# Optimizing Deep Learning Model Labeling Costs

Akshat Gupta, Yangyuan Zhang, Zhen Yang

## Objective

The aim of this project is to optimize labelling costs of medical images in deep learning models using advanced algorithms that utilize the intrinsic properties of the available data to improve the performance of the model. This will be achieved through a multi-faceted approach leveraging semi-supervised learning, active learning, and their integration, thus addressing the time-intensive nature of accurate labeling and the need for vast amounts of labeled data. Specifically, we will utilize semi-supervised learning to leverage the structure of unlabeled data, augmenting labeled datasets. Additionally, we will employ active learning techniques to select the most informative instances for annotation, optimizing labeling resources. Finally, we will develop a framework for semi-supervised active learning, combining both methodologies to achieve superior performance while minimizing labeling costs. We also aim to compare the performance of these algorithms with the traditional supervised learning paradigm to demonstrate the effectiveness of our work.

## Background and Significance

Human solid tumors present complex structures containing diverse tissue types, including clonal tumor cells, tumor stroma, immune cell infiltration etc. Distinguishing between these tissue types is feasible through histopathological examination of tissue sections stained with Hematoxylin and Eosin (H&E) [1].

Histopathological images provide valuable insights into disease analysis, but traditional pathology diagnosis relies heavily on the subjective judgment of trained pathologists. To address this, there's a

growing demand for computer-assisted diagnosis, particularly with the rise of high-throughput tissue banks. Machine learning techniques, such as deep learning methods, offer promising avenues for identifying benign areas and allowing pathologists to focus on complex cases [2].

Colorectal cancer stands as the second leading cause of cancer-related deaths, according to the American Cancer Society [3]. Tumor progression strongly impacts patient prognosis, with research indicating significant morphological changes in various components of the tumor stroma during tumor development [4]. Accurately quantifying tissue composition in colorectal cancer is thus crucial in histopathology for delineating disease characteristics and predicting patient outcomes.

On the other hand, deep learning methods, characterized by their use of multi-layer neural networks, have achieved remarkable success in areas like medical image classification. Specifically, Convolutional Neural Networks (CNNs) [5, 6] and Vision Transformers (ViTs) [7] have emerged as the backbone of image classification tasks. However, in traditional supervised learning scenarios, a large number of labels are required to ensure strong performance. Thus, the advantage gained from using larger datasets can be offset by the significant cost associated with labeling data, especially when expert knowledge is required, as is often the case in medical applications like histopathological imaging.

A compelling solution to this challenge is semi-supervised learning (SSL) [8, 9, 10], which enables training models on expansive data collections without an extensive labeling effort. SSL achieves this by harnessing unlabeled data, which can be acquired with relatively minimal human involvement. With the help of SSL, the state-of-the-art deep learning model can efficiently learn how to capture high-level features, achieving satisfying classification accuracy with a minimal subset of labeled data, reducing the labeling cost drastically. For example, as for image classification tasks on the CIFAR-10 dataset, the best semi-supervised learning approach can achieve a performance of ~95% accuracy using only 40 labeled instances [8].

Also, traditional active learning (AL) methods, without incorporating semi-supervised or self-supervised strategies, have historically demonstrated superior performance in terms of minimizing labeling efforts [11, 12, 13]. However, they are outperformed by recent SSL methods in terms of labeling budget. A natural idea is to apply active learning within the frame of semi-supervised learning, and there are already some efforts on that [14, 15, 16]. However, these trials are generally domain-specific and do not incorporate enough discussions in the active sampling strategies. Besides, these trials are always confronted with a "cold start" problem [14], that is, active learning needs an initial labeled set to get started, and the size of the initial set often exceeds the budget in the semi-supervised context, (150-200 instances to get started), otherwise it would not perform better than random sampling. However, this “cold start” problem can be potentially tackled by incorporating the Unsupervised Selective Labeling method suggested by [17] to select the right training set at the start so that the subsequent active learning steps can be most effective.

In this study, we have developed a method that combines semi-supervised learning for histopathological image classification with active learning techniques. Our work highlights the importance of active sampling strategies even under the settings of semi-supervised learning. Furthermore, the framework we have devised not only reduces labeling expenses but also enhances model accuracy compared to the fully supervised method, representing a significant improvement over traditional methods, which often entail a trade-off between these two factors. The advancement of our algorithm offers substantial advantages for life scientists seeking to train machine learning models for classification tasks, such as cancer diagnosis, particularly in scenarios where labeled data is limited or expensive. This strategy is especially beneficial in regions where there is a shortage of expert diagnostic experience, such as that of pathologists.

# Experimental Design: Methods

## Dataset

In this project, we used a collection of histological images of human colorectal cancer. The dataset includes 5,000 images belonging to eight balanced classes of tissues (Figure 1). Each image has the same size of 150 × 150 pixels with 3 RGB colour channels. The eight tissue classes present in our dataset are tumour epithelium, simple stroma (homogeneous composition, includes tumour stroma, extra-tumoural stroma and smooth muscle), complex stroma (containing single tumour cells and/or few immune cells), immune cells (including immune-cell conglomerates and sub-mucosal lymphoid follicles), debris (including necrosis, haemorrhage and mucus), normal mucosal glands, adipose tissue, background (no tissue). These classes have been labelled as *Tumour, Stroma, Complex, Lympho, Debris, Mucosa, Adipose and Empty* respectively.

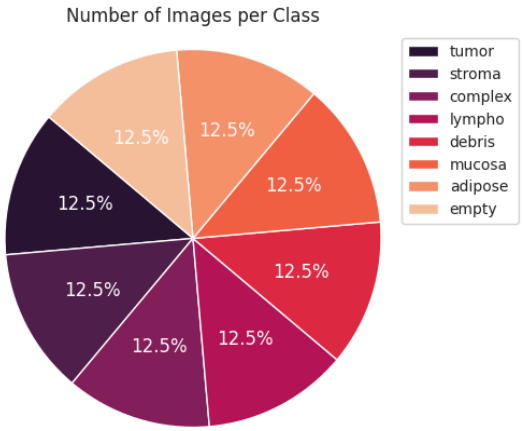


**Figure 1:** Distribution of classes in the dataset

## Stratified splitting

A stratified approach ensured that both subsets contained representative samples from all classes present in the original dataset. Specifically, the data was randomly shuffled for each class, and a portion

corresponding to the specified test and train size was selected. This process guarantees that the distribution of classes is maintained in both the training and testing subsets.

Accordingly, we divided the dataset into train and test subsets using an 80/20 split. In most of our experiments, we just used stratified sampling as a proxy for the unsupervised initial data selection method.

## Uncertainty sampling - MC dropout

We implemented uncertainty sampling to select the most uncertain instances in the unlabelled set. Uncertainty was quantified using Monte Carlo (MC) dropout in neural networks. The MC dropout or the Monte Carlo dropout is a way to learn the predictive distribution in a neural network. It randomly switches off neurons in a neural network, which regularizes the network. Each dropout corresponds to a different sample from the approximate parametric posterior distribution. Each dropout configuration yields a different output, and multiple forward passes with different dropout configurations yield the predictive distribution. Each dropout configuration was chosen by sampling from a Bernoulli distribution.

Hence, for each sample in the unlabelled set, we used the softmax function to obtain class probabilities based on the model predictions. The class probabilities are represented as a 1x*numClasses* vector where each value represents the probability of that sample belonging to that particular class. We obtained *m* probabilities for *m* different dropout configurations of the base-learner. We then computed the mean/variance of these probabilities across each class to obtain a single probability vector for each sample. It is important to note that if using variance, we must normalize the probabilities in the final vector and ensure they sum to 1. This is implicitly accounted for when using the mean. Finally, we used Shannon's entropy with log base 2 to obtain the uncertainty for that sample. This process is repeated for each sample in the unlabelled set. Once the uncertainty is quantified, we pick a batch of *N* samples to add to our training set before the model is retrained on the new training set.

**Algorithm 1** MC Dropout for Uncertainty Estimation

**Input**: Unlabeled data loader $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$, number of forward passes $F$, model with dropout $m$, device for computation
**Initialize**: Uncertainty scores list $\mathcal{S} = []$
**for** each batch $U_b$ from $\mathcal{U}$ **do**
    Initialize tensor for predictions: $P_b = []$
    **for** $f = 1$ **to** $F$ **do**
        Perform forward pass with dropout: $o_f = m(U_b)$
        Apply softmax to get probabilities: $p_f = \text{softmax}(o_f)$
        $P_b$.append($p_f$)
    **end for**
    Calculate mean probabilities: $\bar{p} = \text{mean}(P_b)$
    Compute entropy for the batch: $H_b = -\sum \bar{p} \log(\bar{p})$
    $\mathcal{S}$.append($H_b$)
**end for**
**Select instances**: Indices = argsort($\mathcal{S}$)[: top $K$]
**Return**: Indices of instances with highest uncertainty

**Figure 2:** Pseudo code to implement MC dropout for Uncertainty Sampling

## Image Augmentation

We adopted both weak and strong data augmentation approach for the analysis to add variability of the images. The weak augmentation applied random horizontal and vertical flips independently with a 50% probability, introducing variations in the orientation of the images. We used strong augmentation to identify images that exhibited substantial loss following augmentation. The model predicted the labels for the strongly-augmented images, resulting high loss in such cases suggests that the model struggles to accurately predict these image types. Consequently, these challenging images are prime candidates for selection and labeling by the oracle. We used the same strong augmentation technique introduced in the FixMatch paper [8] with RandAugmentMC library. In particular, RandAugmentMC performs multiple (n = 2) random augmentations chosen from a predefined set of augmentation operations in fixmatch_augment_pool [supplementary Table 1]. Each augmentation operation is randomly selected from the pool, and its magnitude (v) is randomly chosen between 1 and a specified maximum value (m = 10).

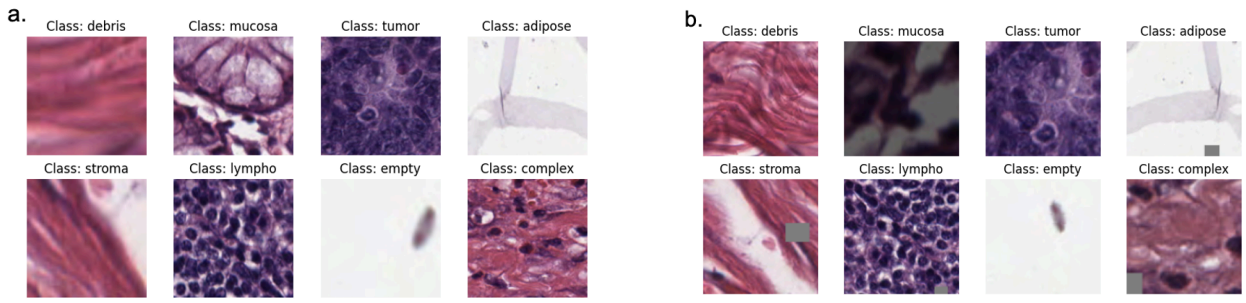


**Figure 3: (a)** Displays images after weak augmentation,

**(b)** Displays images after strong augmentation.

# Modeling Strategy

## Base Learner

We used ResNet-18 (Figure 3) pre-trained model as the base learner. It includes a dropout layer with a dropout probability of 0.25, followed by a linear layer with the number of input features from the original ResNet-18's fully connected layer and set 8 (number of classes) as the output size.

For active learning in supervised case, a dropout layer with probability = 0.25 is inserted before the fully connected layer.

Training hyperparameters: learning rate = 1e-2, batch size = 40, optimizer = Adam, weight decay = 1e-4, number of epochs = 100.

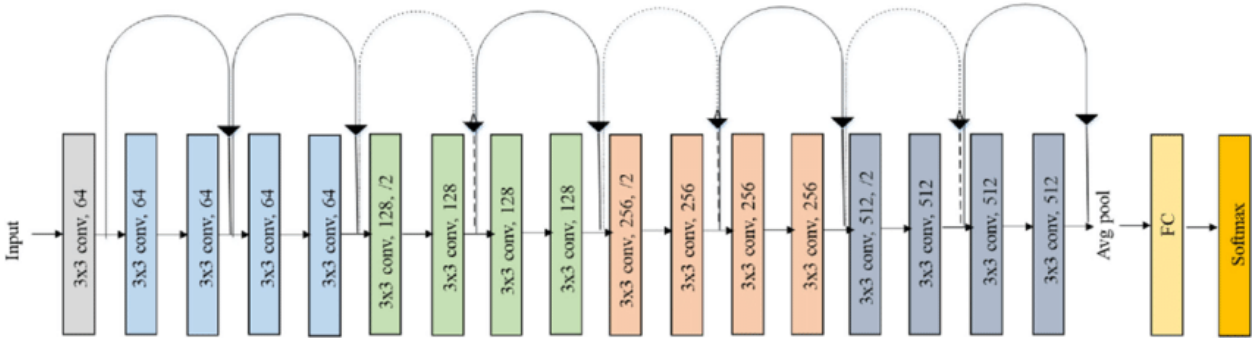

**Figure 4:** The ResNet18 Architecture [5]

## Supervised Learning

We used supervised learning as a baseline for comparison against all methods implemented in this study. To perform supervised learning, we split the data into a training and testing set using two sampling techniques - random sampling and stratified sampling. Post-split, the base learner is trained using the entirety of the training data; once trained, we tested the model on the test set to obtain classification accuracy. We performed three experiments using the same workflow but for varying training set sizes. Firstly, we used 80% of the training data to train the model. Secondly, we used training set sizes ranging from 1% to 50 % of the training data to train the model. All sets were obtained via random sampling in this experiment. Thirdly, we utilized training sets obtained via stratified sampling, and the sizes of training sets varied from 1% to 50% of the training data.

## Supervised Learning with Active Learning

We implemented active learning in a supervised learning setting by choosing the most uncertain set of instances to add to the labelled set until we reach a pre-defined training set size. We trained the base learner initially using 10% of the training data size. This set was chosen via stratified sampling. We then we iteratively selectd a batch of $N$ most uncertain samples using MC-dropout and add them to the labelled set. At every step, we train the model using the new labelled set and then record the test accuracy. We continue this process until we have observed 50% of the total training set.

## Semi-supervised Learning

Here we adopted a semi-supervised learning technique called FixMatch [5], which leverages both labeled and unlabeled data to train a model. It contains two key concepts: consistency regularization and pseudo-labeling.

Consistency regularization involves enforcing the model to produce consistent predictions for different augmentations or perturbations of the same input data. The key idea behind consistency regularization is that a model should output the same prediction for an input, no matter how the input is perturbed or augmented, as long as the semantic content of the input remains the same. This forces the model to focus on the underlying patterns that are invariant to such changes, thus helping it to generalize better from limited labeled data to unseen data.

Pseudo-labeling is a strategy in semi-supervised learning where the model uses its own predictions to generate labels for unlabeled data, thereby extending its training dataset. In the FixMatch algorithm, pseudo-labeling is employed alongside consistency regularization to make effective use of both labeled and unlabeled data.

The process of pseudo-labeling in FixMatch operates as follows: The model predicts labels for unlabeled data using weak augmentation; if the confidence is high enough, these labels are treated as 'pseudo-labels.' The model is then trained to predict these pseudo-labels on the same data but using strong augmentation. This approach encourages the model to make consistent predictions across different views of the same data, enhancing learning efficiency and model robustness in scenarios with limited labeled data

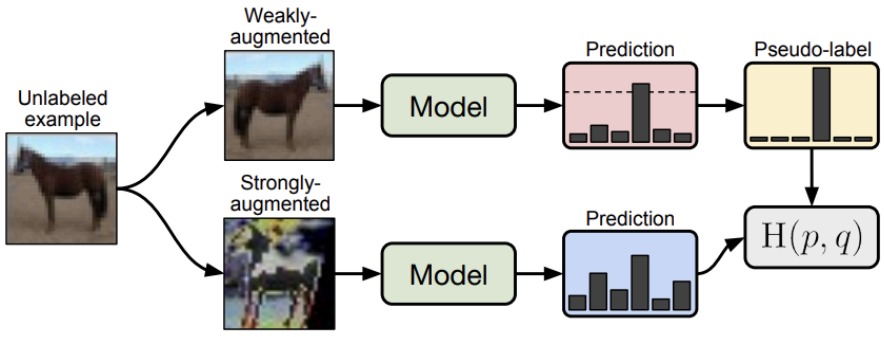


**Figure 5:** Overview of the FixMatch Algorithm [8]

## Uncertainty Sampling in Semi-supervised Learning

In our pursuit of enhancing the semi-supervised learning framework, we are exploring the integration of active learning techniques to make the training process more data-efficient and effective. Active learning is particularly beneficial in scenarios where labeled data is scarce or expensive to obtain, as it allows the model to actively query the most informative samples from an unlabeled dataset for labeling. To implement this, we focus on two distinct sampling strategies based on the metrics of uncertainty and consistency. First, we use uncertainty-based sampling to identify data points where the model exhibits the highest uncertainty in its predictions. This involves applying dropout during inference, which allows us to simulate different neural network paths and generate a distribution over the outputs. By averaging these outputs, we estimate the mean class probabilities and subsequently compute the entropy to gauge uncertainty. Samples with the highest entropy are deemed the most uncertain and are thus prioritized for labeling, as they are likely to contribute most significantly to model learning.

## Consistency Sampling in Semi-supervised Learning

We employed consistency-based sampling to ensure that our model remains stable across different representations of the same data. Here, we manipulate data through weak and strong augmentations and demand high consistency in the predictions of the augmented pairs. We assess this by computing the negative cosine similarity between the prediction vectors of weakly and strongly augmented versions of the same input. Instances exhibiting the highest negative cosine similarity—indicating the greatest inconsistency—are selected for further querying. This approach not only aids in refining the model's robustness but also in verifying that the model can generalize well across varied representations of similar data. By integrating these active learning methods into our semi-supervised learning framework, we aim to reduce the dependency on extensive labeled datasets and enhance the model's performance using strategically selected informative samples. This integration represents a critical step towards developing more efficient and robust machine learning models.

**Algorithm 2** Augmentation Consistency Estimation

**Input**: Unlabeled data loader $\mathcal{U} = \{(inputs_w, inputs_s) : b \in (1, \ldots, \mu B)\}$, number of forward passes $F$, model $m$, device for computation
**Initialize**: Lists for predictions $\mathcal{S} = []$ for strong augmentations, $\mathcal{W} = []$ for weak augmentations
**for** each $(inputs_w, inputs_s)$ batch from $\mathcal{U}$ **do**
    Initialize batch predictions: $\mathcal{P}_s = []$, $\mathcal{P}_w = []$
    **for** $f = 1$ to $F$ **do**
        Compute predictions $o_s = m(inputs_s)$, $o_w = m(inputs_w)$
        Apply softmax: $p_s = \text{softmax}(o_s)$, $p_w = \text{softmax}(o_w)$
        Collect predictions: $\mathcal{P}_s$.append$(p_s)$, $\mathcal{P}_w$.append$(p_w)$
    **end for**
    Calculate mean probabilities: $\bar{p}_s = \text{mean}(\mathcal{P}_s)$, $\bar{p}_w = \text{mean}(\mathcal{P}_w)$
    Add to predictions list: $\mathcal{S}$.append$(\bar{p}_s)$, $\mathcal{W}$.append$(\bar{p}_w)$
**end for**
Compute negative cosine similarity scores $\mathcal{C} = -\frac{S \cdot W}{\|S\| \|W\|}$ for $\mathcal{S}$, $\mathcal{W}$
**Select instances**: Indices = argsort$(\mathcal{C})$[top $K$]
**Return**: Indices of instances with highest inconsistency for labeling

**Figure 6:** Pseudocode of Consistency-based Sampling

# A Combined Way of Active Learning:

In our continued efforts to refine our semi-supervised learning framework, we propose a novel active learning strategy that synergistically combines uncertainty-based and consistency-based sampling. This approach is designed to harness the complementary strengths of each method, optimizing our model's training process by selecting the most informative samples from an unlabeled dataset. The core idea of this strategy is to compute both an uncertainty score and a consistency score for each sample, and then combine these scores by multiplication. This combined score aims to select samples that are not only uncertain but also consistently represented across different data augmentations. The rationale behind this approach is enhanced reliability of uncertainty measures: By integrating consistency measurements, we can ensure that the high uncertainty scores are not merely the result of noise or anomalies in data representation. Consistency acts as a regulatory mechanism, confirming that the uncertainty observed is intrinsic to the model's current state of knowledge and not due to external variability in data presentation. By multiplying the uncertainty and consistency scores, we prioritize samples that the model consistently finds uncertain, ensuring that every queried sample genuinely contributes to significant model

improvements. This approach not only refines the model's understanding and robustness but also efficiently utilizes the labeling resources by focusing on samples that are genuinely indicative of the model's limitations.

**Algorithm 3** Combined Uncertainty and Consistency Sampling

**Input:** Unlabeled data loader $U = \{u_b : b \in (1, \ldots, B)\}$, number of forward passes $F$, model with dropout $m$, device for computation
**Initialize:** Uncertainty scores list $S = []$, Consistency scores list $W = []$, Combined scores list $C = []$
**for** each batch $u_b$ from $U$ **do**
    Initialize tensor for predictions: $P_b = []$, $P_s = []$, $P_w = []$
    **for** $f = 1$ to $F$ **do**
        Perform forward pass with dropout: $o_f = m(u_b)$
        Apply softmax to get probabilities: $p_f = \text{softmax}(o_f)$
        $P_b.\text{append}(p_f)$
        Compute predictions for weak and strong augmentations: $o_s = m(\text{inputs}_s^w), o_w = m(\text{inputs}_s)$
        Apply softmax: $p_s = \text{softmax}(o_s), p_w = \text{softmax}(o_w)$
        $P_s.\text{append}(p_s), P_w.\text{append}(p_w)$
    **end for**
    Calculate mean probabilities: $\bar{p} = \text{mean}(P_b), \bar{p}_s = \text{mean}(P_s), \bar{p}_w = \text{mean}(P_w)$
    Compute entropy for the batch: $H_b = -\sum \bar{p} \log(\bar{p})$
    Compute negative cosine similarity for the batch: $C_s = -(\bar{p}_s \cdot \bar{p}_w)/(\|\bar{p}_s\| \cdot \|\bar{p}_w\|)$
    $S.\text{append}(H_b), W.\text{append}(C_s)$
**end for**
Compute combined scores: $C = [s \cdot w \text{ for } s, w \text{ in } \text{zip}(S, W)]$
Select instances: Indices $= \text{argsort}(C)[: \text{top } K]$
**Return:** Indices of instances with highest combined score (uncertainty and inconsistency)

**Figure 7:** Pseudocode for Combining Uncertainty and Consistency Sampling

# Results

To compare the accuracies of different models, we plotted the test accuracies of each model as a function of the percentage of training data used to train the model.

## Supervised Learning (baseline)

We compared the performance of supervised learning for varying sampling strategies. The comparison of performance of these methods can be seen in Figure 8. For the initial set of experiments (1% - 10%) of the

training data, we observed that stratified sampling (mean accuracy = 54.8%) outperformed random sampling (mean accuracy = 50.6%). This indicated that the way we choose/create our training set plays an important role in the overall performance of the model. We also implemented uncertainty sampling using MC dropout and observed that this technique significantly outperformed random and stratified sampling. Additionally, this method achieved a higher accuracy than both aforementioned methods while utilizing a smaller percentage of the training data.

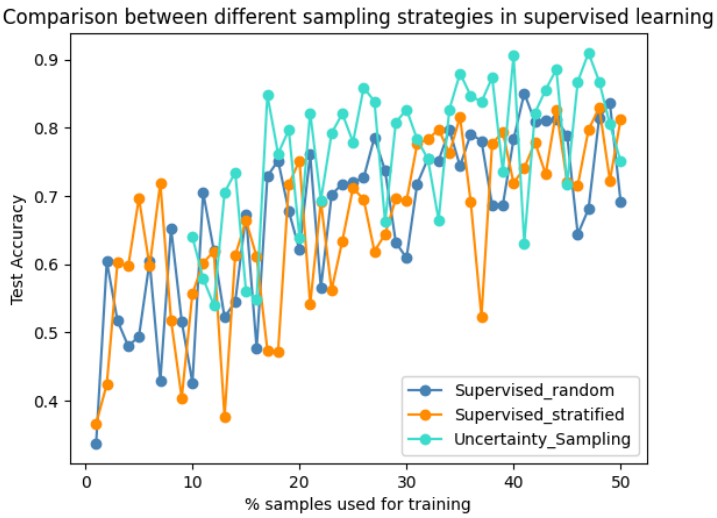


**Figure 8**: Comparison of different sampling strategies in supervised learning

## Semi-Supervised Learning

We compared the performance of semi-supervised learning, where the training set was chosen via stratified sampling against supervised learning implemented using random and stratified sampling (Figure 9). We implemented semi-supervised learning for 30 iterations, and each iteration represented a 1% increase in the size of the training data used to train the model. Based on the plot in Figure 9, we can clearly see that semi-supervised learning significantly outperforms both supervised learning frameworks.

This was expected since semi-supervised learning utilized the unlabelled data's intrinsic properties via a sophisticated framework to learn the labels of the dataset.

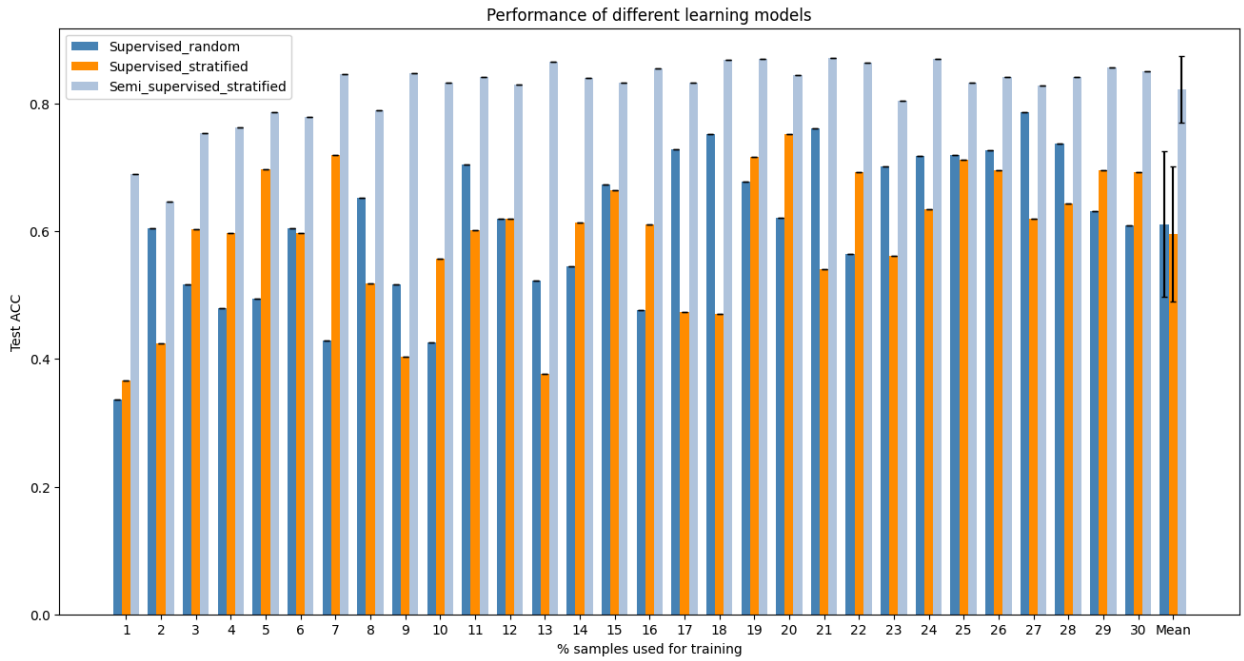


**Figure 9:** Comparison of performance of Semi-Supervised Learning (Stratified Sampling) against different Supervised Learning frameworks (Random and Stratified Sampling)

We also compared the performance of semi-supervised learning against uncertainty sampling in a supervised setting and observed that uncertainty sampling performed better than semi-supervised learning (Figure 10). This may have been because uncertainty sampling was trained for a larger training set size (more iterations) compared to semi-supervised learning.

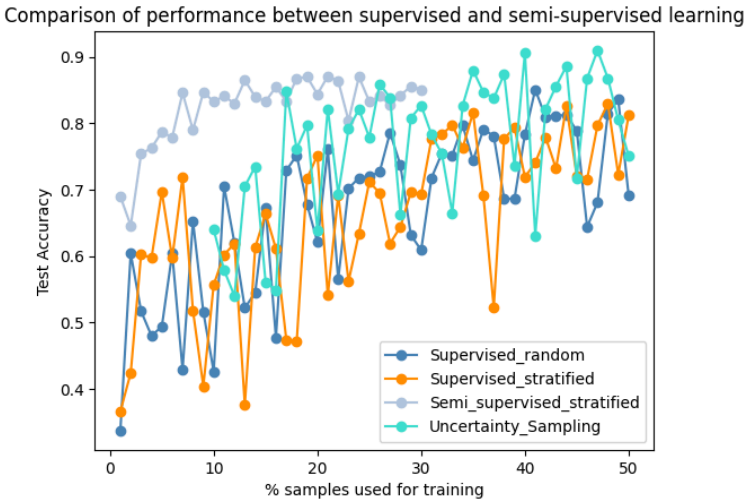


**Figure 10:** Comparison of performance of semi-supervised learning against supervised learning and active supervised learning

## Semi-Supervised Learning with Active Learning

Next, we compared the performances of different sampling strategies within a semi-supervised learning framework, including: Consistency Sampling, Uncertainty Sampling, Stratified Sampling, and a Combined Strategy. We presented the test accuracies achieved by each method relative to the percentage of training data utilized. The graph in Figure 11 delineates the progression of model accuracy as the percentage of training data increases from 5% to 9%, with the initial training set generated from stratified sampling. Since all the experiments are only run once with possibly different initial starts, the curves may not reflect the true trends. However, we did observe that all the three active learning strategies achieve comparable performances with stratified sampling, which is satisfying given that stratified sampling performs generally better than random sampling and is infeasible in reality since we are not aware of the true label distributions ahead of time.

We also compared the average accuracies of the 5 rounds across the four strategies, as shown in Table 1, which also indicates the effectiveness of the active learning strategies.

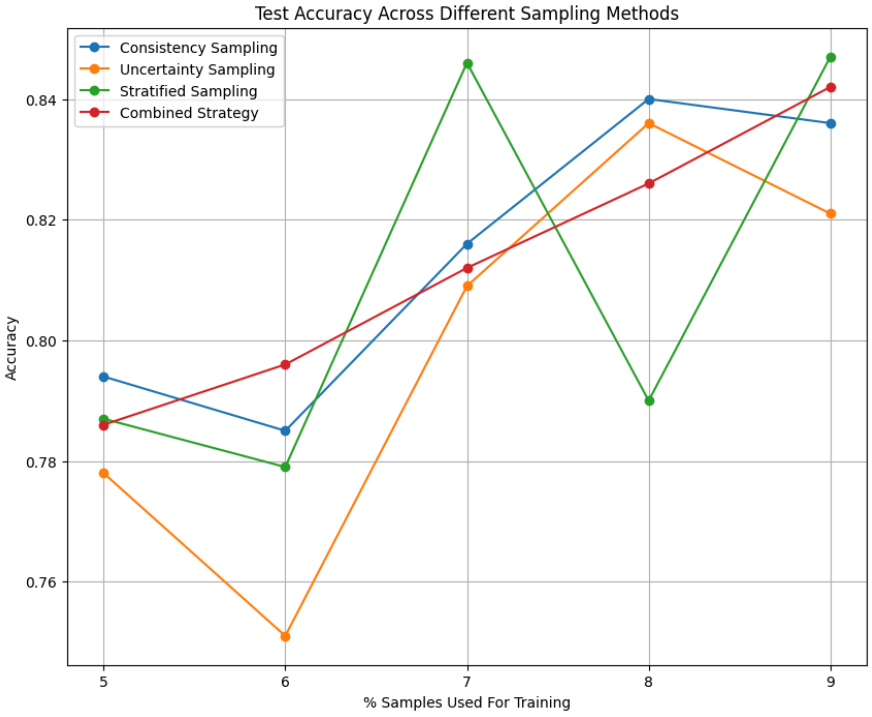


**Figure 11:** Comparison of different query selection strategies with semi-supervised learning

| Methods | Stratefied | Uncertainty | Consistency | Combined |
|---|---|---|---|---|
| **Accuracies %** | 81.0 | 79.9 | 81.4 | 81.2 |

# Comparison of all techniques

In this section, we employed an empirical strategy to demonstrate the effectiveness of semi-supervised learning frameworks in medical image classification. We compared the amount of training data required by every algorithm utilized in this study to achieve an accuracy above a predefined threshold. We set this threshold to be 0.81, as we considered this to be the minimum accuracy required for a model to be considered “acceptable.” The results of this approach have been summarized in Table 2.

| Algorithm | First Occurrence of Accuracy % Above Threshold (81) | % Training Data Used To Obtain Accuracy |
|---|---|---|
| **Supervised Learning (Random Sampling)** | 85.1 | 41 |
| **Supervised Learning (Stratified Sampling)** | 81.6 | 35 |
| **Supervised Learning (Uncertainty Sampling)** | 84.8 | 17 |
| **Semi-Supervised Learning (Stratified Sampling)** | 84.6 | 7 |
| **Semi-supervised learning (Uncertainty Sampling)** | 83.6 | 8 |
| **Semi-supervised learning (Consistency Sampling)** | 81.6 | 7 |
| **Semi-supervised learning (Combination of Consistency and Uncertainty Sampling)** | 81.2 | 7 |

**Table 2:** Comparison of amount of training data used by different frameworks to obtain a test accuracy above a pre-defined threshold.

Given the results in Table 2, we can clearly see that all semi-supervised learning frameworks achieve an accuracy that meets the threshold using significantly less training data than all supervised learning strategies. While the accuracies of supervised learning recorded in Table 2 are higher than those observed in a semi-supervised setting, they do so by using 2 to 5 times more training data.

# Discussion

Our study highlights the efficacy of semi-supervised learning (SSL) frameworks, particularly when initialized with stratified datasets, which significantly outperform traditional supervised learning approaches. Our semi-supervised active learning (SSAL) framework yields results comparable to SSL with stratified sampling but is more practical, avoiding the complexities and high costs associated with

obtaining perfectly stratified samples. Moreover, by synergizing SSL and active learning, our approach substantially reduces labeling costs, making it highly suitable for real-world applications where large labeled datasets are costly or difficult to procure.

Our model's success can be attributed to several key factors: 1) Effective use of FixMatch: The even distribution of classes in our dataset ensures that FixMatch works optimally, leveraging its pseudo-labeling effectively without bias towards any particular class; 2) Enhanced uncertainty estimation with dropout: Dropout not only helps in preventing overfitting but also enhances the reliability of uncertainty estimations. This is critical for active learning where the model's uncertainty guides the selection of samples for labeling. 3) Mitigation of distribution shifts: Our SSL approach counters potential biases and distribution shifts that often arise from aggressive querying in active learning. By utilizing a wide range of unlabeled data, the model gains exposure to diverse data characteristics, which helps in maintaining a generalizable performance across different data distributions.

To enhance the effectiveness of our semi-supervised active learning (SSAL) framework, we are considering several avenues for future research: 1) Beyond uncertainty and consistency, we plan to explore other criteria that could further refine sample selection, such as novelty or representativeness, to ensure a more comprehensive learning from diverse data facets. 2) Investigating methods for intelligently sampling the initial set of data in an unsupervised manner could provide a more robust starting point for the model, potentially improving learning outcomes from the outset. 3) Introducing Monte Carlo (MC) dropout during the pseudo-labeling phase could enhance the probability calibration of our model, providing more reliable confidence estimates for the pseudo-labels used during training. 4) Extending our methods to more complex, long-tail datasets where class distributions are highly imbalanced would test the adaptability and robustness of our framework [18], providing insights into its scalability and effectiveness in more challenging scenarios.

Lastly, we implemented the unsupervised learning using the SimCLR algorithm described in [19]. However, due to limited time and computational resources, we were unable to obtain results for this particular framework when used as a precursor for semi-supervised learning. In the future, we would like to run our code for a sufficient number of epochs and maybe over multiple simulations to obtain comprehensive results.

# Reference


[1] Kather, J. N., Weis, C. A., Bianconi, F., Melchers, S. M., Schad, L. R., Gaiser, T., ... & Zöllner, F. G. (2016). Multi-class texture analysis in colorectal cancer histology. Scientific reports, 6(1), 1-11.

[2] Komura, D., & Ishikawa, S. (2018). Machine learning methods for histopathological image analysis. Computational and structural biotechnology journal, 16, 34-42.

[3] Colorectal cancer statistics | How common is colorectal cancer? (n.d.). American Cancer Society. https://www.cancer.org/cancer/types/colon-rectal-cancer/about/key-statistics.html#:~:text=Overall%2C%20the%20lifetime%20risk%20of,risk%20factors%20for%20colorectal%20cancer.

[4] Egeblad, M., Nakasone, E. S., & Werb, Z. (2010). Tumors as Organs: Complex Tissues that Interface with the Entire Organism. Developmental Cell, 18(6), 884–901. https://doi.org/10.1016/j.devcel.2010.05.012

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[6] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[8] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., ... & Li, C. L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems, 33, 596-608.

[9] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

[10] Zheng, M., You, S., Huang, L., Wang, F., Qian, C., & Xu, C. (2022). Simmatch: Semi-supervised learning with similarity matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14471-14481).

[11] Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. B., ... & Wang, X. (2021). A survey of deep active learning. ACM computing surveys (CSUR), 54(9), 1-40.

[12] Settles, B. (2009). Active learning literature survey.

[13] Budd, S., Robinson, E. C., & Kainz, B. (2021). A survey on active learning and human-in-the-loop deep learning for medical image analysis. Medical image analysis, 71, 102062..

[14] Gao, M., Zhang, Z., Yu, G., Arık, S. Ö., Davis, L. S., & Pfister, T. (2020). Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16 (pp. 510-526). Springer International Publishing.

[15] Yu, W., Zhu, S., Yang, T., & Chen, C. (2022). Consistency-based active learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3951-3960).

[16] Guo, J., Shi, H., Kang, Y., Kuang, K., Tang, S., Jiang, Z., ... & Zhuang, Y. (2021). Semi-supervised active learning for semi-supervised models: Exploit adversarial examples with graph-based virtual labels. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2896-2905).

[17] Wang, X., Lian, L., & Yu, S. X. (2022, October). Unsupervised selective labeling for more effective semi-supervised learning. In European Conference on Computer Vision (pp. 427-445). Cham: Springer Nature Switzerland.

[18] Geifman, Y., & El-Yaniv, R. (2017). Deep active learning over the long tail. arXiv preprint arXiv:1711.00941.

[19] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.

# Supplementary Information

**Supplementary Table 1.**
The augmentation operations available in the fixmatch_augment_pool include:

| | |
|---|---|
| **AutoContrast** | Automatically adjusts image contrast to enhance details |
| **Brightness** | Adjusts the brightness of the image. |
| **Color** | Adjusts the color balance of the image |
| **Contrast** | Adjusts the contrast of the image |
| **Equalize** | Equalizes the histogram of the image to improve contrast |
| **Identity** | No transformation (identity operation) |
| **Posterize** | Reduces the number of bits for each color channel |
| **Rotate** | Rotates the image by a random angle |
| **Sharpness** | Adjusts the sharpness of the image |
| **ShearX** | Shears the image along the x-axis |
| **ShearY** | Shears the image along the y-axis |
| **Solarize** | Inverts all pixel values above a threshold value |
| **TranslateX** | Translates the image along the x-axis |
| **TranslateY** | Translates the image along the y-axis |

**Supplementary Table 2.**
The hyperparameters used for model training:

| **Learning rate** | **Optimizer** | **Weight decay** | **Number of epochs in SL** | **Number of epochs in SSL** |
|---|---|---|---|---|
| 1e-2 | Adam | 1e-4 | 100 | 30 |
| **Labeled Batch Size** | **Unlabeled Batch Size** | **Eval Steps** | **Unlabeled balance λ** | **Forward passes** |
| 40 | 200 | 100 | 1 | 20 |