Project Machine Learning
— Milestone 1 —

Filip Matysik, usr: pml16

November 22, 2024

1 Introduction

This report describes the work that has been done for milestone 1 of the project that is aiming to replicate the results of the TransMIL (Shao et al. (2021)) paper, specifically on the CAMELYON16 (Ehteshami Bejnordi et al. (2017)).

As for milestone 1, so far both the model and code are not yet fully developed. The model is simplified in comparison with the one described in the TransMIL paper, details of this simpler baseline model have been described in the "Baseline method and evaluation section". Code is also quite simplistic at this stage, and not adjusted to the optimal and efficient conduction of experiments and hyperparameter tuning (pytorch lightning approach to running experiments is not yet implemented in the code, same for wandb). However the crucial and fundamental functionality has been developed, which is the dataloader for all train, test and validation sets as well as the simplified model class and the training and evaluation setup. It is thus possible to load the data, train the model and evaluate it. However at this stage only the basic accuracy metric has been implemented, although this very metric is used for evaluation in the TransMIL paper, so it is obviously a reasonable choice of metric to begin with.

It is also worth noting that the simple baseline model has not been trained on the full dataset yet. Instead a mini version of the CAMELYON16 has been created in order to check that the data loading pipeline works without errors, model is able to compute the forward and backward pass without any bugs, and to ensure that evaluation part works well. Another reason to build such a small dataset was to see if the computed loss will be dropping, suggesting that the model is able to converge at least to some extension.

Basic analysis of the dataset properties has been performed as well as developing some functions that allow for visualization of the data and getting a better understanding of what kind of data one is dealing with in this very task. allowing for the

2 MIL - Multiple Instance Learning and TransMIL overview

Multiple Instance Learning (MIL) is a machine learning framework designed to handle problems where the labels are assigned at the bag level rather than the instance level. In the context of pathology and whole slide image (WSI)-based diagnosis, a bag typically represents an entire slide or a collection of image patches extracted from it, while individual instances refer to the patches themselves. The goal in MIL is to predict the bag-level label (e.g., whether the slide is cancerous or not) based on the features of its instances, without explicitly knowing the labels of individual instances (e.g., whether a specific patch is cancerous).

Traditional MIL methods often operate under the independent and identical distribution (i.i.d.) hypothesis, which assumes that instances within a bag are independent of each other. However, this assumption neglects the intrinsic relationships and spatial dependencies among instances, which are particularly crucial in pathology, where tumor

regions often exhibit spatial patterns and structural dependencies.

To address this limitation, a novel framework called Correlated MIL (cMIL) was introduced, along with a proof of its convergence. This framework led to the development of TransMIL, a Transformer-based MIL model was designed to capture both morphological and spatial information within WSIs. By leveraging the global context modeling capabilities of transformers, TransMIL effectively models correlations among instances while preserving critical local features.

3 CAMELYON16 Dataset overview

3.1 CAMELYON16 introduction and visualization

CAMELYON16 dataset consists of the 399 Whole Slide Images (WSI), which are the high-resolution digital scans of entire microscope slides. In this case, the slides represent tissue samples extracted from various lymph nodes. This was done to analyze the presence of cancer cells in the lymph nodes, which would indicate metastasis. Annotations are provided only for the WSIs and allow for both binary (cancer, no-cancer) and multi-class (no-cancer, micro, macro) classification. This implies that CAMELYON16 dataset is suitable for weakly-supervised machine learning methods such as Multiple Instance Learning, because there are no per patch label annotations and no segmentation annotations either. WSIs are stored in the .tiff format, designed to handle high-resolution images (and allow to store the same image in different resolution levels in one file). Resolution of the highest level of WSIs in the dataset varies from the largest - 217088x111104, to the smallest - 45056x35840. It would take a large number of computational resources to process such large images in whole, so is is much easier to split them into patches during the preprocessing stage. In this form it is then possible to utilise Multiple Instance Learning method as described in the TransMIL paper.

In detail, every WSI has been divided into patches of the same size (256x256). The number of patches between WSIs differs, depending on the size of the WSI and its structure. White background has been discarded, thus the patches do not contain it. Below an example of the WSI is presented:

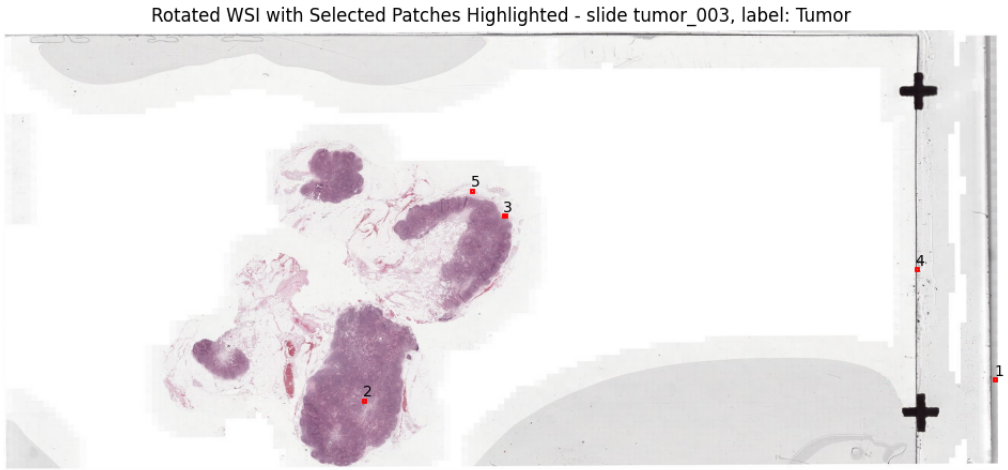

Figure 1: Example of the downscaled Whole Slide Image

It is worth mentioning that each tissue has been collected from different patient according to CAMELYON16 paper ((...) SLNs were retrospectively sampled from 399 patients that underwent surgery for breast cancer at 2 hospitals in the Netherlands: Radboud University Medical Center (RUMC) and UniversityMedical Center Utrecht (UMCU). (...)). This approach minimizes the dataset contating near-duplicate values and exposing the model to evaluation on the data that it has already seen during training (ex. two tissues from the same patient).

For better understanding of the elements present in the WSI, its mask has been visualized below. The mask carries information about the actual area of the tissue in the WSI (grey area), annotated regions where cancer cells form the tumor (white area) and the background (black region), which does not carry any information and has been discarded during the patch extraction.

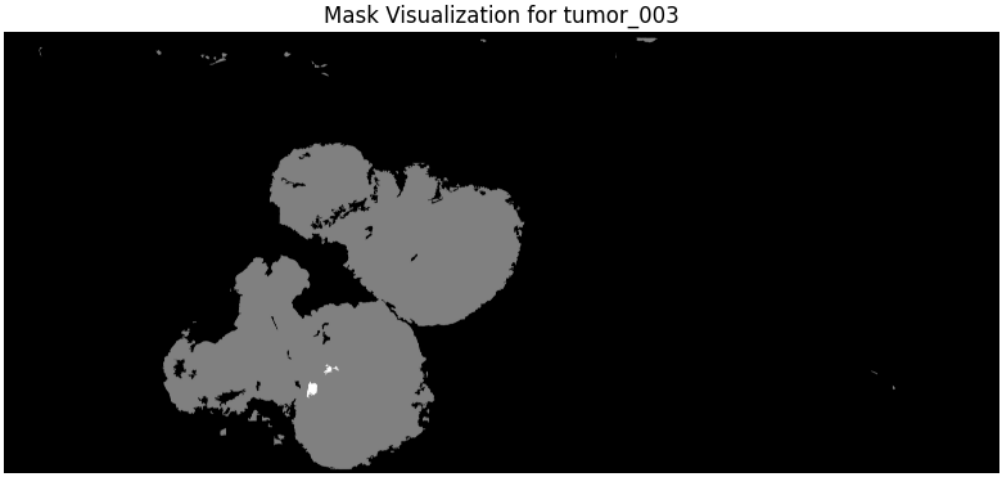

Figure 2: Rotated tumor mask showing the region of the tissue (gray) and marked cancer regions (white)

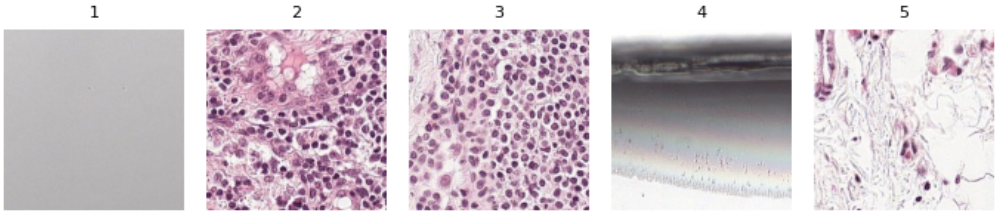

Figure 3: Single patches extracted from the Whole Slide Image

In Figure 3, five random patches extracted from the WSI have been presented. They represent the microscopic view with 20x magnification. It is worth noticing that there's a certain diversity in the dataset in respect to what the patches actually contain. As shown in images 1 and 4 in Figure 3. - patches do not consist of cell regions only and might contain non-informative regions as well. This is desired, because in clinical applications, not all patches will contain useful information and also, it is challenging to perform the patch extraction in a way that it leads to extracting cell regions only, thus training with noisy patches helps to prepare the model for real-world conditions. This however forces the model to learn to differentiate between informative (tissue) and non-informative (background) patches, increasing the complexity of the representation that it has to learn during training.

3.2 Dataset statistics and insight

In total there are 399 WSI images in the dataset. However features have been provided for 398, with features missing for the "normal-144" slide. Thus there are 238 elements in the dataset of the negative class (0 - no cancer), and 160 positive class elements (1 - cancer). The data has been split in the following ratio for the train, test and evaluation subsets:

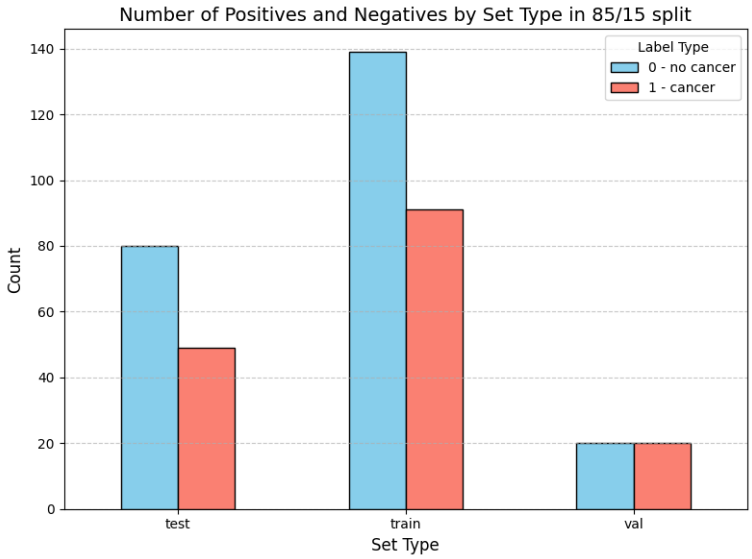

Figure 4: Split of the data for subsets and their within class label ratio

The split mimics the original split provided by the authors of the CAMELYON16 dataset, only difference is the creation of the validation subset out of the training data. There's a slight imbalance between the data classes in both the training and test dataset, thus it is crucial to ensure that the model will actually learn to distinguish between the positive and the negative class and adjust the metrics accordingly, so that there's no possibility of the model simply learning the representation of the majority class and is still able to obtain decent results on the dataset, for which the majority class is the same.

Patches extracted from the WSI are the actual instances within each bag. In this manner one bag represents one WSI. Due to the varying amount of extracted patches from each WSI, input to both the feature extractor and the transformer (Vaswani et al. (2023)) architecture based TransMIL model, vary over the dataset. This is however not a problem for the transformer architecture, because transformers process inputs as sequences, where the length of the sequence can vary. This proves that the combination of the MIL method with transformer architecture is a well suited approach to the problem, where constrainig the inputs to be of the same size would not be possible. Below graph presents the distribution of the number of patches extracted for WSIs in the dataset, to better illustrate the varying input size problem:

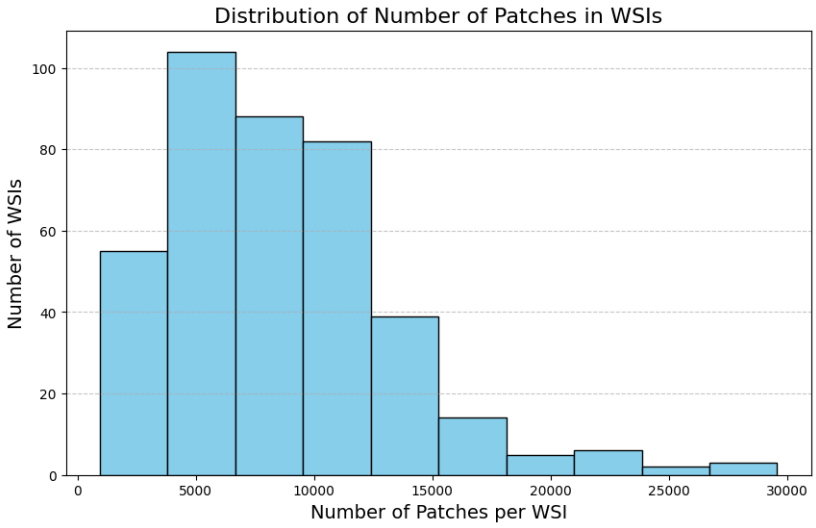

Figure 5: Histogram of the Number of Patches in WSIs, in average about 8,800 patches per bag were obtained

One important characteristics of the dataset is how the positive and negative data

classes actually vary. This can be interpreted from both the macro-statistical and morphological perspective. To get a better understanding of how positive and negative labels differ from the spatial perspective the below graph has been obtained, showing the distribution of the percentage of the tissue area containing cancer cells that form a tumor region, in the positive class (1 - tumor) WSIs:

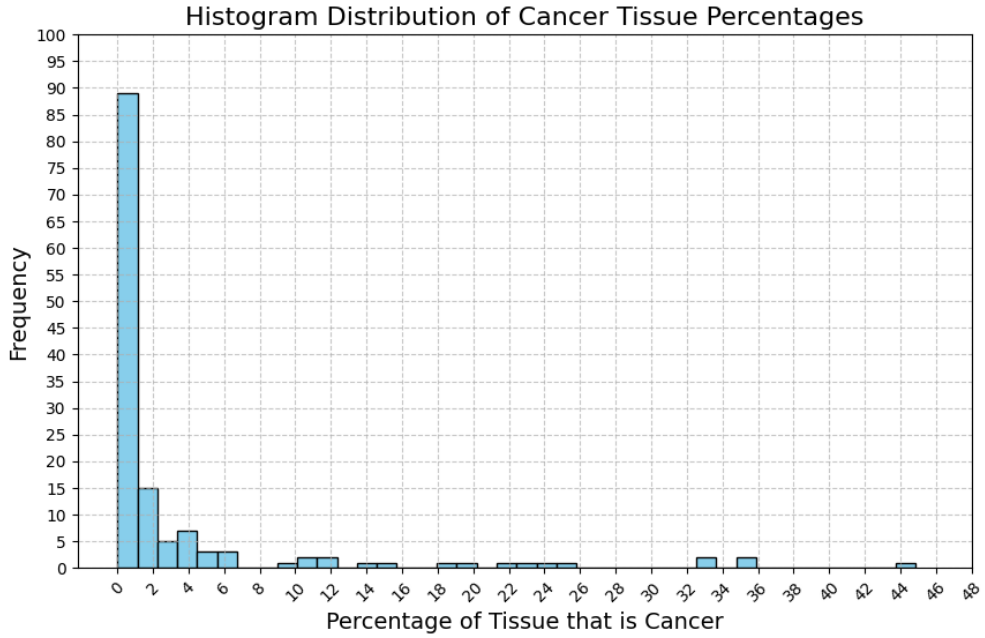

Figure 6: Distribution of the ratio of cancer tissue area with respect to the area of the tissue

In Figure 6 it can be seen that the vast majority of slides contain very low percentages of cancerous tissue, as evidenced by the large spike near the 0–2.5% range. This suggests a significant imbalance, with most samples having negligible cancer content. A few slides contain higher percentages of cancer tissue (e.g., between 5–10% or beyond), but these are far less frequent. This indicates that cancerous regions occupy only a small fraction of the total tissue in most slides, which imposes a rather challenging task on the model to be able to distinguish between the two classes, when their representation, at least from the statistical point of view, does not vary significantly. At the morphological level, the model has to distinguish between the following structures of the tissue:

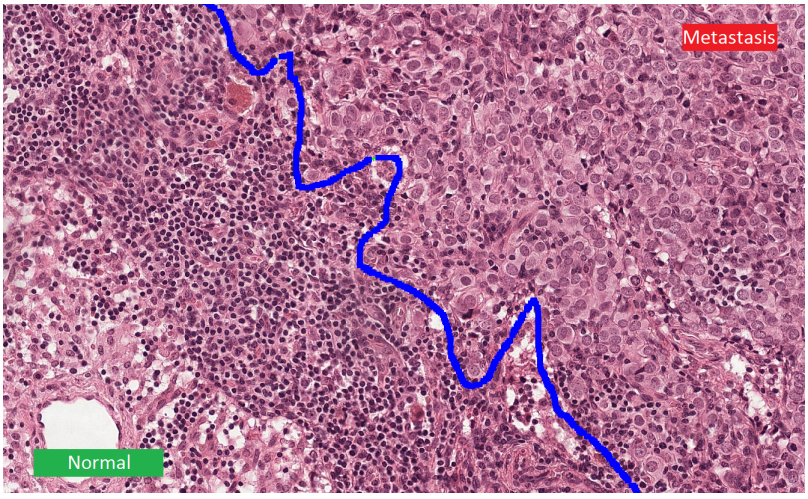

Figure 7: Comparison of the "normal" - negative and "metastasis" - positive, regions in the WSI. Source: `https://camelyon16.grand-challenge.org/Data/`

As seen in the above Figure 7., the task of determining whether a certain WSI contains a metastasis is not trivial and requires specific pathological knowledge to be able to

differentiate between the two. However when dealing with images of such high-resolution, and knowing that cancer regions are relatively small in the CAMELYON16 dataset, even an experienced pathologist could overlook such a region. This is where weakly-supervised machine learning methods are useful and able to learn even complex and non-trivial differences between the patch-level representation of different classes. Utilizing them can help understand in which parts of the tissues these regions usually occur and what is their morphological characteristics.

3.3 Feature extraction

For feature extraction CTransPAth (Wang et al. (2022))has been chosen (or suggested by the project's coordinator to be precise as a solution for workload reduction). It must be noted here that the features extracted from the CAMELYON16 with CTransPath method were made available for the project without the need to obtain them from scratch. It is however still important to describe the process in which they were obtained.

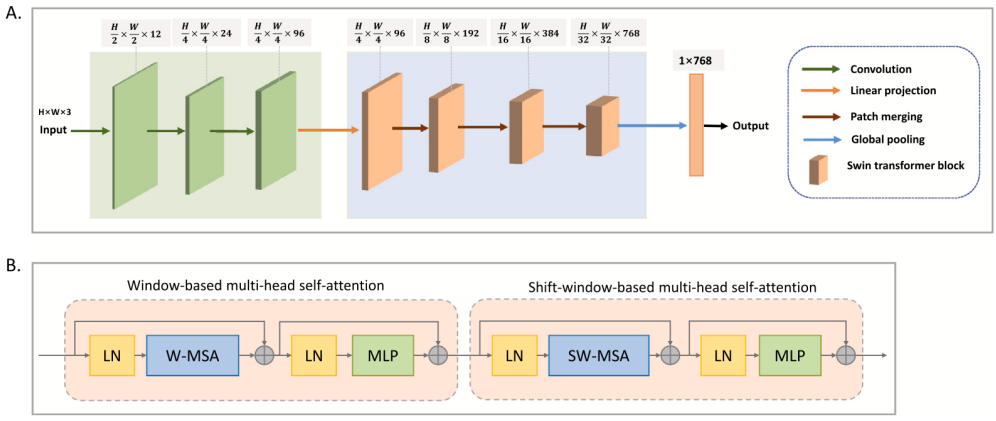

Figure 8: Architecture of the model used for feature extraction. Source: `https://pubmed.ncbi.nlm.nih.gov/35952419/`

As can be seen in the Figure 8., the feature extraction process involves the following:

- CNN backbone: Each patch is passed through CNN but to extract the low-level visual features, such as textures and shapes.

- Transformer Encoder: The feature map from the CNN is flattened and fed into a transformer encoder. The transformer captures long-range dependencies within the patch itself, allowing the model to understand global patterns inside the patch (e.g., structural relationships between different regions of the patch).

- Dimensionality Reduction: After passing through the transformer, the output is processed by dimensionality reduction layers (e.g., linear layers) to produce a fixed-length feature vector for each patch, ([1x768])

In the context of CTransPath and the CAMELYON16 dataset, the network processes one patch at a time as an independent unit. Each patch is fed into the network to extract its local feature representation. This means that while the model excels at identifying patch-level features (e.g., cellular or tissue structures), the direct dependencies or spatial relationships between adjacent patches are not explicitly modeled during the feature extraction process.

Extracted features come out of the feature extractor in the normalized form (centered around the zero value and squashed between the range from -1 to 1.). Shape of the output from the feature extractor vary depending on the input size of the bag (as in how many patches are in each), but for each patch the extracted sequence has the same length - 768. Thus the output for a single bag is of shape: [N, 1, 768], where N is the

number of patches in the bag. This representation is then resized to [N, 768], to get rid of the second, redundant dimension. As described above in 3.2, the model can handle inputs of varying size.

4 Baseline method and evaluation

4.1 Simple baseline model

As mentioned earlier, for this stage of the project, full TransMIL model has not yet been implemented. Instead, a simpler baseline model has been proposed - it still however preserves the main characteristics of the TransMIL model. In regular TransMIL, the model's forwards pass follows this specific algorithm, as described in the section 3.3 of the TransMIL paper, where MSA denotes Multi-head Self-attention layer, MLP denotes Multilayer Perceptron, and LN denotes Layer Norm:

Input: A bag of feature embeddings $\mathbf{H}_i = \{\boldsymbol{h}_{i,1}, \ldots, \boldsymbol{h}_{i,n}\}$, where $\boldsymbol{h}_{i,j} \in \mathbb{R}^{1\times d}$ is the embedding of the jth instance, $\mathbf{H}_i \in \mathbb{R}^{n\times d}$

Output: Bag-level predicted label $\hat{Y}_i$

1) Squaring of sequence;

$\sqrt{N} \leftarrow \lceil\sqrt{n}\rceil$, $M \leftarrow N - n$, $\mathbf{H}_S \leftarrow$ Concat $(\boldsymbol{h}_{i,class}, \mathbf{H}_i, (\boldsymbol{h}_{i,1}, \ldots, \boldsymbol{h}_{i,M}))$, where $\boldsymbol{h}_{i,class} \in \mathbb{R}^{1\times d}$ represents class token, $\mathbf{H}_S \in \mathbb{R}^{(N+1)\times d}$;

2) Correlation modelling of the sequence;

$\mathbf{H}_S^{\ell} \leftarrow$ MSA $(\mathbf{H}_S)$, where ℓ denotes the layer index of the Transformer, $\mathbf{H}_S^{\ell} \in \mathbb{R}^{(N+1)\times d}$;

3) Conditional position encoding and local information fusion;

$\mathbf{H}_S^{P} \leftarrow$ PPEG $\left(\mathbf{H}_S^{\ell}\right)$, where $\mathbf{H}_S^{P} \in \mathbb{R}^{(N+1)\times d}$;

4) Deep feature aggregation;

$\mathbf{H}_S^{\ell+1} \leftarrow$ MSA $\left(\mathbf{H}_S^{P}\right)$, where $\mathbf{H}_S^{\ell+1} \in \mathbb{R}^{(N+1)\times d}$;

5) Mapping of $\mathbb{T} \rightarrow \mathcal{Y}$;

$\hat{Y}_i \leftarrow \mathrm{MLP}\left(\mathrm{LN}\left(\left(\mathbf{H}_S^{\ell+1}\right)^{(0)}\right)\right)$, where $\left(\mathbf{H}_S^{\ell+1}\right)^{(0)} \in \mathbb{R}^{1\times d}$ represents class token.

At this stage, for simplicity, the implementation of the the PPEG (Pyramid Position Encoding Generator) module has been omitted. Without the PPEG module, the model operates without explicitly encoding patch position information. Instead, it relies only on the self-attention mechanism to learn the spatial relationships between patches. This can lead to a worse overall performance, since rejecting the iid assumption across patches in a bag and thus capturing their spatial relationship was one of the main reasons for proposing the TransMIL method and thus underlining the importance of the spatial relationships. However a model with self-attention mechanism should still be able to better capture the spatial relationships between patches than a model that follows the iid assumption, thus the proposed simplified baseline is somewhat of a compromise between the two approaches.

Simplified model omits the "sequence squaring" step, as it is only necessary for dynamic adjustment of the sequences' shape as an input to PPEG. Below is the architecture of the simplified model:

Let the input be a batch of n instances, each of d - 768 features, with a batch size of B. The input tensor is denoted as:

$$\mathbf{X} \in \mathbb{R}^{B\times n\times d}$$

1. **Linear Transformation 1:** The input is passed through the first fully connected layer `fc1` to adjust to the new feature dimension if provided:

$$\mathbf{X} = \texttt{fc1}(\mathbf{X}) \in \mathbb{R}^{B\times n\times new\_feature\_dimension}$$

2. **Class Token Addition:** A learnable class token $\mathbf{h}_{cls} \in \mathbb{R}^{1\times 1\times new\_feature\_dimension}$ is expanded across the batch and concatenated with the input instances:

$$\mathbf{X}_{cls} = \texttt{cls\_token}.expand(B, 1, -1)$$

$$\mathbf{X} = \texttt{concat}(\mathbf{X}_{cls}, \mathbf{X}, \mathrm{dim} = 1) \in \mathbb{R}^{B\times(n+1)\times new\_feature\_dimension}$$

3. **Attention Layer 1:** The input goes through the first multi-head attention layer (`at1`):

$$\mathbf{X}, = \texttt{at1}(\mathbf{X}, \mathbf{X}, \mathbf{X})$$

4. **Attention Layer 2:** The output of the first attention layer is passed through the second attention layer (`at2`):

$$\mathbf{X}, = \texttt{at2}(\mathbf{X}, \mathbf{X}, \mathbf{X})$$

5. **Class Token Extraction:** The first token of the output sequence is selected (i.e., the class token):

$$\mathbf{h}_{cls} = \mathbf{X}[:, 0] \in \mathbb{R}^{B \times new\_feature\_dimension}$$

6. **Layer Normalization and Output:** The class token is normalized using layer normalization, then passed through the second fully connected layer (`fc2`) to produce the output predictions:

$$\mathbf{h}_{cls} = \texttt{ln}(\mathbf{h}_{cls}) \in \mathbb{R}^{B \times new\_feature\_dimension}$$

$$\hat{Y} = \texttt{fc2}(\mathbf{h}_{cls}) \in \mathbb{R}^{B \times num\_classes}$$

Thus, the final output is the bag-level predicted label:

$$\hat{Y}_i = \texttt{MLP}(\texttt{LN}((\mathbf{h}_{cls})^{(0)}))$$

In training and validation, the cross-entropy loss function has been used, same as proposed by the TransMIL authors. It is also worth noting that at this stage the self-attention layers are not using the Nystrom (Xiong et al. (2021)) softmax function as proposed in the TransMIL, so for now the computational complexity of this part is $O(n^2)$, instead of reachable $O(n)$.

4.2 Model evaluation

To evaluate the model's predictions, it is essential to consider the specific types of errors the model should prioritize avoiding. In the context of binary cancer classification, false negative errors are critical. This is because misclassifying a cancerous patient as healthy is far more detrimental than incorrectly labeling a healthy patient as having cancer. Failure to detect cancer could lead to untreated cancer, which very negatively impacts the patient's health. False positive error may result in unnecessary cancer-specific treatment, which, although unpleasant, does not carry the same life-threatening consequences.

Acknowledging this allows for the choice of the specific metrics, that will measure the rate of both false positive and false negative errors. One such metric is AUC (area under curve), which provides an aggregate measure of the model's ability to distinguish between the positive and negative classes across different thresholds by plotting of the True Positive Rate (Recall) against the False Positive Rate at various thresholds.AUC is particularly useful in imbalanced datasets, as it focuses on the model's ranking ability rather than raw prediction accuracy. At this stage the AUC is not yet implemented for the code experiments.

In TransMIL in addition to AUC, the accuracy metric has been used, which could not serve as a standalone metric, due to its inability of adequately represent the class imbalance in the classification task on the imbalanced dataset. However since the data is not heavily imbalanced, it could still serve as an indicator.

4.3 Training and results

As described in the introduction to the report, at this stage, even the simplified baseline model has not yet been trained on the full CAMELYON16 dataset. The purpose of developing the simpler model (described in 3.1) was to become familiar with most parts of the proposed TransMIL model and to be able to successfuly implement it in the code and make sure that the data can run through it without errors.

To check the correctness of the proposed model and its ability to at least minimize the loss over the training progress, a mini split for the CAMELYON16 dataset has been proposed - consisting of 17 data points (WSIs), 11 in the training subset, 3 in the validation subset and 3 in the training set. Below are the results of the training and validation with 10 epochs:

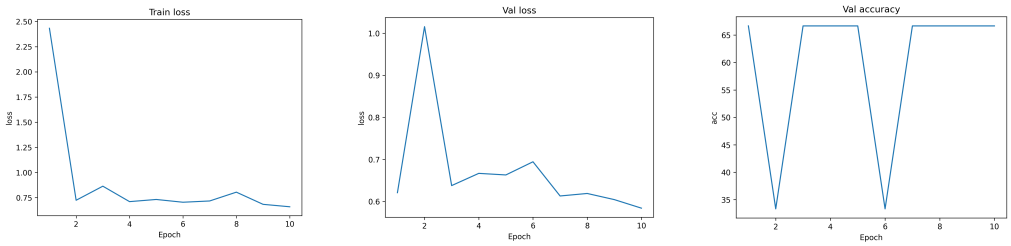

Figure 9: Training loss and evaluation loss and accuracy for training on the mini split of CAMELYON16 dataset

The purpose of this task was mostly to find out whether the model is able to overfit on the training dataset, thus to see if it's capable of minimizing the loss function and memorize the training data. It is able to generalize on the validation dataset, but is not able to generalize on the test set, reaching an accuracy of 0% on 3 samples. However the key takeaway from this mini-training is the fact that the model is able to decrease the loss, thus the training is in some way progressing.

To avoid overfitting in the future stages of the model development and training, one should optimally tune the hyperparameters, specifically the numbers of heads used in the Multi-head Self-attention layers, as well as the dropout rate in those very layers, not to mention the amount of the training epochs. Early stopping should also be implemented for this purpose.

5 Discussion

The dataset imposes several challenges. Firstly, as mentioned before, the amount of actual cells or in practice pixels containing cancer regions (metastasis) is very low compared to the overall amount of the data, which means that a high attention of the model must be drawn to relatively small regions. Secondly, distinction between the metastasis and not-metastasis regions is not trivial and requires the model to learn complex representations of the positive (tumor) class. Also, dealing with the images in a very high-resolutions is also challenging and requires splitting the original WSI into smaller patches, which is a rather complex preprocessing task.

At this stage of the project it is not yet possible to specify whether the model is able to capture the spatial and morphological information in the input tissue data and draw attention to the desired regions containing cancer cells. This is because the mini-split version of the dataset has not been designed to be representative of the full dataset, but rather to check the model's ability to minimize the training loss. It would however be interesting to see how the simple baseline model performs on the full CAMELYON16 dataset and compare these results with the regular TransMIL model's performance on the full dataset. This will be done in the second stage of the project.

The CAMELYON16 dataset is particularly suitable for real-world medical applications because models trained on it address a critical task in healthcare: the identification of cancer in histopathological tissue samples. The dataset is designed in a manner that the majority of samples (WSIs) contain only small cancerous regions relative to the overall tissue size. This aligns with medical knowledge, as smaller areas of cancerous tissue typically indicate early-stage tumors. Early detection of metastases is crucial for effective treatment, as cancer in its early stages has not yet spread extensively through the tissue or to distant organs.

However, detecting cancer at this stage poses significant challenges due to the limited size of cancerous regions. Even expert pathologists can find it difficult to identify such small regions, which increases the likelihood of misclassification. As described in the CAMELYON16 paper: "Accurate breast cancer staging is an essential task performed by pathologists worldwide to inform clinical management. Assessing the extent of cancer spread by histopathological analysis of sentinel axillary lymph nodes (SLNs) is an important part of breast cancer staging. The sensitivity of SLN assessment by pathologists, however, is not optimal. A retrospective study showed that pathology review by experts changed the nodal status in 24% of patients. Furthermore, SLN assessment is tedious and time-consuming. It has been shown that deep learning algorithms could identify metastases in SLN slides with 100% sensitivity, whereas 40% of the slides without metastases could be identified as such." This indicates that not only such ML models could outperform the human error on this task, but also be applied to the crucial, early stage identification of the spread of the disease and not only the stages, in which the presence of the metastasis in the node is more noticeable and obvious due to its size, which is more characteristic for late cancer stages.

References

B. Ehteshami Bejnordi, M. Veta, P. Johannes van Diest, B. van Ginneken, N. Karssemeijer, G. Litjens, J. A. W. M. van der Laak, , and the CAMELYON16 Consortium. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA*, 318(22):2199–2210, 12 2017. ISSN 0098-7484. doi: 10.1001/jama.2017.14585. URL https://doi.org/10.1001/jama.2017.14585.

Z. Shao, H. Bian, Y. Chen, Y. Wang, J. Zhang, X. Ji, and Y. Zhang. Transmil: Transformer based correlated multiple instance learning for whole slide image classication. *CoRR*, abs/2106.00908, 2021. URL https://arxiv.org/abs/2106.00908.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

X. Wang, S. Yang, J. Zhang, M. Wang, J. Zhang, W. Yang, J. Huang, and X. Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical image analysis*, 81:102559, 2022. URL https://api.semanticscholar.org/CorpusID:251207603.

Y. Xiong, Z. Zeng, R. Chakraborty, M. Tan, G. Fung, Y. Li, and V. Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention, 2021. URL https://arxiv.org/abs/2102.03902.