (2004), pp. 2771–2776. DOI: 10.1001/jama.292.22.2771.

[2] Raj Kumar Gupta et al. "Image Colorization Using Similar Images". In: *Proceedings of the 20th ACM International Conference on Multimedia*. MM '12. Nara, Japan: Association for Computing Machinery, 2012, pp. 369–378. ISBN: 9781450310895. DOI: 10.1145/2393347.2393402. URL: https://doi.org/10.1145/2393347.2393402.

[3] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. "Attention-based Deep Multiple Instance Learning". In: *CoRR* abs/1802.04712 (2018). arXiv: 1802.04712. URL: http://arxiv.org/abs/1802.04712.

[4] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG].

[5] Jean-Bastien Grill et al. "Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning". In: *CoRR* abs/2006.07733 (2020). arXiv: 2006.07733. URL: https://arxiv.org/abs/2006.07733.

[6] Joshua Robinson et al. "Contrastive Learning with Hard Negative Samples". In: *CoRR* abs/2010.04592 (2020). arXiv: 2010.04592. URL: https://arxiv.org/abs/2010.04592.

[7] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV].

[8] Pranjal Kumar and Siddhartha Chauhan. "Study on temperature τ variation for SimCLR-based activity recognition". In: *Signal, Image and Video Processing* 16.6 (Sept. 2022), pp. 1667–1672. ISSN: 1863-1711. DOI: 10.1007/s11760-021-02122-x. URL: https://doi.org/10.1007/s11760-021-02122-x.

[9] Wikipedia. *Self-supervised Learning*. https://en.wikipedia.org/wiki/Self-supervised_learning. Accessed: June 14, 2023.

[10] DrivenData. *Visio-MeL: Visualizing the Melanoma*. https://www.drivendata.org/competitions/148/visiomel-melanoma/. Accessed 2023.