Project Machine Learning
— Milestone 3 —

Ricardo Fleck, Hassan Bassiouny, Augustin Krause

February 2, 2024

1 Introduction

During the third milestone of the "Project Machine Learning" we advanced our project in three major ways:
First, we focused on an additional dataset (called "`TCGA`"), which consists of whole-slide images (WSIs) of human tissue from the colon. This was done to test our methods on a similar dataset as the histopathological ones from Ilse et al. (2018). Nonetheless, since the `TCGA`-dataset stems from a different source as theirs, a comparison remains difficult. Our methodology and final results for this dataset are discussed in Section 2.
To better assess the applicability of our models to real-world problems, we defined a measure to judge the confidence of our models, which is described in detail in Section 3.
Lastly, we also performed a sensitivity analysis akin to the a method described in Montavon et al. (2018a). The results of this, as well as the results from the confidence measure, are analysed in Section 4.

2 The Cancer Genome Atlas Dataset

2.1 Dataset

"The Cancer Genome Atlas" Program[1] is a research project that started in 2006, which attempts to enhance our understanding of cancer genomics. Amongst other contributions it resulted in the `TCGA` dataset, which in total comprises over 2.5 petabytes of data.
However, we opted for a subset consisting of 472 WSIs (Whole-Slide Images) of cancerous human tissue from the colon. These images were very large (in our case on average about 0.97GB) and already divided into smaller "patches". Each subdivided slide constitutes one bag.
We also had metadata, which consists of several attributes per slide that we analyzed to define a positive (label 1) and a negative (label 0) category within the data. The choice of label is a crucial one and a poor choice can turn into a strongly compounding factor. During investigation of the data, as well as testing of our models, we additionally stumbled onto several other factors complicating the training. These will be outlined in the next subsection.

2.2 Methodology

Feature Extraction The first step we took was to extract features from the patches because of the large number of instances within the bags (on average about 11000) and the still large pixel-sizes of the patches. For this we used a pre-implemented version of the "ctranspath"-model (Wang et al. (2022)) which extracted 768 features from each of the patches. As we shall see later, many of the final models we obtained

[1] https://www.cancer.gov/ccg/research/genome-sequencing/tcga

learned to predict always the same label and one contributing factor toward this could be that we did not fine-tune the feature-extraction method to the problem at hand.
We chose the same model architecture as for the classical MIL datasets. We made that choice because this architecture was already intended to work on pre-extracted features. In contrast, the architecture suggested in the paper is supposed to work on actual histopathology images Ilse et al. (2018).

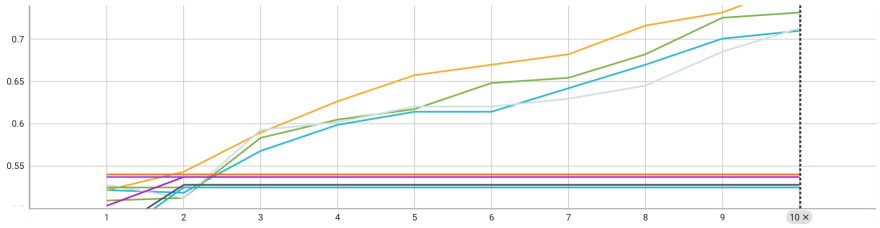
Problem Definition Our labels were defined based on the "ajcc_pathologic_t" attribute which indicates the stage of the cancer detected within the WSI. We ended up trying to distinguish Stage 3 and 4 cancers from the rest of the data, because they make up about 50% of the slides and we therefore do not run into problems associated with class-imbalance.
The first models we obtained learned a "trivial solution", exclusively predicting a constant label. One potential reason for this could be that our initial data loader exacerbated some fundamental flaws within the data:
The different tissue scans stem from a selection of different hospitals and therefore have different "stainings"/colourings. Therefore the images from different hospitals can live within very different color spaces. Our data loader initially sorted by the "case_id" column of the metadata to more easily make sure that slides stemming from the same case (i.e., the same patient) end up in the same partition of the dataset. However, by sorting by case id, we implicitly also sorted by hospital source. Therefore it was likely that the training data came in large part from a very different source than the test data. Ultimately, we simply took a single slide per case id to make sure no case ends up in the test and in the train set at the same time. We also shuffled the data before training to ensure a homogeneous distribution of the different data sources of the slides.

Model-Selection The hyper-parameter selection remained mostly unchanged from the previous milestone. However, because we noticed that training took longer than on the other datasets, we significantly reduced the size of the grid of hyper-parameters we evaluated, which can be seen from Table 1. We chose 10 as the only number of epochs because training over 100 epochs took too long for our time-constraints. Another major difference was that we excluded unnecessary hyper-parameter combinations from being tested in the cross-validation (CV). We also gained a significant speed up in training time by writing the data to the cluster's `/tmp` directory, which consists of fast SSD storage intended for temporary use.
Lastly, we also investigated the development of the training loss and accuracy over the epochs to see whether our models were actually training. Some exemplary (and representative) curves can be seen in Figure 1, which show the training curves for some of the CV iterations of the `TCGA` embedding based attention model.
From these plots, one can observe that during some of the runs the model actually

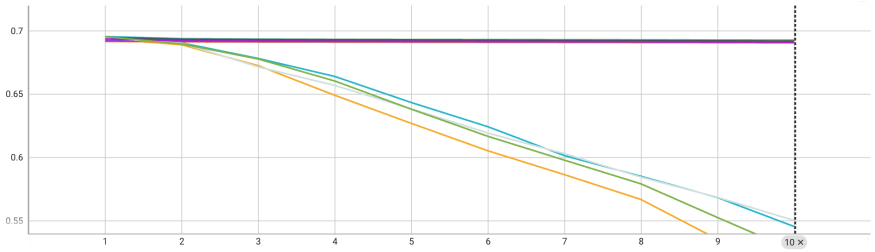
Table 1: The adjusted hyper-parameter grid we tested.

	Tested Values
Learning Rate	0.001, 0.01
Weight Decay	0, 0.005
Momentum	0, 0.09, 0.9
Optimizer	Stochastic Gradient Descent, Adam

trains, while in others, it immediately gets stuck. The ones were it trains were early hyper-parameter combinations, which used the smallest possible values for the learning rate, weight decay and momentum. The ones where it gets stuck are ones where we used larger values for these hyper-parameters. This suggests that our chosen hyper-parameter grid tested for values that were too large for the problem setting and a more optimized grid could lead to significant improvements.



(a) Development of the training accuracy of the `TCGA` embedding based attention model.



(b) Development of the training loss of the `TCGA` embedding based attention model.

Figure 1: Accuracy and training loss over 10 epochs for different runs of CV and different hyper-parameter combinations.

2.3 Test Results

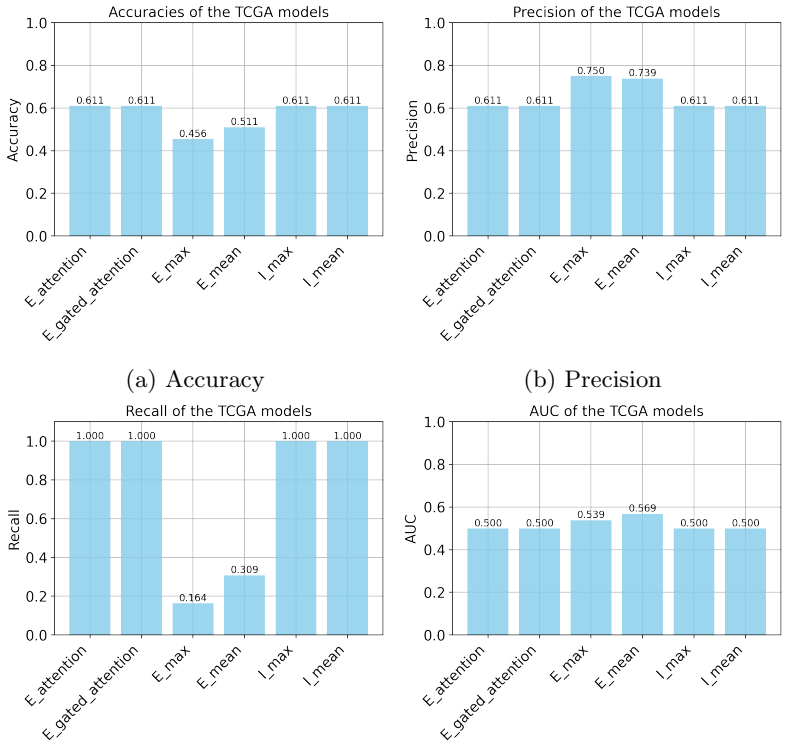
The evaluation results of the models selected by the CV can be seen in Figure 2. Even with this improved set-up, four of the six models learned a "trivial solution", which is visible from Subfigure 3c. Because we saw the training loss decrease and the training accuracy increase during some of the CV iterations (see Figure 1) we wanted to try training with a different order of the shuffling. For this, we changed the random seed and gave it another try. These new results can be seen in Figure 3 These models were able to not just learn a trivial solution. Even though now the accuracies are slightly decreased for most models, this comes at the benefit of increased precision and AUC scores. Most of the models from random seed 0 were never able to improve upon the minimum possible AUC score of 0.5. The higher AUC score for the second random seed is indicative of more sophisticated models. We therefore have reason to believe that with further optimizations to the model architecture, the feature extraction and the model-selection procedure additional improvements are attainable.

3 Confidence Measure

Both the `MNIST-bags` model and the classical MIL model take bags of instances as input and return a value $y \in [0, 1]$. The model prediction is the rounded output $\lfloor y \rceil \in \{0, 1\}$. This allows us to interpret the output of the model as the probability it assigns to a bag being positive. We will build our confidence measure upon this interpretation of the model output above. Clearly, the model is less confident the closer the output is to 0.5 and more confident for outputs close to 0 or 1. This leads us to the following idea for a confidence measure:

$$\hat{\phi}(y) = \begin{cases} y & \text{if } \lfloor y \rceil = 1, \\ 1 - y & \text{if } \lfloor y \rceil = 0 \end{cases}$$

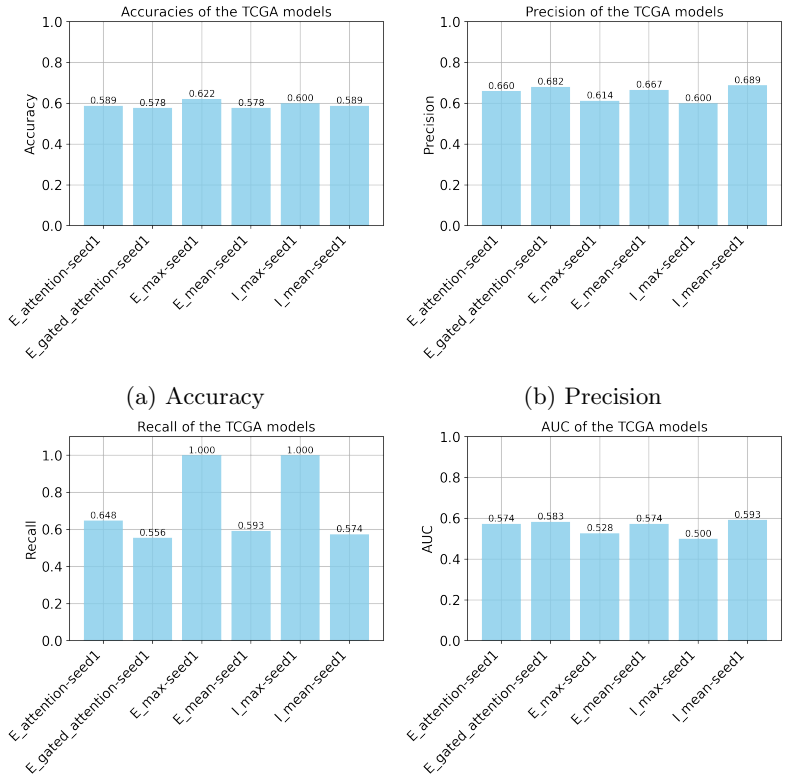
There is a clear problem with this confidence measure, since its image is restricted to $[0.5, 1]$. Thus, we must recast this interval through a monotonically increasing function to $[0, 1]$ in order to obtain the desired confidence measure. We chose the linear function $f(x) = 2x - 1$. This minimally manipulates the values obtained from our original idea for a confidence measure and preserves the relation between confidence measures of different samples. Finally, we arrive at the confidence measure we will use for the evaluation of our model: $\phi(y) = (f \circ \hat{\phi})(y)$.



(a) Accuracy (b) Precision

(c) Recall (d) ROC AUC score

Figure 2: The achieved scores of the `TCGA` models on the different metrics (using random seed 0).



(a) Accuracy (b) Precision

(c) Recall (d) ROC AUC score

Figure 3: The achieved scores of the `TCGA` models on the different metrics (using random seed 1).

4 Evaluation

4.1 Predictions

Regarding the results, we will focus on the two attention mechanisms, since these are the focus of the paper. The Tables below illustrate the confusion matrix for various datasets. For each possible outcome (True Positive (TP), False Positive (FP), True Negative(TN), False Negative(FN)), a cell contains the count of occurrences and the model's average confidence measure for that specific occurrence. Ideally, we want the model to exhibit high confidence for TP and TN (highlighted in green) and conversely, low confidence for FP and FN (highlighted in red).

Notably, all models showcase the highest average confidence measures for True Positives, indicating that the models are confident when predicting positive bags. It is worth noting that the `MNIST-bags` model 1 as a confidence measure for all its predictions, which may indicate overconfidence. Thus, there is a need for confidence calibration to achieve a more meaningful confidence measure in the `MNIST-bags` case. The model for the `TCGA` dataset showcases the lowest average confidence measures compared to the other models. Nevertheless, it demonstrates higher confidence levels for correct predictions than for incorrect ones.

In conclusion, the confidence measure proves meaningful for most datasets, as correct predictions, on average, boast higher confidence than incorrect ones. The only exceptions are the FN for `TIGER` and `MNIST-bags`.

MUSK1

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	9 0.92	1 0.08
	Negative	2 0.34	7 0.89

MUSK2

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	7 0.99	1 0.73
	Negative	4 0.87	9 1.0

ELEPHANT

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	20 0.87	0 -
	Negative	1 0.42	19 0.81

TIGER

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	18 0.94	5 0.84
	Negative	3 0.98	14 0.97

FOX

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	14 0.99	10 0.86
	Negative	7 0.93	9 0.95

MNIST

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	95 1.0	0 -
	Negative	5 1.0	100 1.0

TCGA

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	0.43	0.2
	Negative	0.31	0.51

4.2 Time-Saving Benefits

In this section, we will explore the potential time savings achievable by utilizing various MIL predictor models across different datasets. Considering the confidence levels of these models, we will compare the time it takes for a human to make a prediction, denoted as x seconds, versus the instantaneous prediction capability of the model. Our analysis will delve into the degree of automation facilitated by these models in the prediction process.

Method of using the Model for Time-Saving In this section, we will propose an approach to leverage the model for time-saving benefits. Below are the steps outlining how to utilize the model effectively.

1. For each new sample the model predicts a label along with a confidence.
2. If the confidence exceeds a threshold, we accept the prediction without human intervention. If the prediction is correct, time is saved; if it is incorrect, an error is made.
3. If the model lacks confidence, a human user reviews the prediction and corrects it if necessary. If the prediction is correct, human time is wasted; if it is incorrect, an error is avoided.

Analysis of the Confidence Measure To assess the trade-off between the time-saving benefits and the error rate we will analyze the confidence for the following thresholds: **0.1, 0.2, 0.4, 0.6, 0.8, 0.9**.
We say a model's prediction on a bag has a "positive confidence" if the confidence exceeds the threshold, otherwise it has a "negative confidence". To evaluate the reliability of the confidence, we will examine the following metrics.

- **Confidence Ratio**

  This metric demonstrates the number of bags that can be automatically predicted. It signifies the percentage of data where the model had a positive confidence: $\frac{\#\text{ Positive Confidence bags}}{\#\text{ Total number of bags}}$

- **Positive Confidence Accuracy**

  This metric reflects the prediction *accuracy* of the model when it expresses confidence in its predictions. It delineates the number of data instances where the model's predictions can be automated without errors. Analyzing this metric provides an estimation of the potential errors in the automation process: $\frac{\#\text{ Positive Confidence for true prediction}}{\#\text{ Positive Confidence for true prediction} + \#\text{ Positive Confidence for false prediction}}$

- **Negative Confidence Accuracy**

  This metric indicates the *inaccuracy* of the model's predictions when it lacks confidence. It reveals the instances where human intervention was necessary to achieve correct predictions. A higher value in this metric implies fewer errors and less wasted human intervention time on unnecessary predictions: $\frac{\#\text{ Negative Confidence for false prediction}}{\#\text{ Negative Confidence for true prediction} + \#\text{ Negative Confidence for false prediction}}$

Analysis of the Results

MUSK1 Table 3 illustrates the evaluation results of model confidence on the `MUSK1` dataset using the attention method across various thresholds. As depicted in the table, a higher threshold corresponds to a lower confidence ratio, indicating fewer instances that can be automated without human intervention. It reaches its peak value at 94.7%. At thresholds 0.1 and 0.2, positive confidence accuracy reaches approximately 88.9%. Conversely, as the threshold increases, negative confidence accuracy decreases, implying a reduction in unnecessary human intervention for correctly predicted labels. The optimal tradeoff is observed at threshold 0.4, where both positive and negative confidence accuracy reach 100%, leading to efficient work allocation with also having 0 errors. At this threshold, 84.2% of instances can be automated, resulting in an 84.2% time-saving utilization of the model.

FOX Table 4 reveals intriguing findings from the confidence measure metrics applied to the `FOX` dataset using the gated attention method. Notably, the confidence ratio demonstrates a sharp decline as the threshold increases, plummeting to a minimum value of 17.5% at 0.9 compared to its maximum of 82.5% at 0.1. This indicates a reduction in the number of instances amenable to automation with higher thresholds. However, increasing the confidence threshold does not necessarily correspond to a decrease in error rates, highlighting potential unreliability in the confidence measure. The lowest error rate observed is 31.8% at a threshold of 0.4, accompanied by a positive confidence accuracy of 68.2%. Nonetheless, this entails unnecessary human intervention for correctly predicted instances in half of the cases. These metrics suggest that only 55% of time can be saved under these conditions.

Table 3: Confidence Measure Metrics for `MUSK1` with Attention Method

Threshold	Confidence Ratio	Positive Confidence Accuracy	Negative Confidence Accuracy
0.1	0.947	0.889	1.000
0.2	0.947	0.889	1.000
0.4	0.842	1.000	1.000
0.6	0.737	1.000	0.600
0.8	0.684	1.000	0.500
0.9	0.632	1.000	0.429

Table 4: Confidence Measure Metrics for `FOX` with Gated Attention Method

Threshold	Confidence Ratio	Positive Confidence Accuracy	Negative Confidence Accuracy
0.1	0.825	0.576	0.286
0.2	0.825	0.576	0.286
0.4	0.550	0.682	0.500
0.6	0.350	0.571	0.385
0.8	0.250	0.400	0.333
0.9	0.175	0.286	0.333

MNIST-bags Table 5 presents a set of results from the confidence metrics applied to the `MNIST-bags` dataset using the gated attention method. At low thresholds such as 0.1 and 0.2, the model exhibited complete confidence in all predictions, allowing for 100% automation and thereby saving 100% of the time. Additionally, it achieved a Positive Confidence Accuracy of 98%, indicating a mere 2% error rate. These findings underscore the model's proficiency in predicting the `MNIST-bags` dataset and highlight the reliability of the confidence measure. Further analysis of higher thresholds did not significantly alter the already impressive metrics.

TCGA The Table 6 presents findings from the evaluation of the `TCGA` histopathology dataset using the gated attention method. Initially, it is evident that positive confidence accuracy improves as the threshold increases, resulting in fewer errors. However, this improvement is accompanied by a decrease in the confidence ratio, indicating a reduction in the number of instances suitable for automation. Notably, at threshold 0.4, a favorable tradeoff emerges, boasting a positive confidence accuracy of 87.2% with only a 12.8% error rate, while also enabling automation and time-saving in 43% of cases. However, this comes at the expense of a negative confidence accuracy of 43%, indicating additional unnecessary human intervention in 57% of bags. Choosing the optimal threshold hinges on balancing the desire to automate bags, with the tolerance for errors that can be accepted.

4.3 Explainability of the Model's Decisions

In various domains of machine learning, understanding the rationale behind a model's decisions allows for anticipation and potential correction of its behavior. Numerous methodologies have been proposed in the literature to enhance model interpretability. Among these, Sensitivity Analysis stands out as a particularly intuitive approach, offering insights into the model's behavior and decision-making processes. Analyzing the attention values offered by the attention methods can as well offer a lot of information about the different regions of interest (ROI).

Table 5: Confidence Measure Metrics for `MNIST-bags` with Gated Attention Method

Threshold	Confidence Ratio	Positive Confidence Accuracy	Negative Confidence Accuracy
0.1	1.000	0.980	-
0.2	1.000	0.980	-
0.4	1.000	0.980	-
0.6	1.000	0.980	-
0.8	0.990	0.985	0.500
0.9	0.990	0.985	0.500

Table 6: Confidence Measure Metrics for `TCGA` with Gated Attention Method

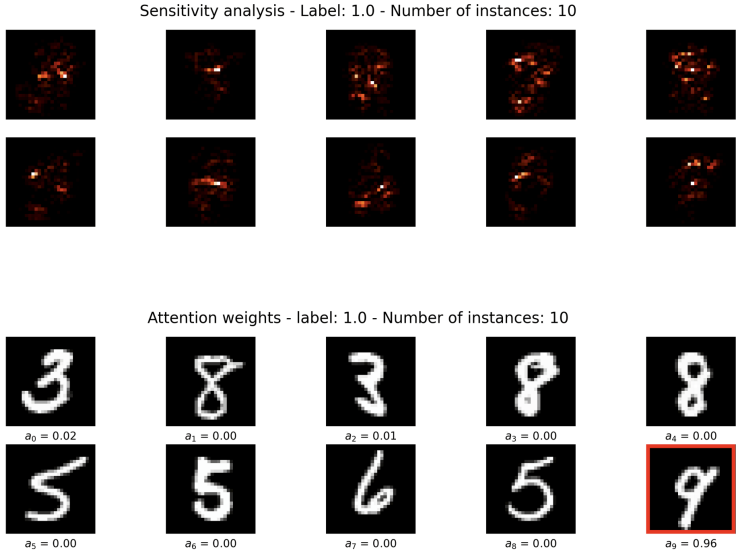
Threshold	Confidence Ratio	Positive Confidence Accuracy	Negative Confidence Accuracy
0.1	0.833	0.733	0.467
0.2	0.722	0.785	0.520
0.4	0.433	0.872	0.431
0.6	0.300	0.963	0.413
0.8	0.144	1.000	0.351
0.9	0.089	1.000	0.329

Sensitivity Analysis As mentioned in Montavon et al. (2018b), Sensitivity Analysis is an approach to identify the most important input features. It is based on the model's locally evaluated gradient or some other local measure of variation. A common formulation of sensitivity analysis defines relevance scores $R_i(x) = \frac{\partial f}{\partial x_i}$, which are calculated using the model's output f and the input feature x_i.

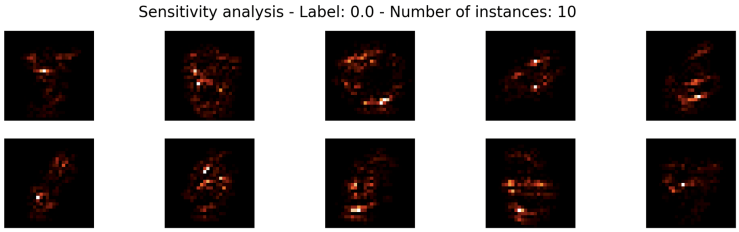
The computation itself returns a vector with the same size as the input, and it is easily computed with backpropagation methods. The gradient indicates how sensitive the output is to the particular input feature, and parts where the sensitivity is high can indicate importance in those regions.

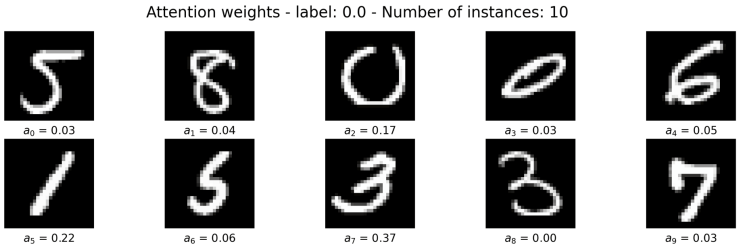
Attention Values The attention and gated mechanism learn to allocate varying degrees of attention to different inputs within the bags based on their importance. Each input in the bag is assigned an attention value, and the sum of all attention values amounts to 1. By extracting these attention values, one can conduct further analysis to discern regions of significance, often referred to as ROIs, as described in Ilse et al. (2018).

Explainability Results We conducted a thorough assessment of sensitivity analysis and examined the attention values for the `MNIST-bags` dataset. Our findings indicate that the `MNIST-bags` dataset offers the most intuitive understanding, as its data consists of known images of numbers that can be plotted and readily interpreted. In contrast, the classical MIL datasets and the `TCGA` dataset primarily enter our models in terms of feature vectors, presenting a higher level of abstraction and making interpretation less straightforward.

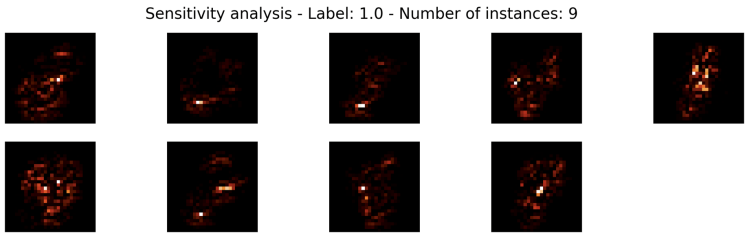


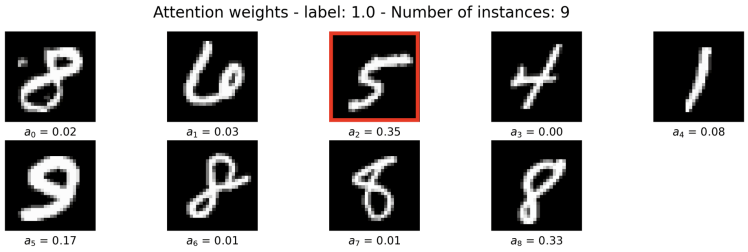
(a) Example of a positive bag.





(b) Example of a negative bag.





(c) Example of a positive bag with sub-optimal attention values.

Figure 4: Results of analyzing the sensitivity and the attention values on the `MNIST-bags` dataset

In Figure 4, we observe various plots analyzing the attention mechanism across different instances of bags within the `MNIST-bags` dataset. Upon initial inspection, the sensitivity appears intuitive, as regions devoid of any digits or drawings are depicted as completely black, signifying insensitivity. Conversely, digits, particularly their curves and edges, are highlighted in red, indicating sensitivity and illustrating how the model incorporates this information to identify the digit.
Additionally, the plots display the distribution of attention values, which within a positive bag could help identify the instances with a positive label. In the case of the `TCGA` histopathology dataset these instances could be cancerous cells.
In most instances within the `MNIST-bags` dataset, we observed that the highest attention value was allocated to the target number. However, there were cases, such as shown in Figure 4c, where attention was directed towards an irrelevant number, such as the digit 5, despite the target digit being 9. Nonetheless, the target digit still received some attention (0.17), which could have been adequate for the model to correctly classify it as the number 9.

4.4 Feasibility of the Automation Process

Through our analysis of different confidence thresholds we showed the automation process offers significant advantages in terms of error rate and time-saving benefits depending on what is required by the problem setting. It is advisable to explore various thresholds for confidence, coupled with relevant metrics, to evaluate error rates and time saved comprehensively. Analyzing and fine-tuning the threshold can enhance the method's applicability to scenarios with distinct objectives. Additionally, assessing the reliability and meaningfulness of the confidence measure, as done in Section 4.2, remains crucial.

5 Conclusion

In this section, we aim to provide a concluding assessment of the paper, outlining suggestions for enhancing the overall setup and unexpected discoveries made during our analysis of the paper and its methodologies.
First of, the prediction set-up working on the `TCGA`-dataset could be improved in several ways. This could involve choosing a more suitable architecture for the problem at hand. One could also include the feature-extraction model in the training and prediction pipelines, in order for it to get fine-tuned to the characteristics of the dataset and the learning problem. Further optimization could be done on the hyper-parameter selection, e.g., by evaluating a better hyper-parameter grid. This could also apply to the models working on the classical MIL datasets and the `MNIST-bags` dataset. Additionally, one could also augment the patch data. Lastly, shuffling the data and running the experiments on different random seeds could yield a more thorough assessment of the strengths of the models for all set-ups.

Based on our interpretation of the model output, we have derived a meaningful confidence measure for all datasets except `MNIST-bags`. However, this measure can be improved upon by performing confidence calibration. Specifically in the field of computer vision, Gao et al. (2022) propose multiple methods for confidence calibration. This would be especially necessary for the `MNIST-bags` model.

In comparison with the paper by Ilse et al. (2018) we made a few interesting discoveries. Firstly, deep neural networks in conjunction with the MIL setting do not seem to be very suitable for the classical MIL datasets. This, we think, is mainly due to the small dataset sizes. While the original authors did not claim their methods were much more suitable than previous methods for these datasets, this lack of applicability was never highlighted in the paper.
The paper mentions data augmentation on the histopathology dataset without specifying the method. On top of this, the original histopathology data referenced, in the paper is no longer accessible on the website specified, rendering it impractical to replicate the results using the same dataset.
The paper only examined a limited number of model variations, focusing solely on a specific model trained with specific parameters. Consequently, there is a lack of information regarding the effectiveness of the attention mechanism across different model sizes.

References

R. Gao, T. Li, Y. Tang, Z. Xu, M. Kammer, S. L. Antic, K. Sandler, F. Moldonado, T. A. Lasko, and B. Landman. A comparative study of confidence calibration in deep learning: From computer vision to medical imaging, 2022.

M. Ilse, J. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018. URL `https://arxiv.org/abs/1802.04712`.

G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15, 2018a.

G. Montavon, W. Samek, and K.-R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018b. ISSN 1051-2004. doi: https://doi.org/10.1016/j.dsp.2017.10.011. URL https://www.sciencedirect.com/science/article/pii/S1051200417302385.

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. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102559. URL https://www.sciencedirect.com/science/article/pii/S1361841522002043.