BMC Medical Imaging

RESEARCH

Open Access

A predicted-loss based active learning approach for robust cancer pathology image analysis in the workplace

Mujin Kim[1], Willmer Rafell Quiñones Robles[1], Young Sin Ko[2], Bryan Wong[1], Sol Lee[1] and Mun Yong Yi[1*]

Abstract

Background Convolutional neural network-based image processing research is actively being conducted for pathology image analysis. As a convolutional neural network model requires a large amount of image data for training, active learning (AL) has been developed to produce efficient learning with a small amount of training data. However, existing studies have not specifically considered the characteristics of pathological data collected from the workplace. For various reasons, noisy patches can be selected instead of clean patches during AL, thereby reducing its efficiency. This study proposes an effective AL method for cancer pathology that works robustly on noisy datasets.

Methods Our proposed method to develop a robust AL approach for noisy histopathology datasets consists of the following three steps: 1) training a loss prediction module, 2) collecting predicted loss values, and 3) sampling data for labeling. This proposed method calculates the amount of information in unlabeled data as predicted loss values and removes noisy data based on predicted loss values to reduce the rate at which noisy data are selected from the unlabeled dataset. We identified a suitable threshold for optimizing the efficiency of AL through sensitivity analysis.

Results We compared the results obtained with the identified threshold with those of existing representative AL methods. In the final iteration, the proposed method achieved a performance of 91.7% on the noisy dataset and 92.4% on the clean dataset, resulting in a performance reduction of less than 1%. Concomitantly, the noise selection ratio averaged only 2.93% on each iteration.

Conclusions The proposed AL method showed robust performance on datasets containing noisy data by avoiding data selection in predictive loss intervals where noisy data are likely to be distributed. The proposed method contributes to medical image analysis by screening data and producing a robust and effective classification model tailored for cancer pathology image processing in the workplace.

Keywords Active learning strategy, Noisy data, Cancer pathology images, Convolutional neural networks, Deep learning, Histopathology image analysis, Predicted loss

Background

Cancer is a major cause of death worldwide, characterized by high heterogeneity and significant barriers to extending human life expectancy [1, 2]. According to a World Health Organization survey, cancer is the leading or second leading cause of death [3]. The current standard for diagnosing cancer involves pathologists reviewing

*Correspondence:
Mun Yong Yi
munyi@kaist.ac.kr
[1] Graduate School of Data Science, Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea
[2] Pathology Center, Seegene Medical Foundation, Seoul, South Korea

glass slides with stained suspicious tissue under a high-power microscope [4]. However, with an annual increase in cancer cases and a relatively scarce number of pathologists, the workload of pathologists has increased, leading to approximately 3–9% of human errors in anatomical pathology [5, 6]. To reduce workload, there has been active research on deep learning (DL)-based models that analyze digitized whole-slide images (WSIs) [7–9].

To develop a DL-based model that analyzes WSIs, it is common to use "patch images," which are relatively small images generated from WSIs [10–12]. WSIs are large digital images of actual slides created by a scanner, typically consisting of many gigapixels; up to 50,000 × 50,000 pixels. If we apply convolutional neural networks (CNNs), which are a representative DL architecture for image processing, directly to WSI classification, there can be two significant drawbacks. First, down-sampling can result in the loss of detailed information, and second, CNNs can only learn some of the distinctive patterns that appear in multiple WSIs [13]. Therefore, it is advantageous to train a CNN with high-resolution patch images and predict a label for a WSI on the basis of patch-level information.

These high-performing CNN models require large amounts of labeled data [14]. The main challenge in building a high-quality dataset for CNN training is the labor-intensive and time-consuming process of labeling medical images by expensive medical experts [15]. Recently, various methods, such as semi-supervision, transfer learning, and multi-instance learning, have been studied to overcome this labeling issue in medical image analysis [16]. Active learning (AL) is one approach that focuses on acquiring labels for the most informative data in an efficient manner, exploring how to efficiently acquire "real" labels. Unlike other methods, AL allows the DL model to actively select and preferentially label the most informative data obtained from medical experts, to optimize the trade-off between labeling efforts and model performances [17]. The AL framework typically consists of a method for measuring the informativeness of each unlabeled data point, as shown in Fig. 1.

First, as shown in Fig. 1, the model selects informative data using a selection algorithm from an unlabeled dataset and requests an oracle (i.e., medical expert) to label the queried images. Second, these labeled data are used to train the DL model. By repeating this process, the size of the labeled dataset gradually increases. Through this framework, a high-performance DL model can be trained at a low labeling cost.

In the context of AL, a model is initially trained with a small amount of labeled data (typically an arbitrary number selected by a researcher such as 40 [18], or 500 [15]) and then repeatedly selects data to request labeling from an oracle using an acquisition function (query), often based on the model's uncertainty information [19]. Various AL strategies have been proposed for medical image analysis to reduce labeling costs. For instance, several studies have explored uncertainty in nuclear segmentation within histopathological images, with some focusing on utilizing the posterior probability of the output to compare model performance for breast and pancreatic cancer [20]. Another study introduced methods for measuring uncertainty using Bayesian CNNs specifically for skin cancer [19], while a more recent study proposed an uncertainty measurement approach that comprehensively considers both entropy and high confidence scores in the context of breast cancer [21].

Recently, in the field of digital pathology, there have been studies that combine methods for removing false-labeled patch images with uncertainty-based AL strategies [22], or that consider both uncertainty and representation in patch-based analysis [15]. However, the use of an AL strategy based on uncertainty can be challenging when dealing with noisy real-world industrial data, as most DL studies use clean or minimally noisy (dirty) publicly available datasets, potentially worsening performance when noisy samples are queried [23]. Noisy images can be generated in the workplace due to various issues, such as out-of-focus scanning, missing tissue, air bubbles, poor staining, poor sectioning, tissue artifacts, tissue folding, or poor dehydration [24, 25], leading to poor quality patch images. Therefore, there is a dire need to develop AL strategies that are suitable for noisy real-world industrial datasets.

In this study, we proposed a novel AL strategy for analyzing histopathological images that minimize the selection of noisy data when querying data from an unlabeled set. The purpose of this study was to construct an AL strategy that can continuously improve the performance of histopathological patch classification models in situations where noisy data are included in the dataset. The proposed AL strategy was based on learning loss (LL) [26] and used a modified version of LossDiff [27] in the sampling stage of AL.

Methods

The primary goal of this research was to develop a robust AL approach for noisy histopathology datasets. The method consisted of three steps: 1) training a loss prediction module (LPM), 2) collecting predicted loss values, and 3) sampling data for labeling (Fig. 2).

As shown in Fig. 2, in the training loss prediction module (LPM) step, the backbone deep learning model and the LPM are trained. In the predicted loss values collection step, the predicted loss value of the correctly predicted data is collected to calculate the appropriate threshold values. Next, in the sampling step, the model calculates the prediction loss to select informative data from D_U, which is filtered by the threshold value calculated in the training step, and then the high uncertainty data is sampled.

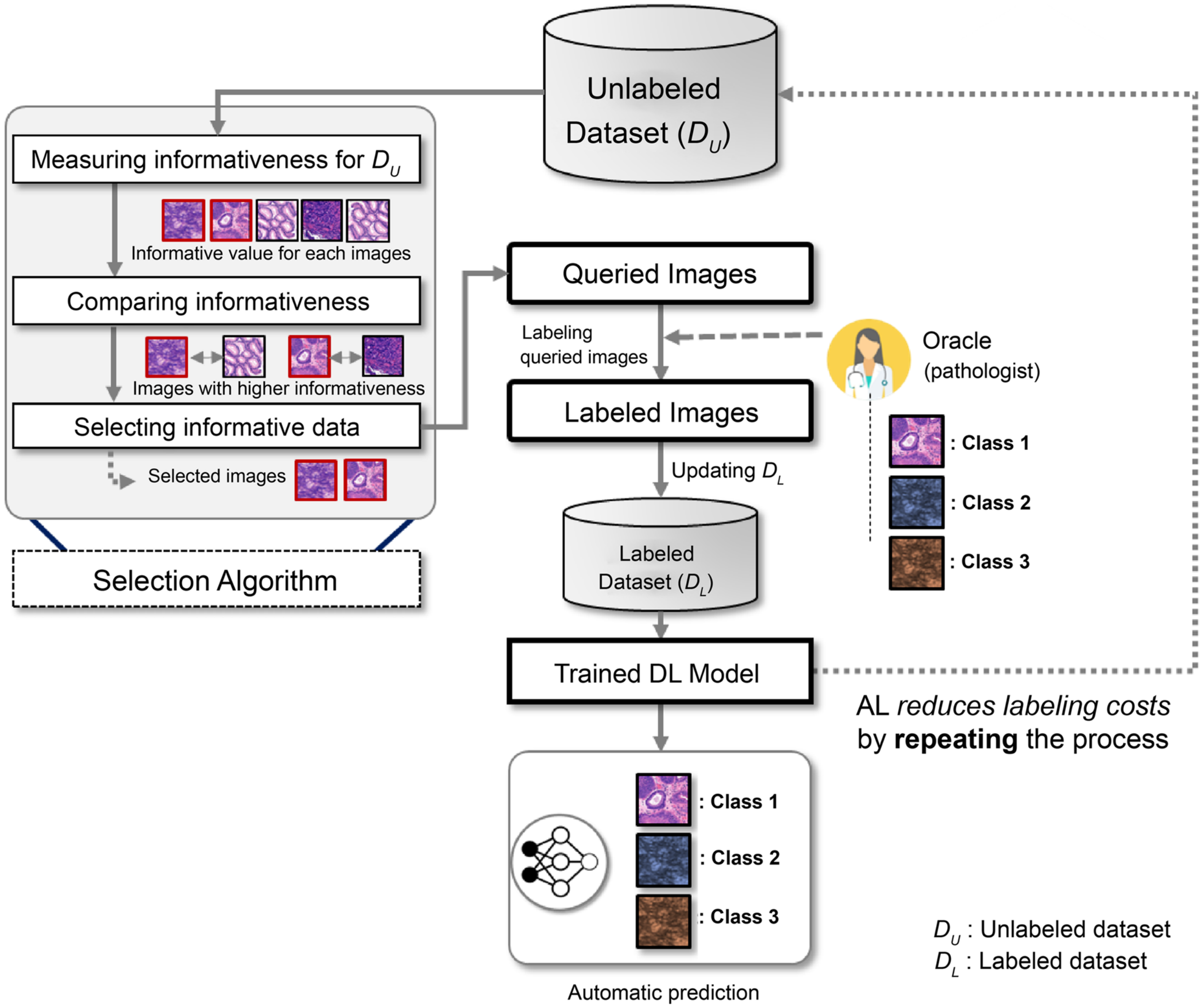

Unlabeled Dataset (D_U)
Measuring informativeness for D_U
Informative value for each images
Comparing informativeness
Images with higher informativeness
Selecting informative data
Selected images
Selection Algorithm
Queried Images
Labeling queried images
Oracle (pathologist)
Labeled Images
Updating D_L
: Class 1
: Class 2
: Class 3
Labeled Dataset (D_L)
Trained DL Model
AL *reduces labeling costs* by **repeating** the process
: Class 1
: Class 2
: Class 3
D_U : Unlabeled dataset
D_L : Labeled dataset
Automatic prediction

Fig. 1 Concept diagram of typical active learning (AL) framework. DL: deep learning

Training LPM step

The first step of the proposed AL method was handled by the LPM. The LPM is attached to a deep network and trained with the backbone to predict the loss of input data. Therefore, LL is expected to be widely useful, as LPMs can be attached to any kind of DL networks. In this study we used Visual Geometry Group-16 (VGG-16, VGG Group, Oxford, UK) [28] as the backbone architecture, without dropout. Figure 3 depicts a conceptual diagram of the LPM, in which the second diagram depicts an expanded view of the first one.

To train the LPM simultaneously with the backbone model, we calculated the total loss value by summing the target model loss originating from the backbone and the loss of the LPM. The overall loss function is represented by Eq. (1).

$$L_{target}\left(\widehat{y}, \mathrm{y}\right) + \lambda \cdot L_{loss}\left(\widehat{l}, l\right) \tag{1}$$

In addition, it is recommended to use the margin-ranking loss function to train the LPM for better performance [26]. The margin-ranking loss is a loss function that pairs samples within a batch to compare ranks. If the size of the mini-batch is B, we can create a $B/2$ data pair and train the LPM by considering the differences between the loss prediction pairs. Therefore, the loss function for the LPM was defined as follows:

$$\begin{aligned} &L_{loss}\left(\widehat{l^p}, l^p\right) = \max\left(0, -\mathbb{1}\left(l_i, l_j\right) \cdot \left(\widehat{l_i} - \widehat{l_j}\right) + \varepsilon\right) \\ &\text{s.t.}\mathbb{1}\left(l_i, l_j\right) = \begin{cases} +1, if\,\widehat{l_i} > \widehat{l_j} \\ -1, otherwise \end{cases} \end{aligned} \tag{2}$$

ε is the predefined positive margin and $\underset{\frown}{p}$ is the pair of i, j. For example, if l_i is larger than l_j and $\widehat{l_i}$ is greater than

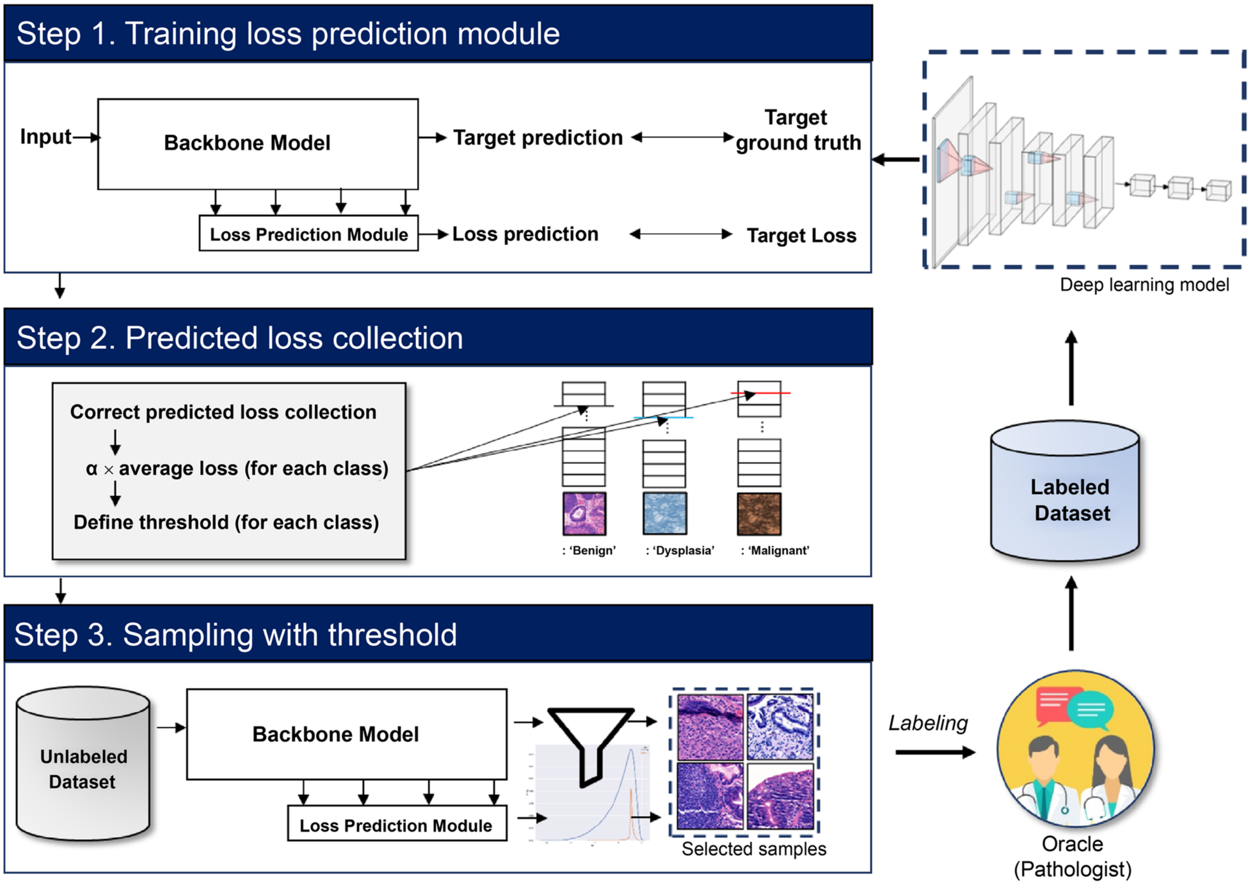


Fig. 2 The overall process of the proposed method

$\widehat{l_j}+\varepsilon$, the loss value is 0. By using this loss function, the LPM can learn to distinguish between informative and non-informative data and predict the loss value more accurately. The following Algorithm 1 represents the algorithm for training the LPM

```
Input: labeled dataset $\boldsymbol{D_L}$, unlabeled dataset $\boldsymbol{D_U}$, labeled patch $\mathbf{p}$, epoch $\mathbf{e}$, backbone
    network $\boldsymbol{F_b}$, loss prediction network $\boldsymbol{F_l}$
Output: $\boldsymbol{trained\ F_b, F_l, and\ \hat{l}}$, where
    $\boldsymbol{\hat{l}}$: predicted loss by the loss prediction module can be defined as = $\{\boldsymbol{\hat{l}_1},\ \boldsymbol{\hat{l}_2},\ \boldsymbol{\hat{l}_3}, \boldsymbol{\hat{l}_4}, \ldots \boldsymbol{\hat{l}_i}\}$
for epochs 1,2 …, $\boldsymbol{e}$ do
    1. compute training loss via Equation (1)
    2. update the network $\boldsymbol{F_b}$
    3. update the network $\boldsymbol{F_l}$,
end for
for 1, 2, … $\boldsymbol{p_i}$ do
    4. compute $\boldsymbol{\hat{l}_i}, \leftarrow \boldsymbol{F_l(p_i)}$: $\boldsymbol{p_i}$ from $\boldsymbol{D_L}$
end for
```

Algorithm 1. Loss prediction module

Predicted loss collection step

One of the essential components of the proposed AL method is the predicted loss collection module (PLCM). The PLCM generates an appropriate threshold for each class based on the predicted loss value. Specifically, the PLCM observes the predicted loss (b_l) based on the baseline labeled dataset (D_L). At this point, the loss results for instances that have been correctly classified as $y=\widehat{y}$ for each patch type (class) t, where y is the ground truth, $\widehat{y}$ is the model prediction, and $t \in \{D, M, N\}$, where D, M, N stands for a different disease class, respectively (the definitions of these disease classes are provided in *Dataset construction* Section), were recorded. For a batch b of m instances, the loss for correctly classified instances can be defined as $b_{cl}=\left\{\widehat{l}_{c1},\widehat{l}_{c2},\widehat{l}_{c3}\ldots\widehat{l}_{cn}\right\}$, where $\widehat{l}_{cn}$ denotes the predicted loss l of n correctly classified instances c. In addition, the loss for correctly classified instances and each patch type t within a batch were recorded and the average loss was obtained using the following Eq. (3):

$$b_{cl(avg)}=\left(\frac{\sum_{i=1}^{n}l_{ci}}{n}\right) \tag{3}$$

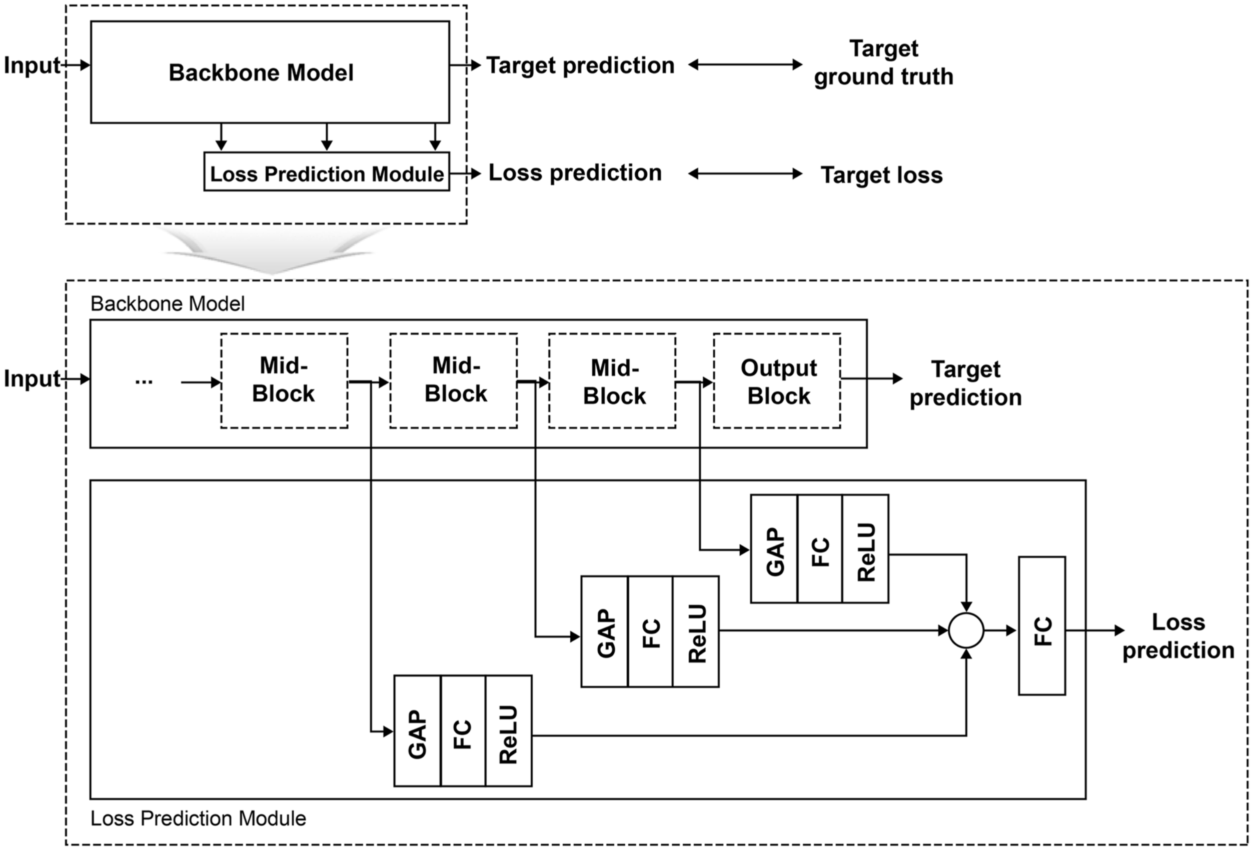


Fig. 3 A conceptual diagram of the Loss Prediction Module (LPM). GAP: global average pooling; FC: full connected layer; ReLU: Rectified Linear Unit

where n is the total number of correctly classified instance c. Furthermore, while $b_{cl(avg)}$ was obtained from each batch, the final average value of $b_{cl(avg)finalt}$ was obtained by collecting all $b_{cl(avg)}$ from the last z epochs, as shown in Eq. (4), where k is the number of batches in each epoch.

$$b_{cl(avg)final\,t} \leftarrow \left[\left(\frac{\sum_{j=1}^{k} b_{cl(avg)}j}{k} \right)_1 + \left(\frac{\sum_{j=1}^{k} b_{cl(avg)}j}{k} \right)_2 + \cdots + \left(\frac{\sum_{j=1}^{k} b_{cl(avg)}j}{k} \right)_z \right] \frac{1}{z} \tag{4}$$

In addition, it should be noted that $b_{cl(avg)finalt}$ has a different value depending on the patch type (i.e., D, M, N), thus if the number of the patch type is multiple t, $b_{cl(avg)final} = \{ b_{cl(avg)final1}, b_{cl(avg)final2}, \ldots, b_{cl(avg)finalt} \}$. Finally, to avoid filtering out difficult cases, α was used as a hyperparameter and was multiplied to $b_{cl(avg)finalt}$ for generating a threshold that could be separately set for each patch type t, as shown in Eq. (5).

$$threshold_t = a_t * b_{cl(avg)\mathrm{final}\,t} \tag{5}$$

The following Algorithm 2 represents the algorithm for training predicted loss collection.

```
Input: labeled dataset $D_L$, epoch $e$, number of collection epoch $z$, patch types $t$, predicted
    loss $\hat{l}$, final epoch $f$ and $\alpha$
Output: $threshold_1$, $threshold_2$, …, $threshold_t$
for epochs $e$-$z$, $e$-$z+1$, $e$-$z+2$ …, $e$ do
  for batches 1, 2, …, $k$ do
    for patch types 1, 2, …, $t$ do
      1. obtain $y = \hat{y}$ patches from $D_L$;
      2. obtain $b_{cl(avg)} \leftarrow \left(\frac{\sum_{i=1}^{n} l_{ci}}{n}\right)$;
    end for
  end for
end for
3. obtain $b_{cl(avg)final\ t}$ via Equation (4)
4. calculate $threshold_t = a_t * b_{cl(avg)\mathrm{final}\ t}$ when $e = f$;
```

Algorithm 2. Predicted loss collection

Sampling step

In the last step, the sampling module seeks to select informative samples, which is done by predicting the loss value in the LPM for all of the data contained in the D_U. For a clean dataset, data with the highest predicted loss can be assumed to be the most informative. However, for a noisy dataset, this assumption may not hold. Therefore, we sought to avoid selecting noisy data by removing those data with excessively high loss values from the selection candidates, based on the threshold calculated by the PLCM. Specifically, data with $\widehat{l} \geq threshold_t$ for the predicated loss, were excluded from the selection candidates [27]. In other words, our sampling module selected the top–k loss value data based on loss value among the data that satisfy "$\widehat{l} < threshold_t$," where k is the number of samples selected for a particular class. Because each threshold and data selection were conducted based on the predicted patch type by the backbone model, the threshold for the predicted class was used to exclude noisy data, and the same amount of data was sampled for each prediction class. Finally, the sampled data were labeled by an oracle and utilized for model training. After the sampling step, we defined D_{Ui+1} and D_{Li+1}, which were the datasets for the next iteration, as follows: D_{Ui+1}= D_{Ui} - p_l, D_{Li+1}= D_{Li}+ p_l, where p_l denotes the patches labeled by the oracle and can be defined as = $\{p_{l1}, p_{l2}, p_{l3}, \ldots p_{ln}\}$). A set of patches that maximize the sum of loss values when put into the loss prediction network can be defined as selected patches, following Equation (6), in which F_l denotes loss prediction network and p_s denotes selected (or sampled) patches. The following algorithm 3 represents the algorithm about the sampling for labeling.

$$\mathrm{argmax}_{S \subseteq D_U, |S| \leq k} \sum_{s} F_l(p_s) \tag{6}$$

```
Input: unlabeled dataset $D_U$, patch types $t$, predicted loss $\hat{l}$, patch data $p$, max number of
    selection $s$, labeled patches $p_l$, $k \leftarrow (s/t)$
Output: $p_l$, where
    $p_l$: labeled patches by oracle can be defined as = $\{p_{l1}, p_{l2}, p_{l3}, \ldots p_{ln}\}$
for patch types 1,2, ...$t$ do
    1. obtain '$\hat{y}$ = t', '$\hat{l} < threshold'_t$
        patches from $D_U$ as candidates ;
    2. candidates =+candidates;
end for
obtain $p_l$
    $p_l \leftarrow p_s$: $argmax_{S \subseteq candidates, |S| \leq k} \sum_{s} F_l(p_s)$;
label by oracle as $p_l$
update $D_U$ and $D_L$ in the next iteration
    $D_{U\ i+1} = D_{U\ i} - p_l$, $D_{L\ i+1} = D_{L\ i} + p_l$
```

Algorithm 3. Sampling for labeling

Experiment

Dataset construction

We constructed a large histopathology dataset extracted from stomach WSIs obtained from Seegene Medical Foundation, one of the largest diagnostic and pathology institutions in South Korea. These slides were stained with hematoxylin and eosin and scanned using a scanner (Pannoramic Flash250 III, 3DHISTECH, Budapest, Hungary) at 200× magnification. The data were collected by the medical foundation and their use for research was approved by the Institutional Review Board (SMF-IRB-2020-007) of Seegene Medical Foundation, as well as by the Institutional Review Board (KAIST-IRB-22-334, KH2020-116) of the Korea Advanced Institute of Science and Technology, the university that collaborated with the medical foundation. The medical foundation's designated collection centers provided informed consent to use their tissue samples for clinical purposes. All experiments were performed in accordance with relevant guidelines and regulations provided by the two review boards. All patient records were completely anonymized, and all images were stored and analyzed only on the organization's server.

To train the model, two types of datasets are needed: the unlabeled dataset (D_U) and the labeled dataset (D_L). The unlabeled dataset is a dataset containing all candidate patches that require labeling. Patches included in the unlabeled dataset cannot be used for training a model because they do not have labels (i.e., correct answers for classification). Therefore, in the process of AL, the model selects some patches with the highest amount of information from the unlabeled dataset and asks an oracle for a label. On the contrary, the labeled dataset is a dataset that includes labels assigned by an oracle. In order to build a labeled dataset, we randomly selected 1,000 patches and used them in the first iteration in all experiments. However, in subsequent iterations, the set of patches that made up the labeled dataset were changed as the amount of information of the patches changed while performing the AL experiment. Accordingly, the patches constituting the unlabeled dataset continuously changed.

To construct the unlabeled dataset, we collected 600 WSIs from different patients, each of which was then converted into a number of 256 × 256-pixel-sized patches, resulting in a total of 118,531 patches. The overall process is depicted in the left box of Fig. 4. However, as mentioned earlier, poor-quality images can be generated due to scanning or slide quality issues, resulting in the generation of noisy (unclean) samples of patch images, as shown in the right box of Fig. 4. Out of the total patches, 6,920 were classified as noisy. In sum, the unlabeled dataset consisted of 111,611 clean patches and 6,920 noisy patches.

For this study's experiment, patch images used for training were restricted to three classes: 1) benign, 2) dysplasia, and 3) malignant. "Benign" refers to a diagnosis of non-neoplastic benign gastric mucosal lesions, including gastritis and polyps. "Dysplasia" includes suspicious for (s/f) and suspicious of (s/o) tubular adenoma with dysplasia of any grade (s/f, s/o); while 'Malignant' covers malignant neoplasms, including adenocarcinoma, (s/f, s/o) adenocarcinoma, (s/f, s/o) high-grade lymphoma, and any other (s/f, s/o) carcinoma or malignant neoplasm. Thus, within the AL process, the model selected data from the D_U, which could be classified into three classes or noisy images by an oracle. In addition, the test dataset for evaluating the trained model's performance with D_L was constructed from images generated from 150 WSIs (50 per class), and a total of 30,523 patch images (11,753 Benign, 8,281 Dysplasia, and 10,489 Malignant) (Table 1).

Table 1 presents the dataset configuration, highlighting the D_U inclusion of 118,531 patch images sourced from 600 WSIs. Among them, certain images lacked sufficient meaningful information for classification into benign, dysplasia, or malignant categories, resulting in noisy patches labels. There was a total of 6,920 noisy patches, accounting for approximately 5.8% of all patches. As the experiment advanced, D_L systematically increased the patch count, reaching a maximum of 10,000 in the final iteration. Each iteration involved the selection of 1,000 samples. Additionally, test data, collected independently from D_U, were used to evaluate the model's performance.

Figure 5 shows examples of noisy and clean patches in D_U. Figure 5(a) shows patches with no or little tissue components. During the scanning process, typical examples include the presence of blood and stain dust. In some cases, foreign substances located outside of the tissue were inadvertently captured by the scanner and transformed into patches. Figure 5(b) specifically shows these patches, which presented difficulties in tissue capturing due to scanner errors or specific artifacts. Scanner focus-out is a frequently encountered error that can impede tissue capturing, and the presence of dust or air bubbles can also interfere with tissue capturing. In addition, there are

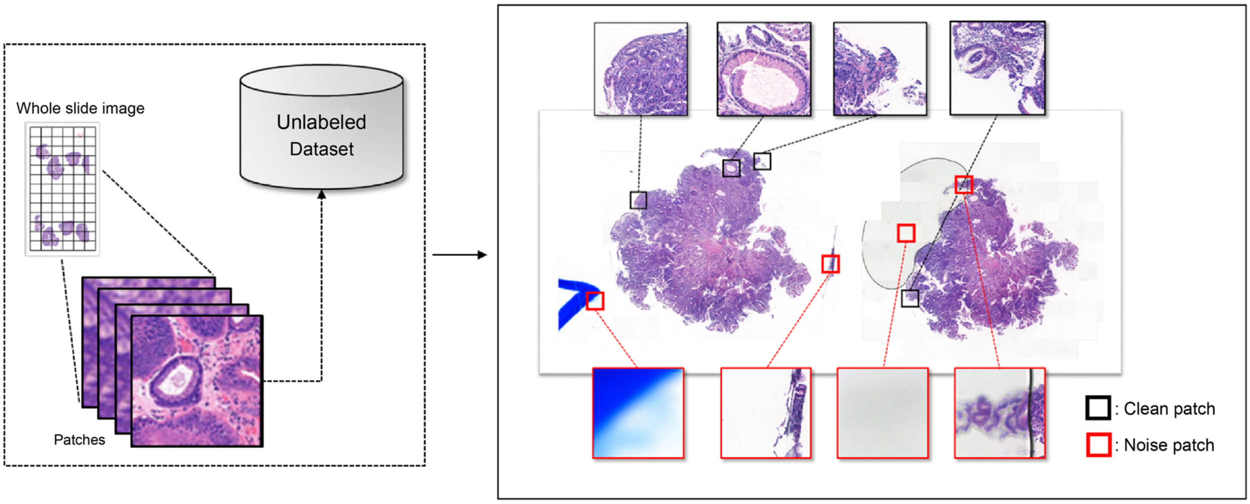


Fig. 4 Patch image generation process. Generation using clean (black boxes) and noise (red boxes) patch images from a field pathology slide.

Table 1 Summary of dataset construction

	Slide		Patch	
D_U	Total	600	Clean	111,611
			Noisy	6,920
			Total	118,531
D_L	-		1st iteration	1,000
			Available cost, *k*	1,000
			Final iteration	10,000
Test	Benign	50	Benign	11,753
	Dysplasia	50	Dysplasia	8,281
	Malignant	50	Malignant	10,489
	Total	150	Total	30,523

patches that pose challenges in tissue classification due to folding that occurs during the slide creation process. In contrast, unlike the aforementioned noisy patches, clean patches contain sufficient tissue shape and information. Figure 5(c) displays these clean patches alongside their corresponding labels.

Implementation details

The proposed method was implemented in Python (Wilmington, DA, USA), using the PyTorch library on a server equipped with two NVIDIA RTX 3080 TI GPUs (Nvidia Corporation, Santa Clara, CA, USA). The goal of this study was to develop a robust AL strategy for industrial applications using noisy data. In most AL studies, the efficacy of each AL method is typically assessed by

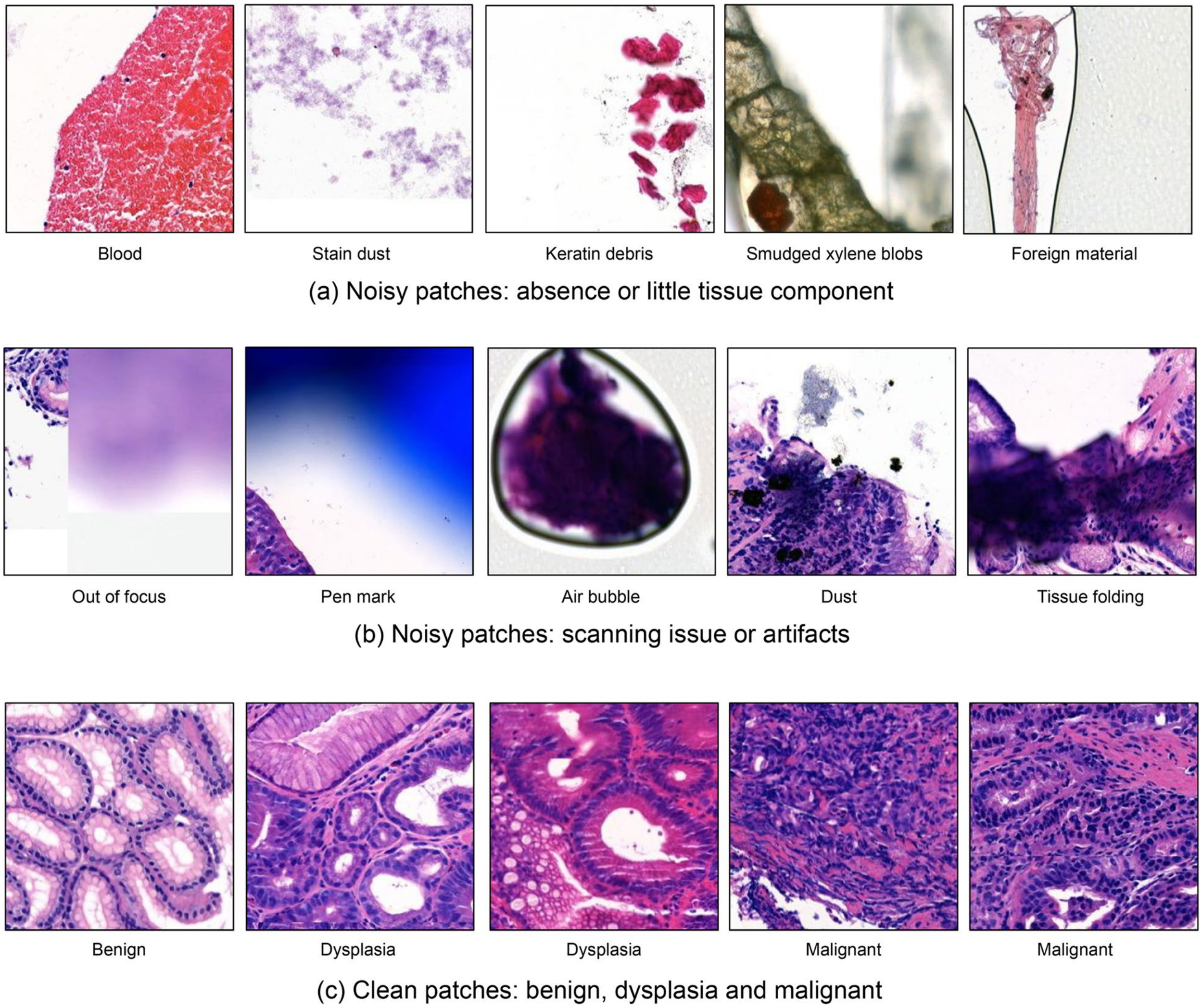


Fig. 5 Examples of noisy and clean patches (**a**) noisy patches that occur during the imaging process, involving substances or objects that are irrelevant to the classification, or of no interest; **b** noisy patches where meaningful tissue imaging is hindered by scanner errors or the presence of other substances; **c** clean patches that contain sufficient information for an accurate classification by an oracle

observing the model's performance changes while varying data selection. Thus, tracking model performance changes associated with a chosen AL method is crucial. Because VGG16 possesses a relatively simpler structure than more advanced deep learning networks, it allowed us to easily track model performance. Moreover, instead of pretrained models, most AL research employs a learning method referred to as training from scratch, which involves training a neural network from the beginning. For these reasons, we chose VGG16 as our backbone model, training it from scratch. This CNN model was trained with the Stochastic Gradient Descent optimizer, and we set the batch size to 32, epoch size to 50, and learning rate to 0.001. Additionally, the backbone model used cross-entropy as a loss function, and the loss function of the LPM used the same margin-ranking loss as specified in Eq. (2).

Furthermore, we implemented 10 iterations of experiments to examine the effectiveness of the AL method. The "available cost" k, which is the number of data selected for labeling by the AL method, was set to the value of 1,000. By repeating this 10 times, a total of 10,000 patch images were selected in the last iteration. If noisy patches were selected during the AL process, they were excluded from the AL cycle because they could not be labeled. That is, p_n is a set of noisy patches among the selected patches by the model, and p_l is a set of labelable patches. In each iteration, two sets p_n and p_l were excluded from D_U, and p_l was added to D_L. Therefore, it can be represented as $D_{Ui+1} = D_{Ui}$ - p_n - p_l,$D_{Li+1} = D_{Li} + p_l$ in the next iteration (i+1). In this experiment, because the model selects 1,000 patches in each iteration, the sum of the elements of the two sets p_n and p_l in each iteration was 1,000 (i.e., $|p_n| + |p_l| = 1{,}000$).

Comparison methods

To evaluate the effectiveness of the proposed method, we compared its performance by applying the proposed method and six different AL methods to the same backbone model. They were least confidence (LC), entropy, Bayesian AL by disagreement (BALD), LL, core-set, and random sampling (RS).

LC [29] queries the most uncertain examples with the lowest softmax confidence while predicting their labels. This method assumes that the model n-classes output nodes are denoted by $\overrightarrow{z}$ and each output node is denoted by z_j. Thus, $j \in [1, n\text{-classes}]$. Then, for an output node z_i from the model, the corresponding softmax is

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{7}$$

The softmax can then use the selected number of data points to select the label for which the model has the lowest confidence, as follows:

$$\text{argmin}_{S \subseteq D_U, |S| \leq k} \sum\nolimits_S (argmax_j(\sigma(\overrightarrow{z}))) \tag{8}$$

Entropy [30]-based AL computes the entropy from a softmax output vector. It is one of the basic AL methods that selects images for which the model is most uncertain. To quantify the uncertainty, entropy is used, and thus, images with maximum entropy are selected. While assuming the model has n-class output nodes and each output node is denoted by z_j ($j \in [1, n\text{-classes}]$), the entropy can then be calculated as:

$$\text{Entropy} = -\sum_j \sigma(z_j) * \log(\sigma(z_j)). \tag{9}$$

This algorithm selects k number of data points for label with the highest entropy.

BALD [19] is an AL method that operates under a Bayesian setting and selects data that maximizes the mutual information between the predicted labels and model parameters. To implement BALD, dropout layers are added to the DL model so that it can be performed in a non-Bayesian setting. Stochastic forward passes are then performed through the dropout, and the difference in prediction entropy (mutual information) is measured. Finally, the data with the highest mutual information is selected for labeling.

LL [26] queries examples with the highest predicted loss by jointly learning an LPM. Unlike other existing uncertainty-based AL methods that rely on additional predictive loss information from the model, LL uses a dedicated module to predict the loss values and selects data based on these predictions.

Core-set [31] is the most popular representativeness-based AL method, which places the data in a feature space and then selects the data that contain the most diverse samples as informative samples. Among the core-set-based methods, we used the k-greedy method. This method aims to select k data points that minimize the radius of the subgraph when placing data in the feature space and repeatedly selecting the furthest data point from one randomly selected data point to produce k subgraphs covering all data points. As a result, the core-set is a method for selecting the most divergent k-examples with the highest coverage in a representative space.

Finally, the RS method is a non-AL approach that randomly selects data without considering their uncertainty or representativeness.

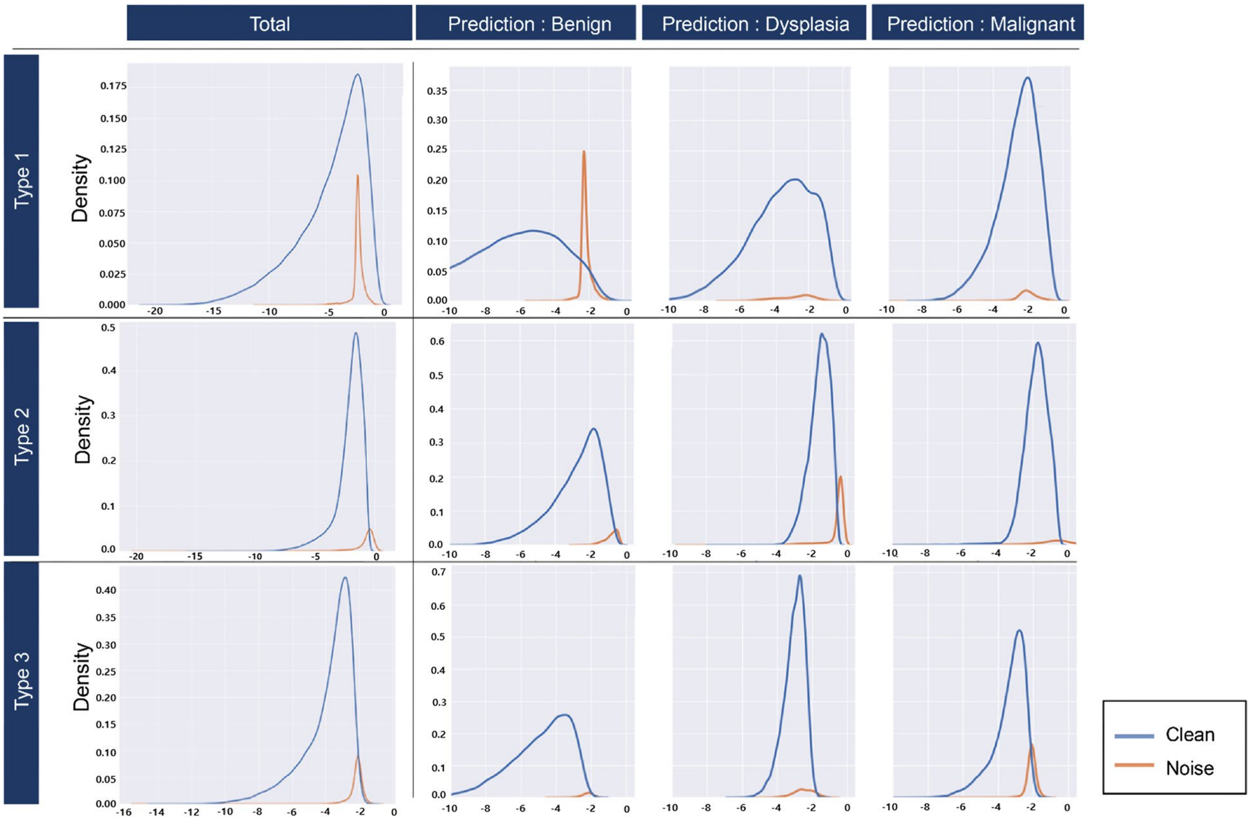
Results

Predicted loss value analysis

We first compared the predicted loss values of trackable noisy data and clean data included in D_U. Representative types of distributions were chosen as typical examples from the distributions generated during three trials of the AL method (LL) with 10 iterations each, resulting in a total of 30 iterations for D_U (Fig. 6). Most noisy data were found to have relatively high predicted loss values. Figure 6(a) shows the result of operating the LPM for the entire unlabeled data, while Fig. 6(b) shows the result of classifying the same data by the predicted class. We identified three main types of noisy data, with Type 1 showing a large distribution of noisy data in the middle position. When classifying them by predicted class and creating a distribution (Fig. 6[b]), each class still had a large number of distributions at a high-loss location. Therefore, we expected that different thresholds would work for each class, as we proposed. Thus, the proposed method included the selection of clean and beneficial data below the threshold by computing different thresholds for each class.

Sensitivity analysis for alpha

Tables 2 and 3 present the results of a sensitivity analysis based on alpha values, along with the corresponding noise selection results. Alpha represents the weight value a_t that determines the threshold in Eq. (5). We evaluated the model's performance and the ratio of noise selection by incrementally increasing this weight value from a low starting point. In the 'no threshold' case, the model had a high proportion (35%) of noisy data. Table 3 shows the average number of noise selections that occurred in 10 iterations and five trials. For instance, the value "353.1" means that, on average, 353 out of 1,000 noisy samples were selected. As a result, the average performance was low, and the model's performance was relatively unstable. When we increased the alpha value, we observed that the rate of noise selection decreased. However, performance no longer improved



(a) Predicted loss (total)

(b) Predicted loss for predicted class

Fig. 6 Example of the distribution of predicted loss values (**a**) Distribution types when predicting the loss with a trained LPM on the entire unlabeled dataset. Most of the noisy data aligns with the high-loss location similar to informative data. **b** Distribution obtained by predicting the loss with the same trained LPM as in (**a**) and classifying it into predicted classes. Within these predicted classes, it can be observed that noisy data are distributed in the higher loss location compared to informative data

Table 2 Sensitivity analysis for alpha (accuracy)

Noisy dataset								Clean dataset	
a	**No threshold (LL)**	**0.5**	**0.8**	**1**	**Proposed**	**1.1**	**1.2**	**LL**	**Proposed**
1	0.810 ± 0.012	0.812 ± 0.010	0.808 ± 0.014	0.815 ± 0.015	0.803 ± 0.007	0.811 ± 0.009	0.808 ± 0.017	0.805 ± 0.004	0.809 ± 0.004
2	0.794 ± 0.040	0.861 ± 0.006	0.817 ± 0.028	0.850 ± 0.020	0.856 ± 0.027	0.833 ± 0.025	0.839 ± 0.023	0.786 ± 0.124	0.846 ± 0.010
3	0.821 ± 0.035	0.808 ± 0.137	0.847 ± 0.030	0.868 ± 0.014	0.864 ± 0.032	0.842 ± 0.034	0.863 ± 0.015	0.855 ± 0.042	0.873 ± 0.013
4	0.815 ± 0.064	0.866 ± 0.011	0.876 ± 0.020	0.880 ± 0.014	0.881 ± 0.026	0.863 ± 0.017	0.860 ± 0.019	0.897 ± 0.012	0.892 ± 0.011
5	0.857 ± 0.044	0.675 ± 0.222	0.892 ± 0.010	0.881 ± 0.013	0.893 ± 0.018	0.892 ± 0.005	0.876 ± 0.029	0.827 ± 0.098	0.902 ± 0.019
6	0.859 ± 0.018	0.898 ± 0.010	0.879 ± 0.015	0.885 ± 0.028	0.899 ± 0.009	0.897 ± 0.009	0.887 ± 0.012	0.894 ± 0.027	0.892 ± 0.031
7	0.843 ± 0.048	0.870 ± 0.027	0.879 ± 0.035	0.905 ± 0.005	0.897 ± 0.021	0.893 ± 0.014	0.888 ± 0.011	0.903 ± 0.007	0.910 ± 0.009
8	0.897 ± 0.019	0.894 ± 0.025	0.906 ± 0.004	0.901 ± 0.014	0.908 ± 0.007	0.892 ± 0.022	0.873 ± 0.020	0.907 ± 0.010	0.912 ± 0.015
9	0.882 ± 0.045	0.882 ± 0.033	0.894 ± 0.021	0.904 ± 0.015	0.914 ± 0.007	0.878 ± 0.045	0.895 ± 0.010	0.882 ± 0.033	0.908 ± 0.019
10	0.863 ± 0.062	0.890 ± 0.019	0.898 ± 0.014	0.917 ± 0.006	0.917 ± 0.004	0.899 ± 0.016	0.888 ± 0.008	0.896 ± 0.014	0.924 ± 0.005

Table 2 shows the change in the accuracy of the model according to the change in the alpha value. We obtained the mean and standard deviation at each iteration after five trials for one alpha value. One trial includes 10 iterations and after 10 iterations, the next trial can be processed

LL Learning loss

Table 3 Sensitivity analysis for alpha (noise selection)

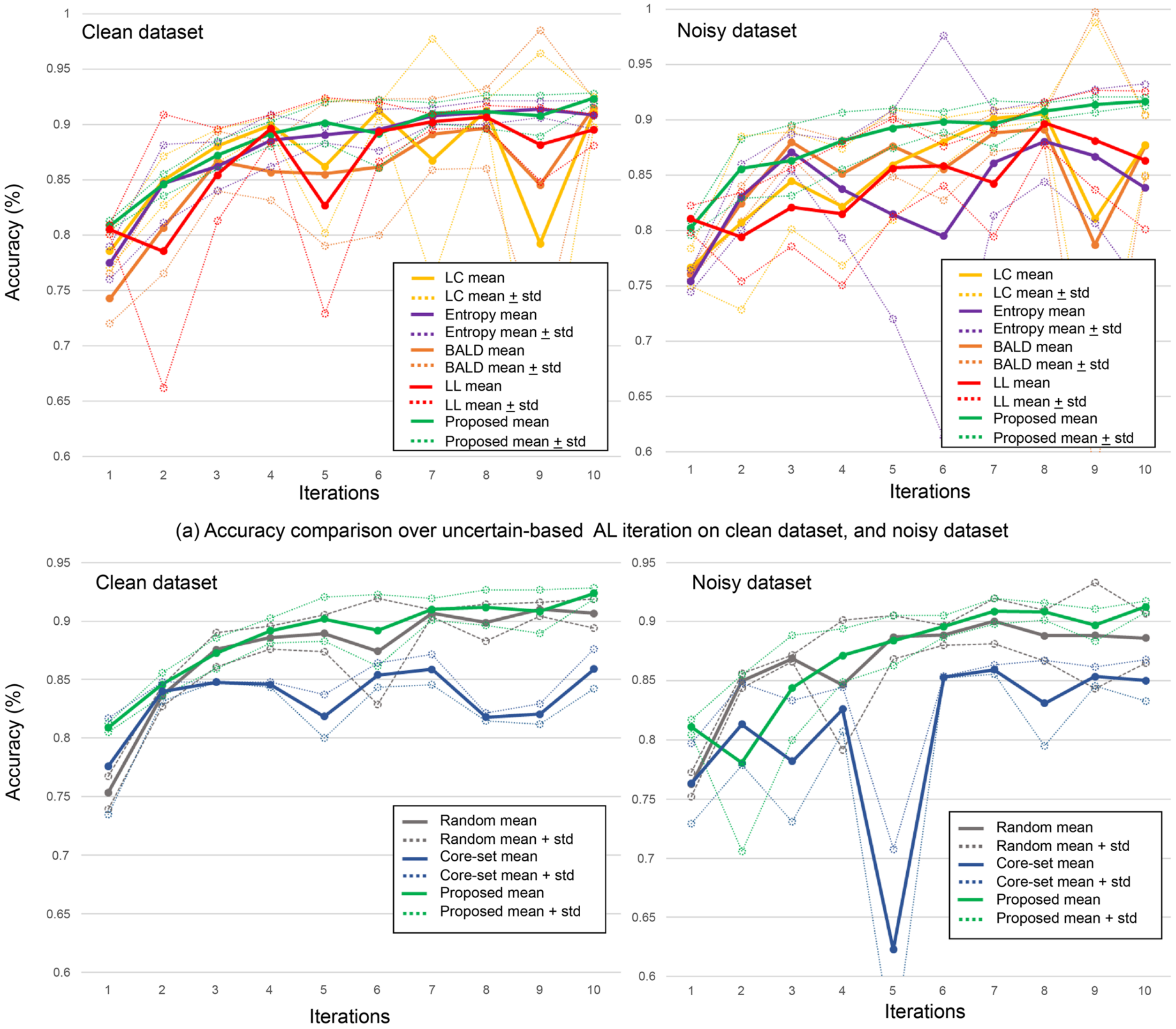
	0 (LL)	**0.5**	**0.8**	**1**	**Proposed**	**1.1**	**1.2**
Mean	353.1	116.4	96.3	80	29.3	24.9	16.6
Total	3177.6	1047.4	866.6	720.4	263.4	224	149.6
N-Ratio	35.31%	11.64%	9.63	8.00%	2.93%	2.49%	1.66%

in larger alpha intervals ($\boldsymbol{a}$ = 1.1 and 1.2). As the alpha value increased, the model tended to select data with relatively low loss values, which reduced the benefits of the selected data and failed to effectively increase performance. Nevertheless, the proposed method achieved high performance with a low noise selection ratio. We applied different alpha values for each class, using “1.0” for Benign, “1.1” for Dysplasia, and “1.0” for Malignant.

Table 3 shows the number of noise selections by the model in five trials. The “Total” item is the average of the total number of noise selections in the five trials. In each trial, the first iteration is excluded from the noise selection amount analysis because clean data are used fixedly at the first iteration. Therefore, the “mean” item is a value obtained by dividing the total item value by nine. The N-ratio refers to the percentage of noise selected, in other words, the ratio of noisy data out of the 1,000 data points selected by the model in each iteration.

Comparative performances of the AL methods

The performances of five alternative AL methods (i.e., LC, entropy, BALD, core-set, random, and proposed) were first evaluated by analyzing the mean and standard deviation of the accuracy obtained from each of the models. We compared the accuracies of the methods in clean and noisy datasets, and also compared and analyzed the amount of noisy data among the data selected by each of the AL methods. Figure 7 presents the accuracy measurements of each AL method in an environment where real-world noise can be used. We measured the performances at each iteration while increasing the size of the labeled data over 10 iterations. In the first iteration, we trained all models with the same clean data. Additionally, we conducted experiments five times for each method and calculated their means and variances. More specifically, Fig. 7(a) compares the uncertainty-based AL method (LC, entropy, BALD) with the proposed method. These AL methods operating on clean datasets had a performance of over 90% in the 10th iteration, which was slightly lower than that of the proposed method. However, when we conducted the same experiment on the noisy dataset, the overall average performances of these AL methods decreased, and the variance increased. In contrast, the proposed method confirmed that the difference in tendency between the results of performing AL with clean data and the results of performing AL with noisy data was not noticeable and the variance remained low. Figure 7(b) compares the core-set and RS methods, which are representativeness-based methods, with the proposed method. The core-set method had a low overall performance due to the large amount of computation required. In the case of the RS method, it showed a



(a) Accuracy comparison over uncertain-based AL iteration on clean dataset, and noisy dataset

(b) Accuracy comparison over other AL iteration on clean dataset, and noisy dataset

Fig. 7 Accuracy comparison over AL methods with clean and noisy datasets. **a** Comparison between the proposed and uncertainty-based methods. The results on a clean dataset are presented on the left, while the results on a noisy dataset are displayed on the right. In each experimental trial, there were 10 sampling iterations, and this process was repeated five times for each method to calculate the accuracy. The figure shows the average and standard deviation of each iteration from the five trials. **b** Comparison of random sampling, coreset, and proposed methods, using the same experimental setup as in (**a**). *Abbreviations AL* active learning, *LC* least confidence, *BALD*, Bayesian active learning by disagreement, LL: learning loss; std: standard deviation

higher variance when learning from a clean dataset and a decrease in stability when learning from a noisy dataset. Both methods showed poor performances, relative to the proposed method.

Figure 7 (b) shows that both the representativeness-based method and the RS method, which induced the selection of various data, showed poor overall performances, supporting the idea that selecting data with higher uncertainty than representativeness is more effective in improving the model's performance in learning pathological tissue image classification. However, a relatively low variance was observed in the core-set and RS methods, indicating that learning various data with representations of the stability of model performance changes had a lasting effect on model training.

The entropy and the proposed methods had an accuracy of over 90% from the seventh iteration on the clean dataset and maintained a performance of more than 90%

until the 10th iteration, the final iteration. Moreover, the proposed method showed a steady increase in performance up to the 10th iteration, where it had its highest performance. For some AL methods, the performance did not increase gradually and, in general, we observed a higher variance than with the proposed method, implying that those models' performances can be highly sensitive to the data selected in each iteration.

Moreover, Fig. 7 provides a comparison of the performance changes on the noisy dataset. Most AL models showed a decrease in average performance and an increase in variance. In contrast, the proposed method demonstrated robust performance against real-world noise, with only a slight performance reduction on the noisy dataset. Specifically, the proposed method had 91.7% accuracy on the noisy dataset, revealing a performance reduction of only 0.7% compared to when it was trained under the clean dataset condition. In contrast, other AL methods had a 2% to 7% performance reduction on the noisy dataset when compared to the clean dataset condition.

Comparative noise selection of the AL methods

To delve into the performance differences under the noisy conditions, we recorded the amount of noisy data selected by each AL method during the aforementioned experimental process and compared the results. The results are graphically summarized in Fig. 8.

Further, Fig. 8 shows the cumulative quantity of noisy data selected by each AL method in terms of iterative performance. Methods that measure uncertainty, such as BALD and entropy, consistently selected a significant amount of noise compared to other methodologies. The core-set and RS methods tended to select a relatively small amount of noisy data because various data were selected. Nevertheless, the proposed method, an AL method for selecting data, tended to select the smallest amount of data, except for intervals where noisy data were prevalent. Table 3 shows the average amount of noisy data selected in each iteration. For the first iteration, the same 1,000 clean data points were used, thus it was not included in the average calculation. Additionally, the ratio of noise selected in each iteration was calculated and is presented in the N-Ratio row of Table 3. N-Ratio represents the ratio of the number of data corresponding to noisy data among the 1,000 data points selected for each iteration and was calculated using the following equation.

$$N - Ratio = \frac{(mean\ of\ noise\ slection)}{1000} \times 100 \tag{10}$$

As shown in Table 4, the proposed method selected an average of 29.3 noisy data points out of 1,000 data points in each iteration, resulting in an average noisy data ratio of 2.93% and a clean data selection ratio of 97.07%. In contrast, the entropy method selected approximately 4,000 noisy data points throughout the experiment, which means that more than 400 noisy data points were selected per iteration. Therefore, the labeled data obtained by the entropy method only accounted for approximately 56% of the total data, which was significantly lower than the proposed method's labeling acquisition rate of approximately 97%. Additionally, in the case of the core-set or RS method, various data were selected, resulting in a noisy data ratio of approximately 6%, which was similar to the 5.8% ratio of the total data.

Comparisons of performance in the final iteration

Table 5 presents the accuracy scores of all of the aforementioned methods obtained in the final iteration for clean and noisy datasets. The methods that measure uncertainty, such as BALD and entropy, consistently selected a significant amount of noise compared to the other methods. The core-set and RS methods tended to

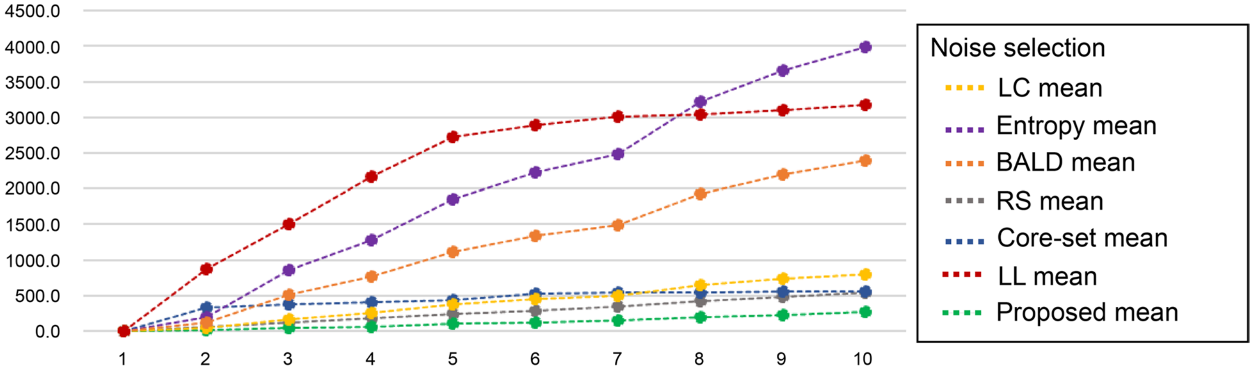


Fig. 8 A cumulative graph depicting the number of noise data selections over iterations by alternative AL methods. AL: active learning; LC: least confidence; BALD: Bayesian active learning by disagreement; RS: random sampling; LL: learning loss

Table 4 Number of noisy data selections by the AL methods in each iteration

Iteration	LC	Entropy	BALD	LL	RS	Core-set	Proposed
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	38.2	191.0	114.6	878.4	60.3	326.3	15.2
3	132.0	660.0	396.0	626.4	57.8	55.0	28.2
4	86.1	430.6	258.3	667.8	61.8	17.7	20.4
5	114.3	571.4	342.8	558.4	57.5	30.3	36.0
6	74.6	373.0	223.8	161.2	53.3	101.3	25.8
7	50.7	253.4	152.0	115.2	60.0	9.3	31.0
8	148.2	740.8	444.5	35	65.5	7.7	33.8
9	88.5	442.6	265.6	62	60.5	7.7	37.0
10	67.0	335.2	201.1	73.2	59.0	6.0	36.0
Mean	88.8	444.2	266.5	353.1	59.5	62.36	29.3
Total	799.6	3998	2398.8	3177.6	535.5	561.3	263.4
N-Ratio	8.9%	44.4%	26.7%	35.3%	6.0%	6.2%	2.93%

LC Least confidence, *BALD* Bayesian active learning by disagreement, *LL* Learning loss, *RS* Random sampling, *N-Ratio* Noise-ratio

Table 5 Accuracy and standard deviation scores obtained in the final iteration for clean and noisy datasets

Dataset	LC	Entropy	BALD	RS	Core-set	LL	Proposed
Clean	0.912 ± 0.014	0.909 ± 0.013	0.916 ± 0.007	0.907 ± 0.012	0.859 ± 0.017	0.896 ± 0.014	0.924 ± 0.005
Noisy	0.878 ± 0.027	0.877 ± 0.028	0.838 ± 0.094	0.886 ± 0.021	0.850 ± 0.018	0.863 ± 0.062	0.917 ± 0.004

Abbreviations LC Least confidence, *BALD* Bayesian active learning by disagreement, *RS* Random sampling

Table 6 Total noise selection by each AL method

	LC	Entropy	BALD	RS	Core-set	LL	Proposed
Mean	88.8	444.2	266.5	59.5	62.36	353.1	29.3
Total	799.6	3998	2398.8	535.5	561.3	3177.6	263.4
N-Ratio	8.90%	44.40%	26.70%	6.00%	6.20%	35.3%	2.93%

Abbreviations AL Active learning, *LC* Least confidence, *BALD* Bayesian active learning by disagreement, *RS* Random sampling

select a relatively small amount of noisy data because a large number of noisy data was distributed in sections of relatively high uncertainty values and various levels of data were selected by the two methods. Nevertheless, the proposed method tended to select the smallest amount of noisy data, except for intervals where noisy data were prevalent. Table 6 also shows the number and proportion of noisy data selected by each method. A high N-ratio means that more noisy data is included in the selected data. The proposed method showed the lowest level of noise selection among the alternative AL methods.

Discussion

The objective of this study was to develop a robust AL method against a noisy histopathological dataset. Constructing a CNN-based DL system requires a pathologist to perform image-labeling tasks [15, 20, 27]. AL has been studied to reduce the workload of the labeling by selectively labeling data that are more effective for learning, which lowers labeling costs and reduces the workload of an oracle [7, 22, 32]. However, in real-world industrial environments, various forms of noisy data are included in WSIs [24, 25, 33]. If the model selects noisy data, an oracle's workload is not effectively reduced. To address this problem, this paper proposes a new method to select data with predictive losses below a certain threshold to develop a robust AL method.

Pathological datasets can be noisy for a variety of reasons, such as interference during image capture or data conversion, mislabeled data, or out-of-distribution samples. Previous studies have addressed this issue by removing noise from the images or separating the noisy data. There are studies aimed at alleviating the interfering image problems by utilizing generative model-based methods and removing perturbations that occur within the image [34–37]. On the other hand, there are studies that have also been conducted to separate noisy data. These studies utilize the features of noisy data in a dataset to perform different processing on noisy vs. clean data and to filter out noisy data from clean data [38, 39]. Additionally, Ponzio et al. [33] applied Bayesian neural networks to measure the uncertainty of the data and to remove data with high uncertainty. Through this method, they filtered spurious data, such as blood, fat, glass, and stroma, for pathological classification. Ashraf et al. [27] proposed a patch-data cleaning method called LossDiff, which automatically sets an appropriate threshold based on the batch average loss for each class.

The proposed AL framework combined uncertainty-based AL with uncertainty-based data-filtering methods. We adopted LL [26] as an informative image selection method, which has several advantages. First, it is simple and task-agnostic, making it suitable for use with deep networks. Deep networks are trained by minimizing a single loss, regardless of task type, number of tasks, or complexity of architecture, which makes LL useful for various purposes, as long as the LPM can be attached. Secondly, LL predicts loss values for unlabeled data, making it possible to calculate thresholds for filtering data collected during the training stage. Finally, LL can be utilized after training to select informative data.

We also focused on data-filtering methods among various existing data-cleaning methods. We used a modified version of LossDiff [27]. We induced the model to generate thresholds by collecting the predicted loss in the training stage and selecting data with lower-than-threshold loss values in the sampling phase, thus allowing the model to select appropriate and informative training data. This is important for robust AL on real-world noisy datasets where noise and informative data are mixed. Most data filtering methods consider data with high uncertainty to be noisy data, which have a high potential for both noisy and informative data. Therefore, our proposed framework ensured that the model selected informative data and filtered out noisy data.

Additionally, we confirmed the tendency of the predicted loss values of the noisy data and clean data and their distribution. The results showed that the noisy data were distributed in a high uncertainty (i.e., high predicted-loss) section in each prediction class distribution. Therefore, different threshold settings were required depending on the prediction class for proper threshold generation. Subsequently, a sensitivity analysis was performed on the alpha value to select an appropriate value. The results of the sensitivity analysis demonstrated a performance change in the proportion of noisy data in the selected data as the alpha value gradually increased. We experimentally found alpha values that showed good performance, while also selecting less noise, and showed the best results when the values were set differently, according to the predicted class (Benign: "1.0", Dysplasia: "1.1", Malignant: "1.0").

In the 10th iteration on the noisy dataset, our model had an accuracy of 0.917, with a performance reduction of less than 1% (0.924–0.917), and it was confirmed that only 29.3 noisy data points were selected on average from 1,000 selected data points. We performed comparisons with other methodologies in repeated experiments with corresponding alpha values. The proposed method showed indistinguishable performance differences on clean datasets and noisy datasets, and a lower numerical noise selection ratio than other AL methods. The study results clearly showed that the performance of the model generated by the proposed method was robust, even in noisy environments.

Tables 5 and 6 provide an overview of the results from our experimental study. On the 10th iteration on the noisy dataset, the proposed method showed a mean of accuracy and standard deviation of 0.917 and 0.004, respectively, unlike other methods with a 3–7% performance reduction from the clean dataset. Our proposed method exhibited only a slight performance reduction of 0.7%. These findings support the notion that our proposed method is more robust than existing methods with noise. For the core-set method, the 10th iteration had an accuracy of 0.850, which was approximately 6% lower than the performance of the proposed model. Moreover, for the uncertainty-based methods, the differences in performance between the noisy and the clean datasets were larger. In the case of BALD, the most affected by the noise, a performance reduction of approximately 7% was found. The proposed method demonstrated little variation in performance on both clean and noisy datasets, with a lower numerical noise selection ratio than the other AL methods. In contrast, the proposed method achieved the highest accuracy on both noisy and clean datasets, while maintaining the smallest performance decline against the noisy dataset. Furthermore, we observed that our proposed AL method selected the least amount of noisy data.

Notably, although LC and entropy, which are uncertainty-based AL methods, differ significantly in their noise selection ratios, the differences in performance were negligible. Furthermore, even with BALD, which selected even fewer noisy data points than entropy, the results showed that selecting less noise was not a guarantee of good performance. This point was further emphasized by the core-set and RS methods. By performing two functions simultaneously, beneficial data selection and noisy data avoidance, and in a straightforward manner, our proposed method yielded high performance and selected more labeled data. In fact, given that only 56% of the labeled data was obtained by the extant entropy method, the actual amount of data acquired was only approximately half of that by the proposed method. In contrast to the average of 3,998 noisy data points selected out of 10,000 by the entropy methods, it was confirmed that only an average of 263.4 were selected by the proposed method, indicating that utilizing existing AL methods on real-world noisy datasets would increase the workload of the oracle due to the need for more labeling tasks. At the same time, the proposed method minimized the increase in the oracle's workload by performing AL in noisy environments, with an average clean data selection rate of 97.07%.

The proposed method can be applied to training an image classification model aptly in real-world industrial practice with noisy data. We experimented with alternative AL strategies using patch images generated from real-world WSIs, enabling us to test the proposed method against real-world noisy data and understand its performances and general tendencies, relative to the state-of-the-art methods. Among them, the proposed method showed better performances in terms of noise selection levels and accuracies.

Nevertheless, this study's proposed method has certain limitations and boundary conditions that need to be noted. We used predictive losses collected during the training process to create thresholds and exclude data above them from candidate data, to avoid selecting noisy data. However, there can be noises that are difficult to classify, thereby making it difficult to achieve good model performance. For example, in the case of noise labels that arise during the oracle's labeling process, the quality of the labeled data degraded pose an inherent risk to the collected label data. To reduce noise labels that may occur in various AL scenarios, the data cleaning method needs to be extended to the AL training data. Additionally, the proposed method focused on quantitatively reducing the workload of the oracle. However, considering that the task weight caused by the actual labeling process is not only affected by the amount of data, it is also necessary to consider how to qualitatively reduce the task weight during the labeling process. Therefore, in our future studies, we plan to analyze the differences between the data that experts find beneficial and those that the AL method judges to be beneficial.

Conclusions

In this study, we propose a novel AL method for pathological image classification that minimizes noisy data selection when querying data from an unlabeled set. Our model selected data with high informativeness while avoiding the distribution interval of noisy data, by taking the characteristics of the predicted loss values of the noisy patches occurring in the field into account. When we trained our model using this method, it achieved an accuracy of 91.7% on the noisy dataset and 92.4% on the clean dataset in the final iteration, resulting in a performance reduction of less than 1%.

With its reduced noise selection ratio and increased accuracy, the proposed method may contribute to relieving the workload of pathologists in the context of AL applied to automated image processing for cancer detection in the workplace. The data reflected the actual level of noise embedded in the WSIs created for a large medical diagnosis organization in Korea. Against this dataset collected from a real workplace, the proposed method produced a superior and more robust performance, compared to the state-of-the-art methods. The study findings are expected to be applicable to other pathological image processing areas, even though the proposed method was tested against stomach images generated by a high-quality medical scanner. Furthermore, the DL model training system proposed in this study has the potential to enhance the working environment for pathologists while continuously improving DL models. This advancement can result in better resource utilization, increased productivity, and ultimately benefit both pathologists and the patients they care for.

Abbreviations

AL	Active learning
BALD	Bayesian active learning by disagreement
CNNs	Convolutional Neural Networks
DL	Deep learning
LC	Least confidence
LL	Learning loss
LPM	Loss prediction module
PLCM	Predicted loss collection module
RS	Random sampling
s/f	Suspicious for
s/o	Suspicious of
WSIs	Whole-slide images

Acknowledgements
Not applicable.

Authors' contributions

Mujin Kim: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Visualization, Supervision, and Project administration. Willmer Rafell Quiñones Robles: Software, Validation, and Investigation. Young Sin Ko: Resources, Data Curation, and Writing - Review & Editing. Bryan Wong: Software. Sol Lee: Software. Mun Yong Yi: Conceptualization, Funding acquisition, Supervision, and Writing – review & editing. All authors read and approved the final manuscript.

Funding

This research was supported by the Seegene Medical Foundation, South Korea, under the project "Research on Developing a Next Generation Medical Diagnosis System Using Deep Learning" (Grant Number: G01180115). Funding body had no role in the design of the study, analysis and interpretation of data, and in writing the manuscript. but contributed to the collection of data.

Availability of data and materials

Patient clinical data and whole slide images that support the findings of this study cannot be made publicly available due to ethical concerns regarding patient privacy. The datasets collected and analyzed during the current study are not publicly available due to limited computing, storage resources, and the security policy of Seegene Medical Foundation. However, they are restrictively available from the Seegene Medical Foundation Institutional Data Access/Ethics Committee (contact via corresponding authors) for researchers who meet the criteria for access to confidential data.

Declarations

Ethics approval and consent to participate

This study was approved by the institutional review board of Seegene Medical Foundation and the Institutional Review Board (KAIST-IRB-22-334, KH2020-116) of the Korea Advanced Institute of Science and Technology, the university that collaborated with the medical foundation.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 27 July 2023 Accepted: 1 December 2023
Published online: 02 January 2024

References

1. Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, et al. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA Cancer J Clin. 2021;71:209–49.
2. Siegel RL, Miller KD, Fuchs HE, Jemal A. Cancer Statistics, 2021. CA Cancer J Clin. 2021;71:7–33.
3. Bray F, Laversanne M, Weiderpass E, Soerjomataram I. The ever-increasing importance of cancer as a leading cause of premature death worldwide. Cancer. 2021;127:3029–30.
4. Irshad H, Veillard A, Roux L, Racoceanu D. Methods for nuclei detection, segmentation, and classification in digital histopathology: a review—current status and future potential. IEEE Rev Biomed Eng. 2014;7:97–114.
5. Metter DM, Colgan TJ, Leung ST, Timmons CF, Park JY. Trends in the US and Canadian pathologist workforces from 2007 to 2017. JAMA Netw Open. 2019; https://doi.org/10.1001/jamanetworkopen.2019.4337.
6. Peck M, Moffat D, Latham B, Badrick T. Review of diagnostic error in anatomical pathology and the role and value of second opinions in error prevention. J Clin Pathol. 2018;71:995–1000.
7. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal. 2017;42:60–88.
8. Aresta G, Araújo T, Kwok S, Chennamsetty SS, Safwan M, Alex V, et al. BACH: Grand challenge on breast cancer histology images. Med Image Anal. 2019;56:122–39.
9. Bandi P, Geessink O, Manson Q, Van Dijk M, Balkenhol M, Hermsen M, et al. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. IEEE Trans Med Imaging. 2019;38:550–60.
10. Wahab N, Khan A, Lee YS. Two-phase deep convolutional neural network for reducing class skewness in histopathological images based breast cancer detection. Comput Biol Med. 2017;85:86–97.
11. Takahama S, Kurose Y, Mukuta Y, Abe H, Fukayama M, Yoshizawa A, et al. Multi-stage pathological image classification using semantic segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV): 2019 October 27 – November 2; Seoul. 2019 IEEE/CVF International Conference on Computer Vision (ICCV): IEEE; 2020: 10701–10. https://doi.org/10.1109/ICCV.2019.01080.
12. Wang X, Chen H, Gan C, Lin H, Dou Q, Tsougenis E, et al. Weakly supervised deep learning for whole slide lung cancer image analysis. IEEE Trans Cybern. 2020;50:3950–62.
13. Hou L, Samaras D, Kurc TM, Gao Y, Davis JE, Saltz JH. Patch-based convolutional neural network for whole slide tissue image classification. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): Las Vegas. IEEE; 2016: 2424–33. https://doi.org/10.1109/CVPR.2016.266.
14. Hussain Z, Gimenez F, Yi D, Rubin D. Differential data augmentation techniques for medical imaging classification tasks. AMIA Annu Symp Proc. 2018;2017:979–84.
15. Meirelles AL, Kurc T, Saltz J, Teodoro G. Effective active learning in digital pathology: a case study in tumor infiltrating lymphocytes. Comput Methods Programs Biomed. 2022;220:106828.
16. Cheplygina V, de Bruijne M, Pluim JPW. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Med Image Anal. 2019;54:280–96.
17. Budd S, Robinson EC, Kainz B. A survey on active learning and human-in-the-loop deep learning for medical image analysis. Med Image Anal. 2021;71:102062.
18. Liu X, Wang K, Chen T. Deep active learning for fibrosis segmentation of chest Ct scans from Covid-19 patients. In: 2021 IEEE International Conference on Image Processing (ICIP): 2021 September 19-22 2023. Anchorage. IEEE, 2021:175-79. https://doi.org/10.1109/ICIP42928.2021.9506413
19. Gal Y, Islam R, Ghahramani Z. Deep Bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning ; 2017: JMLR; 70: 1183–92. https://dl.acm.org/doi/10.5555/3305381.3305504.
20. Wen S, Kurc TM, Hou L, Saltz JH, Gupta RR, Batiste R, et al. Comparison of different classifiers with active learning to support quality control in nucleus segmentation in pathology images. AMIA Jt Summits Transl Sci Proc. 2018;2017:227–36.
21. Qi Q, Li Y, Wang J, Zheng H, Huang Y, Ding X, et al. Label-efficient breast cancer histopathological image classification. IEEE J Biomed Health Inform. 2019;23:2108–16.
22. Li W, Li J, Wang Z, Polson J, Sisk AE, Sajed DP, et al. Pathal: an active learning framework for histopathology image analysis. IEEE Trans Med Imaging. 2022;41:1176–87.
23. Wu M, Li C, Yao Z. Deep active learning for computer vision tasks: methodologies, applications, and challenges. Appl Sci. 2022;12:8103.
24. Schömig-Markiefka B, Pryalukhin A, Hulla W, Bychkov A, Fukuoka J, Madabhushi A, et al. Quality control stress test for deep learning-based diagnostic model in digital pathology. Mod Pathol. 2021;34:2098–108.
25. Ko YS, Choi YM, Kim M, Park Y, Ashraf M, Quiñones Robles WR, et al. Improving quality control in the routine practice for histopathological interpretation of gastrointestinal endoscopic biopsies using artificial intelligence. PLoS One. 2022;17:e0278542.
26. Yoo D, Kweon IS. Learning loss for active learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. Long Beach, 2019: 93-102https://doi.org/10.1109/CVPR.2019.00018.
27. Ashraf M, Quiñones Robles WR, Kim M, Ko YS, Yi MY. A loss-based patch label denoising method for improving whole-slide image analysis using a convolutional neural network. Sci Rep. 2022;12:1392.
28. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, {ICLR}, San Diego, CA, USA, 2015. https://arxiv.org/abs/1409.1556.
29. Wang D, Shang Y. A new active labeling method for deep learning. In: 2014 International Joint Conference on Neural Networks (IJCNN): 2014 July 6-11; Beijing. 2014 International Joint Conference on Neural Networks (IJCNN): IEEE; 2014: 112–9. https://doi.org/10.1109/IJCNN.2014.6889457.
30. Luo W, Schwing AG, Urtasun R. Latent structured active learning. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems. Lake Tahoe: Neural Information Processing Systems Foundation, Inc. (NeurIPS), Curran Associates, Inc.; 2013. p. 728–36. https://www.proceedings.com/21521.html.

31. Sener O, Savarese S. Active learning for convolutional neural networks: a core-set approach. In: ICLR 2018 Conference. Vancouver: Vancouver Convention Center; 2018. https://arxiv.org/abs/1708.00489 .
32. Ren P, Xiao Y, Chang X, Huang PY, Li Z, Gupta BB, et al. A survey of deep active learning. ACM computing surveys (CSUR). 2020;54:1–40.
33. Ponzio F, Macii E, Ficarra E, Di Cataldo S. W2WNet: A two-module probabilistic Convolutional Neural Network with embedded data cleansing functionality. Expert Syst Appl. 2023;214:119121.
34. Nasrin S, Alom MZ, Burada R, Taha TM, Asari VK. Medical image denoising with recurrent residual u-net (R2u-net) base auto-encoder. In: 2019 IEEE National Aerospace and Electronics Conference (NAECON). Dayton: IEEE; 2019:345–50. https://doi.org/10.1109/NAECON46414.2019.9057834.
35. Tian M, Song K. Boosting magnetic resonance image denoising with generative adversarial networks. IEEE Access. 2021;9:62266–75.
36. Kascenas A, Pugeault N, O'Neil AQ. Denoising autoencoders for unsupervised anomaly detection in brain mri. In: The 5th International Conference on Medical Imaging with Deep Learning. Zurich: PMLR; 2022. p. 653–64.
37. Gil Zuluaga FH, Bardozzo F, Rios Patino JI, Tagliaferri R. Blind microscopy image denoising with a deep residual and multiscale encoder/decoder network. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). Mexico: IEEE; 2021. p. 3483–6.
38. Xue C, Dou Q, Shi X, Chen H, Heng PA, Robust learning at noisy labeled medical images: applied to skin lesion classification. IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE. 2019;2019:1280–3.
39. Le H, Samaras D, Kurc T, Gupta R, Shroyer K, Saltz J. Pancreatic cancer detection in whole slide images using noisy label annotations. In: Internation Conference on Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Cham: Springer International Publishing; 2019. p. 541–9.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.